

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРКАСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ БОГДАНА ХМЕЛЬНИЦЬКОГО

Авраменко В. С., Бєсєдіна С. В.

ВИКОНАННЯ ТА ОФОРМЛЕННЯ
ВИПУСКНИХ РОБІТ

Освітній ступінь «Бакалавр»

Навчально-методичний посібник

Черкаси – 2018

УДК 378.2(075.8)

Рецензенти:

Косенюк Г. В., к.т.н., доцент кафедри інформаційних технологій ЧНУ ім. Б. Хмельницького.

Рудницький В. М., д.т.н., професор Черкаського державного технологічного університету.

Рекомендовано до друку Вченою радою

Черкаського національного університету імені Богдана Хмельницького

(Протокол №8 від 18.06.2018 р.)

Авраменко В. С., Бєседіна С. В.

Виконання та оформлення випускних робіт. Освітній ступінь «Бакалавр»: навчально-методичний посібник. Черкаси: Черкаський національний університет ім. Б. Хмельницького, 2018. 227 с.

ISBN

Навчальний посібник призначений для студентів ЧНУ, які навчаються за спеціальностями 121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки», 123 «Комп'ютерна інженерія», 124 «Системний аналіз». У посібнику розкрито мету й задачі дипломного проектування, докладно охарактеризовані вимоги до структури, змісту і оформлення, також описані порядок, вимоги та критерії оцінки захисту випускних дипломних робіт. Матеріал посібника призначений для наукових керівників та рецензентів, студентів денної та заочної форм навчання, членів екзаменаційних комісій із захисту випускних робіт освітнього ступеня «Бакалавр».

УДК 378.2(075.8)

ISBN

© ЧНУ ім. Б. Хмельницького, 2018

© Авраменко В. С.

© Бєседіна С. В.

ЗМІСТ

ВСТУП.....	6
1 МЕТА ТА ЗАДАЧІ ВИКОНАННЯ ДИПЛОМНОЇ РОБОТИ.....	8
2 ТЕМАТИКА ТА ПОРЯДОК ПІДГОТОВКИ ДИПЛОМНОЇ РОБОТИ....	10
3 ОРГАНІЗАЦІЯ ВИКОНАННЯ ДИПЛОМНОЇ РОБОТИ	14
4 ВИМОГИ ДО СТРУКТУРНИХ ЕЛЕМЕНТІВ ВСТУПНОЇ ЧАСТИНИ РОБОТИ	16
4.1 Титульний лист.....	18
4.2 Завдання на дипломну роботу	18
4.3 Анотація	19
4.4 Зміст дипломної роботи	19
4.5 Перелік умовних скорочень.....	20
5 ВИМОГИ ДО СТРУКТУРНИХ ЕЛЕМЕНТІВ ОСНОВНОЇ ЧАСТИНИ .	21
ВСТУП.....	21
1 ОГЛЯД ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ АНАЛОГІВ ОБ'ЄКТА РОЗРОБКИ.....	25
2 АНАЛІЗ І ПРОЕКТУВАННЯ СИСТЕМИ	27
2.1 Моделювання предметної області.....	31
2.1.1 Поняття предметної області.....	32
2.1.2 Основні елементи моделювання предметної області	33
2.1.3 Визначення робочої області моделювання	37
2.2 Формування та аналіз вимог	40
2.2.1 Формування вимог до програмного забезпечення.....	41
2.2.2 Формування та аналіз вимог за допомогою діаграм прецедентів.....	44
2.3 Проектування логічної структури системи.....	51
2.3.1 Діаграма класів	51
2.3.2 Діаграма пакетів	78
2.4 Архітектурне проектування.....	81
2.4.1 Діаграма компонентів.....	82
2.4.2 Розгортання програмної системи на апаратних засобах	86

2.5	Моделювання поведінки системи.....	91
2.5.1	Діаграма діяльності	92
2.5.2	Діаграма послідовності	98
2.5.3	Діаграма комунікації	106
2.5.4	Діаграма кінцевого автомату	108
2.6	Проектування і розробка бази даних.....	118
2.6.1	Концептуальна модель предметної області	119
2.6.2	Логічна модель бази даних	128
2.6.3	Фізична модель бази даних	133
3	РОЗРОБКА І ТЕСТУВАННЯ СИСТЕМИ	137
3.1	Розробка системи	137
3.2	Тестування системи.....	147
	Висновки.....	149
	Список використаних джерел	150
	Додатки	151
	Відгук керівника та рецензія на дипломну роботу	152
6	ПОРЯДОК ВИКОНАННЯ ТА ОРГАНІЗАЦІЯ ЗАХИСТУ ДР	154
6.1	Робота студента	154
6.2	Організація і виконання дипломної роботи	154
6.3	Нормоконтроль дипломної роботи.....	156
6.4	Відгук керівника дипломної роботи.....	156
6.5	Рішення про допуск дипломної роботи до захисту	159
6.6	Рецензія на дипломну роботу	160
6.7	Підготовка до захисту дипломної роботи	163
6.8	Процедура захисту дипломної роботи	164
6.9	Оцінювання дипломної роботи.....	168
7	ОФОРМЛЕННЯ ДИПЛОМНОЇ РОБОТИ.....	172
7.1	Загальні вимоги	172
7.2	Вимоги до мови і стилю написання роботи.....	174
7.3	Вимоги до нумерації	176

7.3.1 Нумерація сторінок дипломної роботи	176
7.3.2 Нумерація розділів, підрозділів, пунктів, підпунктів	176
7.4 Рисунки	178
7.5 Таблиці	180
7.6 Формули та рівняння	183
7.7 Переліки	185
7.8 Примітки	187
7.9 Підрядкове бібліографічне посилання (виноски)	188
7.10 Позатекстове бібліографічне посилання	189
7.11 Перелік умовних позначок (скорочень)	192
7.12 Оформлення списку використаних джерел	193
7.13 Оформлення додатків	194
7.14 Демонстраційний матеріал	195
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	196
ДОДАТОК А – Зразок титульного листа дипломної роботи	197
ДОДАТОК Б – Завдання та календарний план роботи студента (зразок..	198
ДОДАТОК В – Зразок анотації дипломної роботи	200
ДОДАТОК Д – Бланк відгуку керівника на дипломну роботу	202
ДОДАТОК Ж – Бланк рецензії на дипломну роботу студента	204
ДОДАТОК К – Зразок оформлення списку використаних джерел	206
ДОДАТОК Л – Орієнтовна тематика дипломних робіт	211
ДОДАТОК М – Приклад вступу до дипломної роботи.....	219

ВСТУП

Написання випускної дипломної роботи (далі – ДР або дипломна робота) освітнього ступеня «бакалавр» є заключним та найважливішим етапом навчальної підготовки студента. За наявності в окремих студентів схильності та інтересу до наукових досліджень може бути рекомендоване виконання дипломної роботи науково-дослідницького характеру, що ведуть студенти у процесі всього періоду навчання в університеті.

При виконанні ДР студенти повинні виявити повну самостійність, уміння ставити та розв'язувати поставлені перед ними науково-технічні задачі та обґрунтовувати прийняті науково-технічні рішення як з технічної, так і з науково-технічної та економічної точки зору. Виконуючи дипломну роботу, студент поглиблює знання з фундаментальних та фахових дисциплін; освоює методики програмування технологічних процесів; оволодіває навичками співставлення результатів своїх розробок із літературними даними, аналізу, узагальнення і оформлення одержаних результатів з теми проектування; набуває вміння вести науковий пошук, користуватися передовими комп'ютерними технологіями.

У ДР студент має засвідчити, що він оволодів необхідними теоретичними знаннями та навичками їх практичного застосування в конкретних умовах. У зв'язку з цим робота повинна виконуватись на основі результатів аналізу однієї або кількох актуальних задач та об'єктів і містити конкретні пропозиції.

Опанування проблеми в галузі комп'ютерних наук та програмної інженерії передбачає глибоке вивчення робіт в цій галузі, системний вибір та аналіз фактичного матеріалу, вміння робити самостійні висновки.

Метою виконання ДР є:

- 1) систематизація, закріплення, розширення і перевірка теоретичних знань студентів;

- 2) поглиблене вивчення студентом однієї з конкретних проблем в галузі інформаційних технологій і ґрунтовніше оволодіння навичками самостійної науково-дослідної роботи;
- 3) одержання нових результатів у вигляді інженерних узагальнень, висновків, формування конкретних пропозицій і рекомендацій щодо їх реалізації (упровадження);
- 4) підтвердження кваліфікаційного рівня випускника, спектру та глибини його знань, умінь і навичок у різних сферах функціонування суб'єктів господарювання.

Для успішної підготовки і захисту ДР студенту необхідно:

- обґрунтувати актуальність і значущість теми роботи стосовно до практичних умов функціонування об'єкту проектування;
- провести огляд джерел з предмету аналізу й узагальнити вивчені підходи;
- проаналізувати особливості функціонування об'єкта проектування, встановити закономірності;
- обґрунтувати принципово нові чи прив'язати існуючі рішення у сфері інформаційних технологій, проблем, які розв'язуються на прикладі об'єкта проектування;
- здійснити логічно-системне обґрунтування запропонованих рішень;
- виконати розрахунки очікуваних переваг та ефективності від впровадження запропонованих рішень у практику об'єкта дослідження;
- викласти результати самостійних досліджень по обраній темі.

На прикладі дипломної роботи студент повинен показати вміння грамотно оформляти технічні документи. Цей посібник є зразком для оформлення дипломної роботи на здобуття освітнього ступеня «бакалавр».

1 МЕТА ТА ЗАДАЧІ ВИКОНАННЯ ДИПЛОМНОЇ РОБОТИ

Дипломне проектування проводиться на завершальному етапі освітнього процесу і базується на теоретичних знаннях і практичних навичках, отриманих студентом упродовж усього терміну навчання та самостійної роботи, пов'язаної з розробкою конкретних теоретичних задач прикладного характеру, що визначаються специфікою спеціальності 122 «Комп'ютерні науки» вищої освіти.

Дипломна робота є самостійною творчою роботою аналітичного, розрахункового або експериментального характеру, яка виконана відповідно до вимог освітньої програми випускника закладу вищої освіти. Вона відображає рівень професійної його підготовки, здатності виконувати виробничі функції та типові задачі фахової діяльності.

Мета дипломного проектування – систематизація, розширення, поглиблення і закріплення теоретичних знань і умінь, отриманих студентом за увесь період навчання в університеті.

Під час роботи над дипломною роботою випускник повинен підтвердити наявність необхідних знань, навичок і умінь, що забезпечують рішення професійно-значимих задач у процесі самостійної професійної діяльності.

До початку виконання роботи випускові кафедри мають забезпечити студента методичними вказівками, в яких наведено вимоги щодо змісту і оформлення дипломної роботи.

Дипломна робота виконується студентом і до неї пред'являється вся сукупність задач і вимог, як до робіт такого характеру, а саме:

- високий теоретичний рівень;
- чіткість викладення і побудови роботи;
- аргументованість;
- наповненість достовірними експериментальними даними;
- обґрунтованість застосованих методів досліджень;
- структурно-логічна послідовність викладення розділів чи підрозділів;

- формулювання власного відношення до задач, що розглядаються;
- переконливість аргументації та обґрунтованість висновків і пропозицій;
- відображення особистого внеску в розробку теми;
- внесення пропозицій і рекомендацій по впровадженню отриманих результатів у практику.

До того ж важливим є також формування у студента навичок написання і оформлення наукової рукописної роботи, які необхідні для майбутньої професійної діяльності випускника.

У дипломній роботі слід стисло та логічно викласти зміст та результати інженерної розробки, уникати загальних слів і бездоказових тверджень.

Необхідно обов'язково посилатися на джерела та їх авторів, з яких використано матеріали або окремі результати.

Працюючи над дипломною роботою, студент має засвоїти навички обґрунтування її актуальності, формулювання мети і завдань, визначення об'єкту і предмету, побудови комплексної схеми прикладного дослідження, роботи з різними джерелами інформації, аналізу та оцінки різноманітних аспектів функціонування об'єкта проектування, формулювання власних пропозицій, рекомендацій та висновків.

Результати, отримані в ході підготовки, виконання і захисту дипломної роботи, дозволяють оцінити ступінь готовності випускника до самостійної професійної діяльності.

Успішний захист дипломної роботи є підставою для присвоєння випускникові відповідного освітнього ступеня «бакалавр» із врученням йому диплома державного зразка (звичайного чи з відзнакою).

2 ТЕМАТИКА ТА ПОРЯДОК ПІДГОТОВКИ ДИПЛОМНОЇ РОБОТИ

Вибір теми дипломної роботи є початком процесу її виконання. Вона повинна бути актуальною, мати теоретичне та прикладне значення, відповідати сучасному рівню розвитку науки і техніки, напряму та спеціальності, вказувати на мету роботи. Назву слід формулювати коротко, без використання ускладненої термінології. Тематику ДР розробляють провідні викладачі профільних кафедр, і щорік коректують з погляду на науковий досвід, побажання фахівців, які беруть участь у рецензуванні робіт, і рекомендацій Екзаменаційної комісії (ЕК). Орієнтовна тематика дипломних робіт приведена в додатку Л.

Студент обирає тему ДР з переліку тем, що запропоновані керівником роботи, або пропонує свій варіант (за погодженням з керівником) з обґрунтуванням доцільності її розробки. Студент обирає «свою» актуальну тему дипломної роботи:

- з числа тем, які складають типовий перелік і потребують обов'язкового уточнення і конкретизації разом з науковим керівником, у залежності від мети, предмета і об'єкта досліджень та бази переддипломної практики;
- для продовження розвитку тематики, якою він займався в наукових студентських гуртках, під час виробничої практики, у курсових роботах та за власною ініціативою.

Обравши тему дипломної роботи, студент повинен упродовж першого тижня переддипломної практики порадитись зі співробітниками установи, на якій вона проводиться, щодо актуальності проблеми, запропонованої до розгляду, та забезпечення необхідною інформацією для подальших розрахунків.

У випадку, коли на розробку окремої теми в одній установі претендують кілька студентів (комплексна дипломна робота), закріплення і конкретизація її для окремих здобувачів проводиться спільно науковими керівниками або завідувачем кафедри з урахуванням можливостей виконання в умовах різних служб і відділів, знань і умінь здобувачів та необхідності забезпечення самостійності в їх роботі.

При цьому кожен здобувач відповідає за свою конкретну частину роботи і оформляє її окремо. Доцільно, щоб всі студенти, які брали участь у виконанні комплексної теми, захищали свої проекти на одному засіданні ЕК.

Всі дипломні роботи повинні ґрунтуватися на реальному матеріалі, зібраному та узагальненому в ході переддипломної практики.

Для опису дипломної роботи (в частині вимог) використовують три основні специфікації:

- визначення систем, або специфікація вимог користувачів;
- системних вимог;
- програмних вимог.

Специфікація вимог користувачів визначає високорівневі вимоги, для досягнення яких створюється програмна система. Принциповим моментом є те, що такий документ містить вимоги до системи з позицій прикладної галузі.

Специфікація системних вимог описує програмну систему в контексті системної інженерії. Зокрема, високорівневі вимоги до програмного забезпечення (ПЗ), що містить декілька або багато взаємозв'язаних підсистем і додатків. При цьому, система може бути як цілком програмною, так і містити програмні та апаратні компоненти.

Специфікація програмних вимог встановлює основні угоди між користувачами (замовниками) і розробниками (виконавцями) відносно того, що робитиме система і чого від неї не варто чекати. Цей документ може включати процедури перевірки створеного програмного забезпечення на відповідність вимогам, що пред'являються (аж до планів тестування), описи характеристик стосовно якості та методів його оцінювання, питань безпеки тощо.

Якщо студенти обирають комплексну дипломну роботу, то специфікація ПЗ може бути одна для всієї роботи, але з чітко визначеними розділами, за які відповідають окремі студенти.

Керівники ДР призначаються зі складу досвідчених науково-педагогічних працівників або співробітників науково-дослідного підрозділу випускової кафедри, а також провідних спеціалістів відповідної галузі з підприємств, установ, науково-дослідних інститутів тощо за їх згодою.

За пропозицією керівника ДР, у разі необхідності, кафедра може запропонувати консультантів із окремих розділів за рахунок ліміту часу, відведеного на керівництво дипломною роботою. Ними можуть бути науково-педагогічні працівники закладу вищої освіти за профілем розділу. Консультанти повинні надавати допомогу студенту в розробці відповідного розділу і перевіряти якість його виконання. Консультанти не мають права вирішувати питання про допуск чи не допуск студента до захисту ДР на засіданні ЕК.

Тема ДР подається студентом та його науковим керівником завідувачу кафедри. Назва теми повинна цілком характеризувати поставлену перед студентом загальну технічну або наукову задачу й містити конкретне завдання щодо об'єкту проектування.

Теми ДР розглядаються й ухвалюються на засіданні випускової кафедри і затверджуються рішенням Вченої ради факультету на початку останнього навчального року випускника упродовж вересня місяця за відповідною програмою підготовки. Юридично закріплення за студентом теми ДР та призначення керівника і рецензента здійснюється наказом по університету за поданням випускаючої кафедри і деканату факультету не пізніше двох місяців від початку передостаннього семестру навчання за відповідною програмою підготовки.

За необхідності затверджена тема ДР за погодженням з науковим керівником і поданням випускової кафедри може бути **уточнена** рішенням Вченої ради факультету **не пізніше, ніж за два місяці до призначеної дати захисту**. Після видання наказу студент разом з науковим керівником формулює завдання, розробляє розгорнутий план виконання ДР з визначеними термінами. Завдання до ДР затверджується завідувачем випускової кафедри і видається студенту впродовж двох тижнів після ухвалення теми роботи кафедрою.

У календарному плані випускової кафедри передбачається щосеместрове заслуховування проміжних результатів роботи та офіційний передзахист роботи за місяць до дати захисту. Разом із спорідненими кафедрами власного або іншого факультету та зовнішніми організаціями випусковою кафедрою визначаються рецензенти дипломних робіт, а також співробітники кафедри, які мають здійснювати рецензування на інших кафедрах, та подаються кандидатури рецензентів для затвердження декану факультету не пізніше, ніж за місяць до початку роботи ЕК.

Рецензенти призначаються зі складу досвідчених науково-педагогічних працівників або наукових співробітників університету та фахівців зовнішніх організацій, фахова кваліфікація яких відповідає спеціальності студента.

Не допускається призначення керівників ДР для взаємного рецензування дипломних робіт.

Завідувач кафедрою і викладачі надають методичну, консультаційну допомогу студенту-випускнику в період всього циклу підготовки дипломної роботи. Безпосереднє керівництво виконанням дипломної роботи здійснюється науковим керівником. Студент регулярно, не рідше одного разу на два тижні, зустрічається з керівником для консультування й інформування про стан виконання роботи згідно з календарним планом.

Передзахист роботи на кафедрі проводиться не пізніше, ніж за місяць до призначеної дати захисту. Впродовж одного тижня після успішного захисту на кафедрі студент подає оформлену згідно вимог роботу та відгук керівника роботи на кафедру, отримує **від завідувача кафедри направлення на рецензування** та передає роботу рецензенту. (Відгук на роботу керівник пише за 10 днів до захисту, після цього робота подається на рецензування.)

Рецензент повертає студенту прорецензовану роботу і **рецензію не пізніше, ніж за три робочі дні до захисту**, після чого студент ознайомлюється з рецензією, по можливості усуває вказані недоліки і повертає пакет документів на випускову кафедру.

3 ОРГАНІЗАЦІЯ ВИКОНАННЯ ДИПЛОМНОЇ РОБОТИ

Організація та контроль за ходом підготовки й захисту ДР покладається на завідувачів випускових кафедр. З метою надання студенту теоретичної та практичної допомоги у період підготовки і написання ДР, кафедра призначає йому наукового керівника. Як правило, ним є викладач кафедри, під керівництвом якого студент проходив переддипломну практику. Крім керівника ДР, в особливих випадках може **призначатися консультант**.

Студент має право: вносити свої пропозиції щодо обрання наукового керівника зі складу профілюючої кафедри, враховуючи власні напрацювання; клопотати перед випусковою кафедрою про заміну наукового керівника, якщо для цього є поважні причини.

Підготовка ДР повинна бути запланована у визначені календарні терміни. Упродовж першого місяця початку випускового навчального року та затвердження теми ДР дипломник представляє керівникові детально розроблений календарний план виконання дипломної роботи. План носить індивідуальний характер, тому в додатку Б наводиться типовий зразок (шаблон) завдання та календарний план виконання ДР.

Керівник ДР зобов'язаний:

- надати студенту методичну та практичну допомогу в обранні теми ДР;
- видати завдання на ДР та допомогти студенту в розробці плану її виконання;
- надати допомогу в обранні методики проведення досліджень;
- надати кваліфіковану консультацію щодо підбору літературних джерел і фактичних матеріалів, які необхідні для виконання роботи;
- здійснювати систематичний контроль за ходом виконання ДР студентом відповідно до затвердженого плану;
- провести попередній захист дипломної роботи з метою виявлення ступеня готовності студента до публічного захисту;

- після виконання дипломної роботи дати оцінку якості її виконання та відповідності вимогам, що ставляться до робіт такого типу (відгук керівника).

Здобувач повинен періодично (за спільною домовленістю, наприклад, не рідше ніж один раз у два тижні) інформувати наукового керівника про хід виконання ДР, консультуватися з питань, що викликають сумніви, обов'язково доводити до його відома інформацію про можливі відхилення від затвердженого графіка.

На першому етапі підготовки роботи науковий керівник радить, з чого розпочати вивчення теми, корегує план, дає рекомендації щодо літератури, яка повинна бути використана в роботі.

У ході виконання ДР керівник виступає як опонент, вказує здобувачу на недоліки аргументації, позиції, стилю тощо, радить, як краще їх усунути. Також, керівник ДР одночасно є і **нормоконтролером**, який перевіряє відповідність оформлення роботи вимогам методичних рекомендацій кафедри щодо написання дипломних робіт. У випадку виявлення грубих порушень та недоліків робота повертається студентові для їх усунення.

Однак студент повинен мати на увазі, що науковий керівник не є ні співавтором, ні редактором ДР і тому не повинен виправляти наявні в роботі методологічні, стилістичні, орфографічні та інші помилки.

Рекомендації та зауваження наукового керівника дипломник повинен сприймати творчо. Він може враховувати їх або відхиляти на свій розсуд, оскільки відповідальність за розробку й висвітлення теми, аргументовану якість змісту й оформлення дипломної роботи повністю покладається на нього.

Закінчена і підписана студентом ДР надається керівникові не пізніше, ніж за 14 днів до захисту, після чого керівник складає письмовий відгук.

4 ВИМОГИ ДО СТРУКТУРНИХ ЕЛЕМЕНТІВ ВСТУПНОЇ ЧАСТИНИ РОБОТИ

Дипломна робота повинна повністю розкривати зміст робіт, виконаних на кожному з етапів відповідно до методології розробки інформаційних систем та їх програмного забезпечення.

Назви розділів, крім стандартних компонентів, для яких не передбачена нумерація, рекомендується обирати відповідно до індивідуальних особливостей конкретної розробки. Зміст складових дипломної роботи має відповідати вимогам як логіки та зв'язності викладення у межах роботи в цілому, так і автономності компонентів, які дозволяють зрозуміти їхній зміст.

У дипломній роботі відповідно до її теми та завдань структура розділу може бути різною як за кількістю підрозділів, так і за їхньою спрямованістю. Можливе об'єднання підрозділів. Назви розділу та підрозділів рекомендується обирати такими, щоб вони відображали як їхню спрямованість, так і тематику конкретної розробки.

Пояснювальна записка ДР повинна містити:

- титульний лист;
- завдання на виконання дипломної роботи;
- анотацію;
- зміст;
- перелік умовних скорочень (за необхідністю);
- вступ;
- основну частину (три розділи);
- висновки;
- список використаних джерел;
- додатки (за необхідністю).

Крім того, до дипломної роботи додаються:

- письмовий відгук керівника дипломної роботи;

- письмова рецензія внутрішнього або зовнішнього рецензента (завірена печаткою установи, де працює рецензент);
- фотокопія опублікованої статті та/або тез конференції з тематики роботи з вихідними даними (за наявності);
- подання голові Екзаменаційної комісії (готується секретарем ЕК);
- електронна копія пояснювальної записки роботи на CD-диску у форматі .pdf з типовим ім'ям файлу: наприклад, Diplom_Stepanenko_2018.pdf.

Місцеположення додаткових документів – вклеєний конверти формату А5 (148x210) на жорсткій обкладинці ДР: на зворотній стороні першої обкладинки – з паперовими документами; у кінці дипломної роботи – з CD-диском.

Подана до захисту ДР є письмовим звітом, який відображує результати власних розробок, спостережень, теоретичних або експериментальних досліджень студента.

Загальний обсяг ДР визначається особливостями розробки, а також специфікою конкретної роботи (нормативи затверджуються методичними рекомендаціями кафедри). У загальній частині дипломної роботи значна частка припадає на аналіз об'єкта проектування або моделювання, розробку БД, алгоритмів та програмного комплексу.

Рекомендований об'єм пояснювальної записки – **60-80** сторінок формату А4 (без урахування завдання, анотації і додатків).

Отримані **нові наукові результати** і запропоновані нові технічні рішення ДР оформляються та **подаються для опублікування** у вигляді статей, тез конференцій та патентів на винаходи.

Текст роботи оформляється згідно зі стандартами (див. розділ 6).

4.1 Титульний лист

Титульний лист дипломної роботи оформлюється за встановленою формою (додаток А), на якому обов'язковою є наявність підписів студента та наукового керівника, а також завідувача кафедри.

Титульний лист дипломної роботи є першою сторінкою пояснювальної записки і містить такі обов'язкові елементи:

- Міністерство освіти і науки України;
- найменування вищого навчального закладу, де виконана робота;
- структурний підрозділ (найменування факультету та кафедри);
- шифр та назву спеціальності;
- тему дипломної роботи;
- прізвище, ініціали, шифр групи студента, назву та шифр спеціальності;
- прізвище, ініціали, науковий ступінь, вчене звання наукового керівника;
- прізвище, ініціали, науковий ступінь, вчене звання зовнішнього рецензента (за наявності);
- місто і рік виконання дипломної роботи.

Консультант вказується (після керівника), якщо він призначався.

4.2 Завдання на дипломну роботу

Завдання, до якого додається календарний план роботи студента за обраним напрямом дослідження (додаток Б), видається на початку виконання роботи після вибору теми та затвердження її наказом ректора університету, видається студенту керівником дипломної роботи та затверджується завідувачем кафедри.

Завдання на ДР друкується на сторінці формату А4 з обох сторін і не входить до загальної нумерації сторінок дипломної роботи, але безпосередньо знаходиться за титульним аркушем роботи.

4.3 Анотація

Анотація українською та англійською мовами призначена для ознайомлення зі змістом. Вона містить стислу інформацію про дипломну роботу, яка дозволяє прийняти рішення про доцільність знайомства з усією роботою. В анотації слід використовувати прості речення, стандартизовану термінологію, уникати складних граматичних зворотів маловідомих термінів і символів. Текст анотації на пункти не поділяють.

Обсяг анотації не повинен перевищувати однієї сторінки формату А4. Ключові слова, що є визначальними для розкриття сутності роботи, наводять після тексту анотації. Перелік ключових слів повинен включати від 5 до 10 слів (словосполучень) у називному відмінку. Перелік подається в рядок через кому великими буквами. Перший рядок – з абзацного відступу, вирівнювання «за шириною».

Ключовим словом називається слово або стійке словосполучення з тексту документу, яке з точки зору інформаційного пошуку несе смислове навантаження.

Міжрядковий інтервал анотації – 1. Зразок анотації наведений в додатку В.

4.4 Зміст дипломної роботи

Зміст розташовують безпосередньо після анотації, починаючи з нової сторінки за номером 2 (**сторінки завдання і анотації в нумерацію пояснювальної записки не входять**), і оформляють тим самим шрифтом, що і текст документу. У змісті відображаються структурні елементи із номерами сторінок, розташовані один під одним (вирівняні по ширині листа). Найменування елементів відокремлюють від номерів сторінок крапками.

До змісту включають: перелік умовних скорочень (якщо він має місце); вступ; послідовно перелічені назви всіх розділів, підрозділів, пунктів і підпунктів (якщо вони мають заголовки); висновки за розділами без нумерації; загальні висновки; список використаних джерел; назви додатків (якщо вони є з відповідними номерами сторінок).

Назви розділів і підрозділів мають бути стислими, зрозумілими, грамотно сформульованими та тісно пов'язаними з темою та змістом роботи.

Назви структурних елементів (умовні скорочення, вступ, розділи, загальні висновки, список використаних джерел, назви додатків) відносяться до стилю заголовків 1-го рівня, під ними з відступом вправо на два символи розташовуються заголовки 2-го рівня (назви підрозділів, висновки до розділів), а за ними заголовки 3-го рівня (пункти).

Зразком змісту ДР можна вважати зміст даного посібника.

4.5 Перелік умовних скорочень

Якщо у ДР використовуються маловідомі скорочення, нові символи, позначення, то їх перелік має бути подано в роботі у вигляді окремого списку, який розміщують перед вступом. Незалежно від цього за першої появи цих елементів у тексті роботи наводять їх розшифрування.

Перелік умовних скорочень треба друкувати двома колонками, в яких зліва за абеткою наводять скорочення, справа – їх детальне розшифрування.

Якщо у роботі спеціальні терміни, скорочення, символи, позначення повторюються менше трьох разів, перелік не складають, а їх розшифрування наводять у тексті при першому згадуванні.

5 ВИМОГИ ДО СТРУКТУРНИХ ЕЛЕМЕНТІВ ОСНОВНОЇ ЧАСТИНИ

ВСТУП

Вступ – це частина ДР, яка дає загальне уявлення про конкретний напрям наукового пошуку й містить необхідні кваліфікаційні характеристики проведеної прикладної розробки.

У вступі необхідно обґрунтувати актуальність дослідження проблеми, що розглядається; визначити об'єкт, предмет, мету та завдання розробки; подати характеристику структури дипломної роботи.

Текст вступу має бути чітким, стислим та інформативним. Рекомендований обсяг – до 3-4 листів. Вступ остаточно формулюється на завершальному етапі оформлення дипломної роботи.

Актуальність теми. Актуальність – обов'язкова вимога до будь-якої практичної або наукової роботи. Тому цілком зрозуміло, що вступ до неї починають з обґрунтування актуальності обраної теми для розвитку сучасної теорії і практики певної галузі науки. Актуальність теми та доцільність даної розробки для розвитку галузі знань 12 «Інформаційні технології» обґрунтовують шляхом критичного аналізу та порівняння з вже відомими розв'язаними проблемами.

Обґрунтування актуальності теми має містити положення, які розкривають її важливість та необхідність вирішення саме зараз. Чіткого критерію встановлення ступеня актуальності немає. При розгляді прикладних розробок звичайно оцінюється можливість досягнення найвищої ефективності, яка може забезпечуватися оптимізацією функціональних можливостей, розширенням сфери застосування, поліпшенням експлуатаційних характеристик тощо. Актуальною також є тематика, яка сприяє розвитку наукових знань. Орієнтовна схема обґрунтування актуальності проблеми дослідження може бути такою:

- дати об'єктивну оцінку сучасного стану об'єкта розробки з акцентуванням уваги на інших проблемах і суперечностях;

- показати, як на сучасному етапі розв’язуються існуючі проблеми на практиці (з посиланням на офіційні документи);
- оцінити стан обраної проблеми: на основі стислого аналізу літературних джерел визначити коло науковців, які займалися її розробкою, висвітлити проблематику їх досліджень і наголосити на маловивчених або зовсім невивчених моментах;
- зробити висновки про те, що нерозробленість конкретного аспекту проблеми зумовила вибір теми розробки.

Мета і задачі розробки. У вступі також зазначається кінцевий результат, на досягнення якого спрямована робота (мета роботи), та підпорядковані йому задачі, які повинні бути послідовно виконані для досягнення мети. Мета розробки – це бажаний кінцевий результат, відповідь на питання: «Чого потрібно досягти в результаті виконання дипломної роботи?».

Мета роботи формулюється шляхом представлення очікуваних результатів в узагальненому вигляді та повинна перетинатися з назвою дипломної роботи, а задачі – з назвами розділів, параграфів. При написанні мети рекомендується використовувати такі дієслова, як «визначити», «виявити», «обґрунтувати», «розробити» тощо. Керуючись метою роботи, визначають основні задачі розробки.

Задачі розробки конкретизують її мету і дають уявлення про те, в яких напрямках повинна вестися робота. Задачі повинні бути взаємопов’язані. Одні з них можуть бути вирішені на підставі аналізу літератури чи інтернет-джерел, інші – у результаті проектної діяльності.

Формулювання задач необхідно робити якомога більш ретельно, оскільки опис їхнього розв’язання має відобразити зміст складових частин роботи. У подальшому, при написанні висновків доцільно їх будувати з точки зору досягнення мети і виконання поставлених задач.

Мета повинна бути сформульована таким чином, щоб указувати на об’єкт і предмет розробки.

Об’єкт і предмет розробки. Обов’язковим елементом вступу є визначення об’єкта і предмета розробки.

Об'єктом розробки є процес або явище, що породжує проблемну ситуацію і обране для розробки або для вивчення. Предмет розробки (дослідження) міститься в межах об'єкта – це вузька частина об'єкта розробки (дослідження), проблема (коло питань), що розробляються в роботі. Об'єкт і предмет повинні завжди бути взаємозв'язаними як загальне і конкретне, як система та її компоненти. Об'єкт завжди є загальною сферою наукового пошуку, а предмет – тим конкретним, що виявляється у певному дослідженні (окремі його аспекти, сегменти). Саме на предмет спрямована основна увага дипломника, оскільки предмет дослідження визначає тему ДР.

Практичне значення одержаних результатів (за наявності). Важливу частину вступу займає практична значущість, що показує теоретичне та практичне значення одержаних результатів. Практична значущість дипломної роботи показує, що результати розробки, отримані студентом, можуть бути використані на конкретному підприємстві, у виробництві, в освітньому процесі тощо.

Якщо у роботі є дослідницька частина, то коротко приводиться обґрунтування актуальності (важливості) дослідження для розвитку відповідної галузі науки чи виробництва, а також мета і завдання дослідження.

Студент має коротко викласти нові наукові положення (рішення), запропоновані особисто. Необхідно показати відмінність отриманих результатів від відомих раніше, описати ступінь новизни (уперше отримано, удосконалено, дістало подальший розвиток), якщо вони є в роботі.

Враховуючи те, що ДР на здобуття освітнього ступеня «бакалавр» – це передусім робота навчально-дослідницького характеру, висувати високі вимоги до рівня наукової новизни та її результатів недоцільно. Тому, для рівня її результатів достатньо наявності окремих елементів новизни. Але автор як науковець-початківець повинен показати уміння правильно визначити вид і рівень новизни одержаного результату і конкретно, не допускаючи узагальнених тверджень, сформулювати положення відповідно до перелічених вимог і змісту проведеного дослідження.

Відомості про впровадження результатів розробки необхідно подавати із зазначенням найменувань організацій, в яких здійснено впровадження, форм реалізації та реквізитів відповідних документів.

Апробація роботи (*за наявності*). Вказується, на яких наукових конференціях, симпозіумах, семінарах, школах оприлюднені результати розробки (досліджень). Необхідно вказати статус, повну назву, місто та дати проведення наукової конференції, симпозіуму, семінару тощо.

Якщо матеріали опубліковано, теми доповіді (повідомлення) вказувати не потрібно. Якщо матеріали не опубліковано, але є програми або інші документи, що підтверджують апробацію, необхідно також вказати теми доповіді (повідомлення).

Якщо певна організація прийняла до використання (впровадження) рекомендації або розробки студента, необхідно вказати назву цієї організації з зазначенням реквізитів відповідного документа, що підтверджує впровадження.

Публікації (*за наявності*). Зазначається, у скількох статтях наукових журналах та збірниках наукових праць (із зазначенням їх належності до числа фахових чи нефахових), а також у скількох матеріалах (тезах) наукових конференцій, симпозіумів, семінарів опубліковано результати ДР.

Структура та обсяг роботи. У цьому пункті треба вказати, що робота складається зі вступу, кількості розділів, висновків, списку використаних джерел, додатків. Вказати загальний обсяг роботи (сторінок), кількість рисунків і таблиць, а також кількість найменувань у списку використаних джерел.

Наприклад:

Дипломна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел (25 найменувань), 5 додатків. Загальний обсяг роботи становить 75 сторінок основного тексту, 17 рисунків та 8 таблиць.

Приклад вступу до дипломної роботи наведений в додатку М.

1 ОГЛЯД ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ АНАЛОГІВ ОБ'ЄКТА РОЗРОБКИ

У назві розділу текст «об'єкт розробки», як правило, не застосовується. Назва цього елемента зазвичай збігається з темою роботи.

У першому розділі на підставі огляду літератури та інтернет-джерел за темою повинні бути викладені стан наукової проблеми (задачі), теоретичні основи проблеми, розкриті її окремі теоретико-методологічні аспекти, актуальність і важливість її розробки на даному етапі розвитку.

Стисло, критично висвітлюючи роботи попередників, студент повинен визначити ті питання, що залишились невирішеними, а, отже, визначити своє місце в розв'язанні проблеми.

Літературу, що підлягає вивченню, у загальному випадку можна розділити на такі основні групи:

- монографії, підручники, навчальні посібники;
- статті, опубліковані у збірниках, журналах та інших періодичних виданнях;
- словники, дисертаційні роботи та інтернет-ресурси.

У цьому розділі може бути три-чотири підрозділи, що містять аналіз та оцінку конкретних сфер діяльності об'єкта розробки, результати теоретично-аналітичного інформаційного пошуку.

Основні підрозділи – теоретико-методичний та аналітичний – повинні бути взаємопов'язані, а матеріал викладено послідовно і логічно з критичним аналізом теоретичних положень, статистичних даних та іншої інформації.

У першому підрозділі розглядаються теоретичні аспекти предметної області розробки, аналізується фактичний стан проблеми на основі матеріалів, які характеризують об'єкт проектування. Приводиться теоретичний виклад важливих аспектів проблеми та критичний огляд наукової літератури і періодичних джерел з визначеної тематики; розглядаються різні підходи до вирішення проблеми, дається їх оцінка.

Теоретична частина має бути підсумована думками про сучасний стан проблеми, рівень розробленості поглядів на дане явище як сучасних, так і тих, що склалися історично.

Аналітичний (*аналітика* від грецької *ἀναλυτικά* – мистецтво аналізу) підрозділ повинен бути присвячений розгляду аналітичних аспектів досліджуваної проблеми. У цьому підрозділі повинен бути здійснений аналіз предметної області дослідження, результати якого дозволяють виділити і обґрунтувати існуючу проблему, вирішення якої буде актуально в галузі інформаційних технологій.

Зокрема в аналітичній частині повинні висвітлюватися такі питання:

- призначення та область застосування об'єкта проектування;
- огляд способів і засобів вирішення поставлених задач;
- аналіз переваг і недоліків існуючих методів систем-аналогів (методів).

Розгляд перелічених питань спрямований на дослідження предметної області, аналізу існуючих рішень та досвіду розв'язання аналогічних задач, методів та засобів їхньої реалізації.

Розділ має бути достатньо насиченим фактографічною інформацією, що відображає стан об'єкта розробки за певний період.

Даний розділ необхідно закінчити підрозділом **«Постановки задачі на розробку»** – коротке резюме стосовно доцільності проведення розробки, яке включає обґрунтування вибору об'єкта і предмета розробки, формулювання мети та задач розробки.

У постановці задачі на розробку уточняється, деталізується та конкретизується завдання і кінцева мета дипломної роботи, що є підставою для подальших етапів розробки.

Розділ закінчується підрозділом – **Висновки до розділу** (без нумерації підрозділу) обсягом 0,5-1 с.

Загальний обсяг першого розділу, як правило, не повинен перевищувати 20% обсягу основної частини ДР (**20-25** сторінок).

2 АНАЛІЗ І ПРОЕКТУВАННЯ СИСТЕМИ

Американський програміст і технічний письменник Алан Купер, широко відомий як «Батько Visual Basic», у своїй книзі «Психіатрична лікарня в руках пацієнтів» сказав: *програмуванню повинне передувати проектування* [8]. Словами авторів підручника можна сказати і так: *перш ніж програмувати необхідно навчитися проектувати*.

Метод об'єктно-орієнтованого моделювання передбачає послідовне виконання двох етапів: об'єктно-орієнтованого аналізу та об'єктно-орієнтованого проектування. Тому термін «об'єктно-орієнтоване моделювання» еквівалентний терміну «об'єктно-орієнтований аналіз і проектування».

Етап *аналізу* полягає в дослідженні системних вимог і проблеми, а не в пошуках шляхів її рішення. Наприклад, при розробці нової інформаційної системи (ІС) для автоматизації торгівлі необхідно описати способи її використання і основні функції. Аналіз – це досить широке поняття, зміст якого точніше передають терміни *аналіз вимог* (тобто аналіз вимог до системи) і *об'єктно-орієнтований аналіз* (дослідження об'єктів предметної області).

При цьому під предметною областю розуміють ту частину реального світу, що має істотне значення чи безпосереднє відношення до процесу функціонування програми. Іншими словами: предметна область містить у собі тільки ті об'єкти і взаємозв'язки між ними, які необхідні для опису вимог і умов розв'язання деякої задачі.

У загальному випадку виділення базових об'єктів (чи компонентів) предметної області є нетривіальною задачею. Складність виявляється у неформальному характері процедур чи правил, які можна застосовувати з цією метою. Окрім того, таку роботу необхідно виконувати спільно з фахівцями чи експертами, що добре знають предметну область.

У процесі *проектування* основна увага приділяється концептуальному рішення (у вигляді ПЗ або апаратних засобів), що забезпечує виконання основних вимог, але не питанням його реалізації. Наприклад, на етапі

проектування описуються програмні об'єкти або схема бази даних. Ідеї проектування виключають низькорівневі або «очевидні» деталі з точки зору потенційного користувача.

Під час аналізу системних вимог необхідно з'ясувати потреби замовника, аналізуючи отриману інформацію від керівництва компанії та майбутніх користувачів системи. Під час аналізу необхідно визначити:

- функціональні вимоги до системи (або бізнес-процеси), тобто встановити варіанти використання програмної системи для реалізацій конкретних функцій чи дій у даній предметній області («визначити те, що система має робити»);
- потоки даних для кожного бізнес-процесу;
- границі системи;
- користувачів системи та процеси їхньої взаємодії з системою.

У результаті аналізу оформляють словник (або глосарій) предметної області (містить текстовий опис термінів, сутностей, користувачів тощо) і технічне завдання, у якому сформульовано функціональні та нефункціональні вимоги до системи. До нефункціональних вимог відносяться питання надійності, зручності використання, продуктивності, можливості супроводу програм, питання безпеки, проектні та апаратні обмеження тощо.

Технічне завдання є гарантією єдиного трактування вимог замовниками і проектувальниками. Воно дає змогу також вирішувати спірні питання з приводу функцій системи, що виникають у процесі її створення.

Згідно концепції розробки архітектури програмних додатків, керованих моделями, розробка ІС повинна розпочинатися зі створення платформи-незалежної моделі (*Platform Independent Model, PIM-модель*), яка визначає склад, структуру і поведінку майбутнього програмного додатку. PIM-модель представляє собою сукупність архітектурних елементів проектованої системи і зв'язків між ними на досить абстрактному рівні, тобто без прив'язки до конкретних мов програмування, технологій реалізації компонентів і ОС [5].

На подальших етапах розробки ІС необхідно адаптувати РІМ-модель до конкретних платформ розробки і технологій реалізації програмних компонентів. У результаті цього можуть бути створені одна або декілька платфрмо-залежних моделей (*Platform Specific Model, PSM-модель*).

Процес розробки програмних додатків з використанням концепції моделей включає декілька етапів (рис. 2.1) [5].

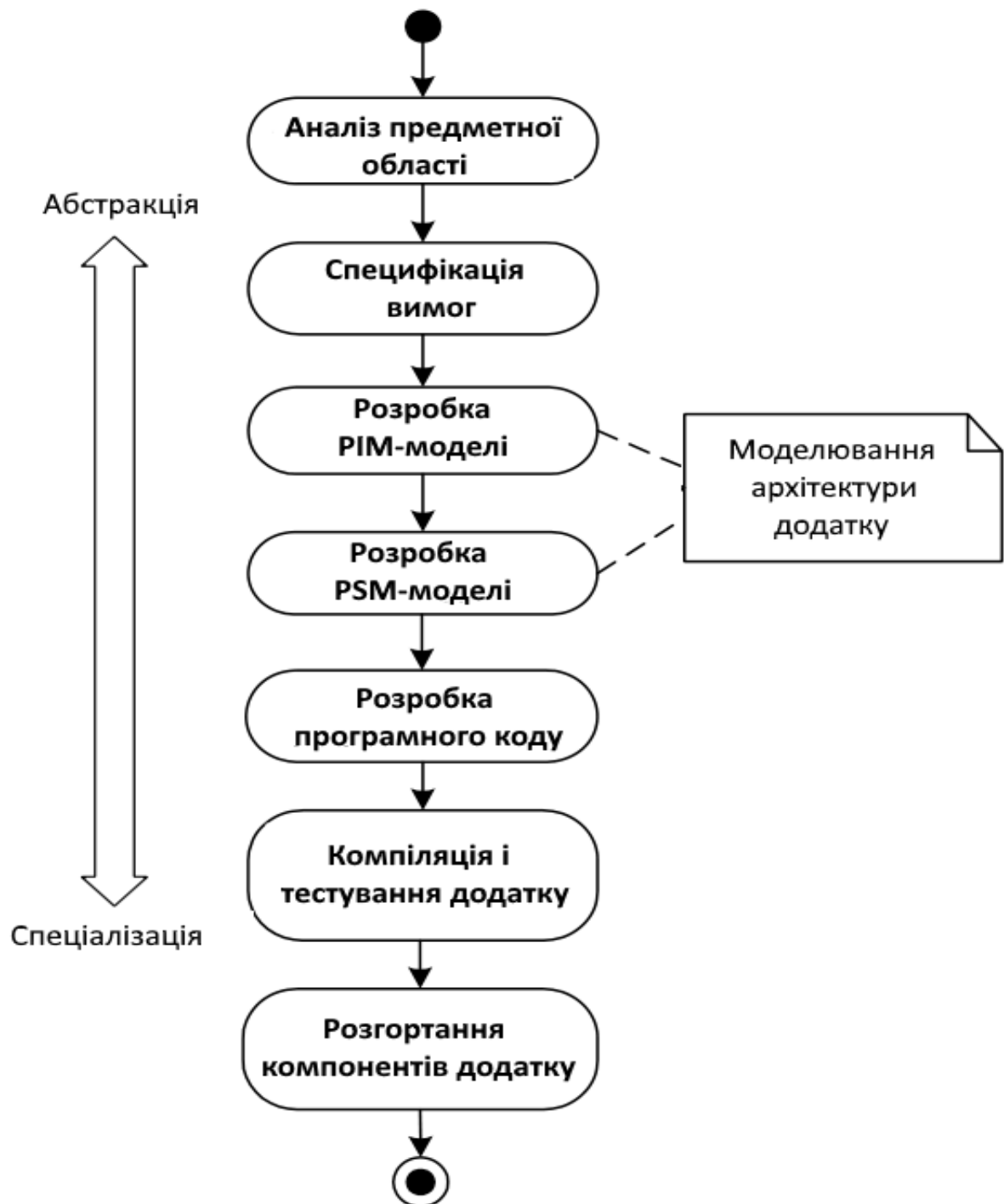


Рисунок 2.1 – Процес розробки програмних додатків з використанням концепції моделей

На початку розробки додатку виконуються такі послідовні етапи:

- аналіз предметної області й специфікація вимог до проекрованої ІС;
- розробляється детальна PIM-модель, яка повністю абстрагується від особливостей конкретних платформ реалізації;
- створюється одна або декілька PSM-моделей, які служать основою для створення або генерації програмного коду додатку;
- компіляція, збірка та тестування додатку;
- розгортання компонентів додатку.

Об'єктно-орієнтований аналіз і проектування (ООАП) – це підхід до розв'язання задач з використанням моделей, заснованих на поняттях реального світу [9]. Для моделювання етапів аналізу вимог і об'єктного проектування ІС застосовують уніфіковану мову моделювання (Unified Modeling Language – UML), основна ідея якої – це можливість моделювати програмне забезпечення й інші системи як набори взаємодіючих об'єктів.

Проектна частина дипломної роботи повинна містити опис процесів проектування ІС відповідно до її призначення. При застосуванні технологій об'єктно-орієнтованого проектування процес складається з таких етапів:

1. Моделювання предметної області.
2. Аналіз вимог – описує процес дослідження вимог до системи.
3. Проектування логічної структури системи – описує, які типи об'єктів важливі для моделювання системи і як вони взаємозв'язані.
4. Проектування фізичної структури системи – реалізація елементів логічного представлення в конкретні матеріальні сутності (підсистеми, компоненти) та їх зв'язки.
5. Розгортання програмної системи на апаратних засобах – розподіл елементів (результатів розробки) програмного додатку за апаратними вузлами комп'ютерної системи.
6. Динамічна поведінка системи – описує життєві цикли цих об'єктів і те, як вони взаємодіють один з одним для забезпечення необхідної функціональності системи.

Таким чином, конструкції мови UML дозволяють моделювати статичну структуру і динамічну поведінку системи, яка представлена у вигляді взаємодіючих об'єктів (програмних модулів), що реагують на зовнішні події. Дії об'єктів дозволяють виконати певні задачі або отримати клієнтам (користувачам) системи деякі конкретні результати.

У даному посібнику описуються основні елементи нотації діаграм UML, на конкретному прикладі розглядається процес візуального моделювання ІС із застосуванням чудового і безкоштовного CASE-інструменту StarUML.

Цей розділ (20-25 сторінок) закінчується висновками до розділу обсягом 0,5-1 сторінок.

2.1 Моделювання предметної області

Необхідність аналізу предметної області до початку написання програми була усвідомлена давно при розробці масштабних проектів. При цьому предметна область включає тільки ті об'єкти і взаємозв'язки між ними, які потрібні для опису вимог і умов розв'язання деякої задачі. Сам процес виділення або ідентифікації компонентів предметної області називається *концептуалізацією* предметної області.

Тобто, щоб створити якісну ІС, не досить зрозуміти бізнес-процеси і потреби замовника. Важливо зрозуміти, якою саме інформацією система повинна управляти. А для цього треба знати, які об'єкти потрапляють у предметну область ІС і які логічні зв'язки між ними існують.

Метою побудови моделі предметної області є отримання графічного представлення логічної структури досліджуваної предметної області. Логічна модель предметної області ілюструє сутності у формі класів, а також їх взаємовідносини між собою.

Побудова моделі предметної області розпочинається з виявлення абстракцій, існуючих у реальному світі, тобто тих основних концептуальних об'єктів, які зустрічаються в системі.

Якщо використати ітеративний процес розробки ПЗ (наприклад, Rational Unified Process – RUP або дуже компактну методологію екстремального програмування – eXtreme, XP-процес), то кожній ітерації відповідатиме своя модель предметної області, що відображає сценарії прецедентів, які реалізуються на цьому етапі. Таким чином, модель предметної області еволюціонує у процесі розробки системи.

2.1.1 Поняття предметної області

Модель предметної області – це найважливіша модель об’єктно-орієнтованого аналізу, яка відображає візуальне подання концептуальних класів або об’єктів реального світу в термінах предметної області, а не програмні компоненти. Такі моделі також називають концептуальними моделями або моделями об’єктів предметної області. Це свого роду словник основних абстракцій, тобто найважливіших іменників у просторі задачі.

Іменники, які описують поняття з предметної області, називають **доменними об’єктами**. На самому початку аналізу і проектування необхідно створити модель предметної області, в якій всі доменні об’єкти будуть зображені на одній великій діаграмі класів [7, розділ 10.1].

У термінології UML модель предметної області – це **діаграма класів**. Зазвичай в цій моделі опускається велика частина деталей, зокрема атрибути та операції (методи) класів. Ось чому можна вважати, що модель предметної області є зведеною діаграмою класів, як візуальний словник важливих абстракцій або візуальний словник предметної області.

Словник предметної області – часто згадуване поняття об’єктно-орієнтованого аналізу і проектування. Усі теоретики об’єктно-орієнтованого підходу одностайно стверджують, що словник термінів предметної області – найважливіший артефакт, що лежить в основі аналізу і проектування, яким автори прецедентів можуть користуватися на більш пізніх етапах.

2.1.2 Основні елементи моделювання предметної області

У об'єктно-орієнтованих системах структура коду визначається класами (узагальненими об'єктами). Класифікація означає, що об'єкти з однаковими структурами даних (атрибутами) і поведінкою (операціями) групуються у класи. Тому, перш ніж приступати до кодування, треба з'ясувати, які класи знадобляться. З цією метою слід намалювати одну або декілька діаграм класів. Насправді, це перший крок до отримання справжньої діаграми класів, при якій потрібно представити систему в цілому. Потім у процесі роботи над прецедентами поступово уточнюються діаграми і врешті-решт буде отримано детальну статичну модель системи.

Модель предметної області повинна бути побудована до складання варіантів використання, щоб уникнути дублювання понять. Якщо схожі назви потраплять у тексти варіантів використання, то знайти їх там набагато складніше, ніж у словнику проекту. Модель предметної області, яка є діаграмою класів аналітичного рівня, – перше наближення до статичної структури системи.

Вимоги до системи зазвичай виглядають як фрази, побудовані у вигляді тверджень: система повинна робити те і не повинна робити це. Тобто, кращими джерелами класів, мабуть, є високорівневий опис завдання, низькорівневі вимоги, описи прецедентів, і експертна оцінка завдання. У міру уточнення даного переліку необхідно враховувати, що:

- іменники та іменні групи стають об'єктами і атрибутами;
- дієслова і дієслівні групи стають операціями і асоціаціями;
- родовий відмінок показує, що іменник має бути атрибутом, а не об'єктом.

Для моделювання предметної області та при проектуванні системи в цілому будемо використовувати мову UML. Для діаграм мови UML існують три типи візуальних позначень: *графічні символи, зв'язки (єднальні елементи), текст* (рис. 2.2, 2.3).





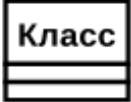

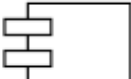





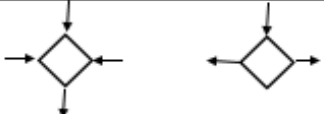
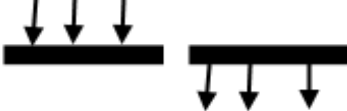
Графічний символ	Назва елемента
	дійова особа (актор)
	варіант використання
	коментар
	пакет
	клас
	об'єкт
	КОМПОНЕНТ
	вузол
	діяльність
	стан
	лінія життя, фокус управління
	початковий і кінцевий стан
	логічний з'єднувач, розгалужувач
	паралельний з'єднувач, розгалужувач

Рисунок 2.2 – Основні графічні символи UML


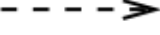

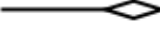






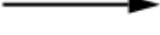


Графічний символ	Назва елемента
	відношення асоціації
	відношення залежності
	відношення успадкування
	агрегація
	композиція
	приймач події
	джерело події
	заміна
	межа
	простий потік управління
	послати
	виклик
	рекурсія
	повернути

Рисунок 2.3 – Єднальні елементи UML

Більшість розробників при створенні ПЗ задіють лише невелику підмножину мови UML. Необхідно знайти свою підмножину, яка б підходила для розв'язання поставленої задачі.

При проектуванні ПЗ дипломної роботи необхідно використовувати основні діаграми мови UML для моделювання предметної області, формування та аналізу вимог до системи, логічної і фізичної структури та поведінки системи (рис. 2.4).

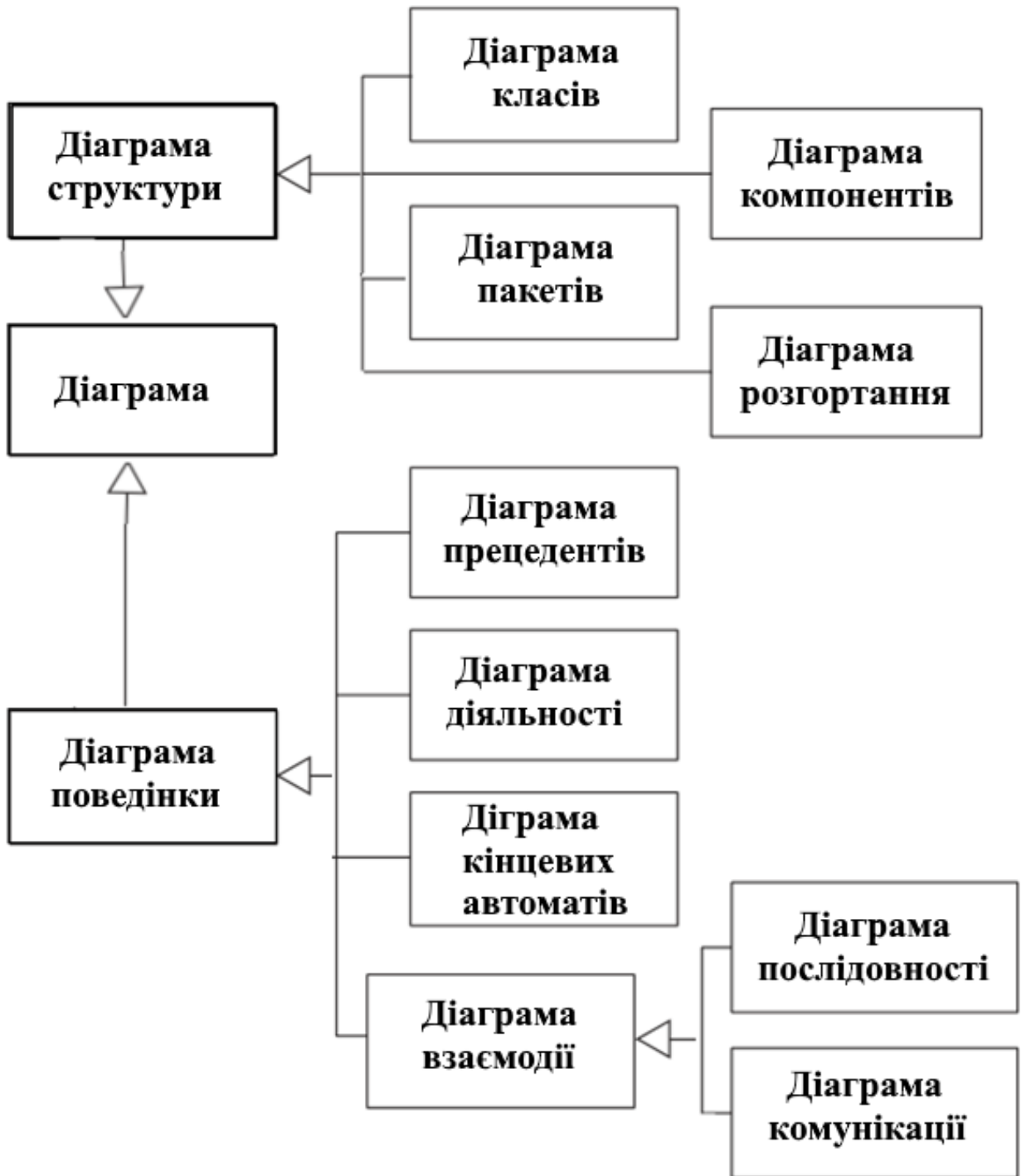


Рисунок 2.4 – Основні діаграми UML

У ході виявлення об'єктів з предметної області необхідно встановити, які зв'язки існують між ними. Особливий інтерес представляють два види відношень: *узагальнення* (відношення між підкласом і суперкласом, відношення виду « ϵ », $\langle | \text{---}$) й *агрегація* (відношення між цілим і частиною, відношення виду «*має*», $\diamond \text{---}$). Між класами предметної області можуть існувати й інші відношення, у тому числі прості *асоціації* (---), але ці два відношення дуже важливі. Класи, що відображають модель предметної області, потім увійдуть до основи статичної моделі діаграми класів.

У результаті модель предметної області, доповнена асоціаціями (статичними відношеннями) між парами класів, повинна адекватно описувати аспекти завдання, не залежні від часу (статичні), і в цьому нагадувати діаграми сутність-зв'язок, що використовуються в моделюванні даних.

Необхідно також відвести фіксований час на побудову початкової моделі предметної області. Не потрібно домагатися її досконалості, просто необхідно її закінчити швидше, а потім модифікувати в міру просування вперед.

Класи в UML – це місце для розміщення атрибутів (даних-членів) і операцій (функцій, що виконуються об'єктами). Проте, починаючи моделювати предметну область, не потрібно витратити багато часу на ідентифікацію атрибутів і операцій; це потрібно буде робити при уточненні і наповненні статичної частини моделі. Зараз же слід сконцентрувати увагу на виявленні власне об'єктів і відношень між ними.

2.1.3 Визначення робочої області моделювання

Як приклад робочої області моделювання розглянемо internet-магазин з продажу дитячого і дорослого одягу різних брендів. Покупець (клієнт) переглядає каталог і робить замовлення. Потенційний клієнт заходить на сайт магазину, він може натиснути кнопку перегляду (або завантаження) каталогу, далі може покласти відібраний товар в кошик, змінити кошик і, прийнявши рішення про купівлю товарів, перейти з кошику до оформлення замовлення [10].

Для того щоб коректно створити систему, що відповідає усім вимогам замовника, необхідно абсолютно чітко уявити собі її основні бізнес-функції і з'ясувати вимоги, що пред'являються до системи. Для цього необхідно провести обстеження компанії і побудувати її повну бізнес-модель. Оскільки наш приклад є придуманим, то не можливо провести таке обстеження і не маємо можливості спілкуватися із замовником, тому спиратися необхідно на придуманий словесний опис системи.

Кожен товар в каталозі описується артикулом, розмірним рядом, ціною і фото з коротким описом.

Покупець може завантажити каталог товарів. Каталог складається з набору товарів з фото, ціною і розмірами. Покупець складає вподобані товари в кошик, при цьому вибираючи розмір і кількість необхідного товару цього артикулу.

Кошик можна змінити: проглянути, видалити товар, змінити кількість позицій одного артикулу, повернутися в каталог.

Коли покупець робить замовлення, він вводить свої особисті дані, телефон і оплачує його по банківській картці або готівкою.

Після того як зроблено замовлення, його можна забрати із складу через один робочий день. Дані про замовлення поступають співробітникам магазину (співробітник відділу продаж), він перевіряє наявність товарів і передає його комірникові на комплектацію. Комірник, зібравши замовлення, робить відмітку про готовність.

Замовлення видається зі складу комірником іншою системою, що буде називатися Склад. Комірник видає замовлення і відмічає в системі, що замовлення видане.

Оскільки від створення до видачі замовлення проходить різні стадії, то введемо поняття статусу замовлення. Співробітники магазину можуть змінювати статус замовлення, а покупець може простежити за формуванням замовлення. У такому разі дана система надає ще одну функцію: узнати статус замовлення.

На рис. 2.5 наведена повна модель предметної області для internet-магазину. Ті прецеденти і класи, які вибрані із вимог до системи, і які будуть

зустрічатися пізніше, покликані задовольнити вимоги замовника (власника магазину). Для відображення класів у діаграмі, окрім стандартного позначення класу (прямокутник, розділений на три частини: назва класу, дані-атрибути і операції представлені пустими клітинами), можна використати й інші варіанти відображення класів аналізу [7, розділ 10.1].

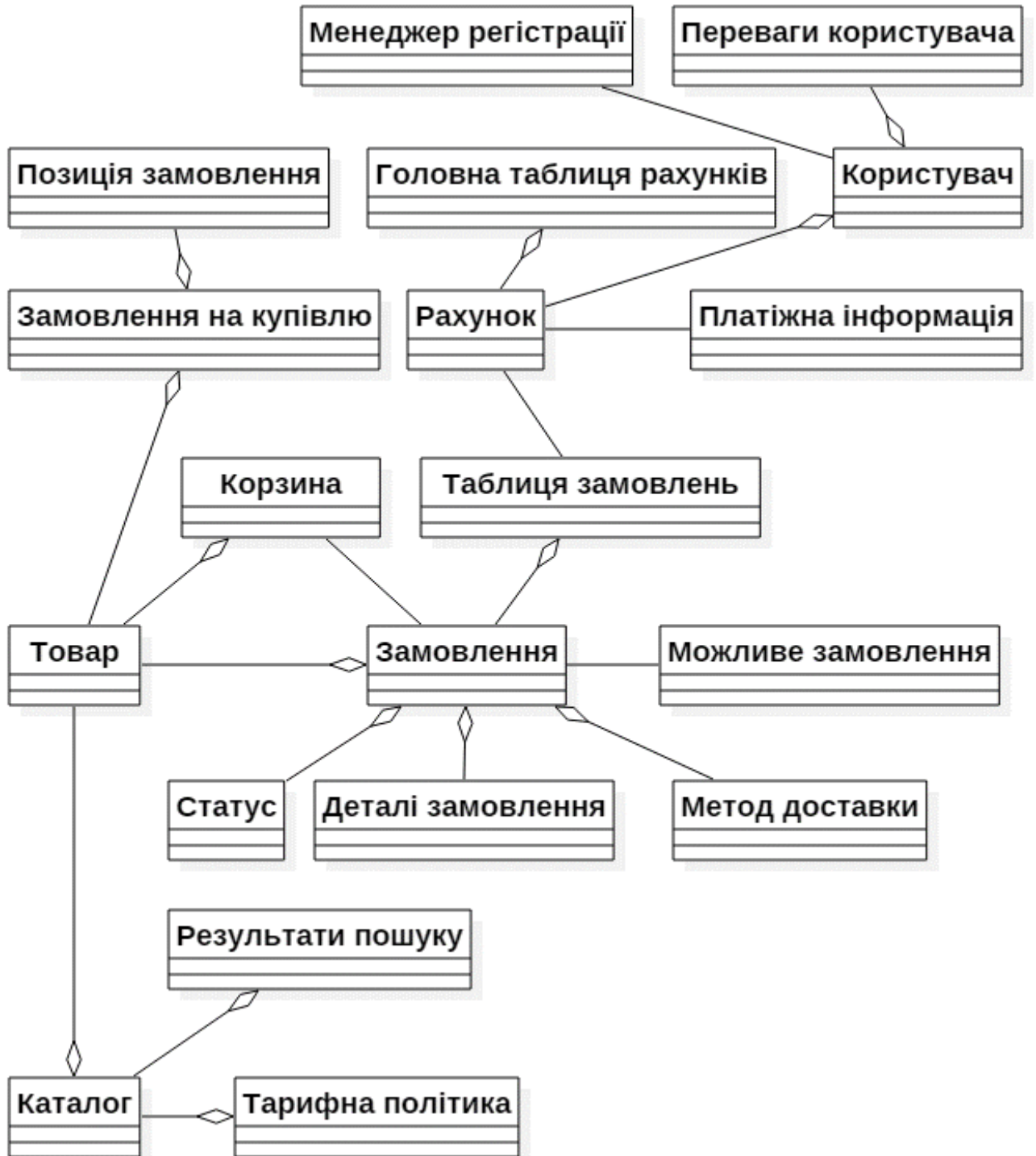


Рисунок 2.5 – Модель предметної області internet-магазину

2.2 Формування та аналіз вимог

Для побудови чогось, передусім, необхідно зрозуміти, чим *це повинно бути і що робить система*. Тому процес розуміння та його документування називається *аналізом вимог*.

Вимога – це умова чи можливість, якій повинна відповідати система. Основна задача етапу визначення вимог – знайти, обговорити і зафіксувати, що дійсно вимагається від системи у зрозумілій формі як клієнтам, так і членам команди розробників. Результатом аналізу вимог є документ, який називають *специфікацією вимог*, або *специфікацією вимог до ПЗ*.

Для моделювання і опису усіх процесів, що протікають під час розробки додатку, застосовують уніфіковану мову моделювання (UML). Усі діаграми можна умовно розділити на поведінкові й структурні.

Поведінкові діаграми відображають процеси, що протікають у модельованому середовищі. **Структурні діаграми** відображають елементи, з яких складається система. При цьому одні й ті ж типи діаграм можуть використовуватися як для моделювання бізнес-процесів, так і для безпосереднього проектування архітектури.

Кількість діаграм для моделі конкретного додатку не є строго фіксованою. Це означає, що для простих додатків немає необхідності будувати всі без виключення типи канонічних діаграм. Деякі з них можуть бути просто відсутніми у проекті системи, і цей факт не вважатиметься помилкою розробника. Наприклад, модель системи може не містити діаграму розгортання для додатка, який виконуватиметься локально на комп'ютері користувача. Тобто, склад діаграм залежить від специфіки конкретного проекту системи.

Кожна діаграма в нотації мови UML має область змісту для зображення графічних вузлів і шляхів між ними, які є власне елементами моделі в нотації UML. Додатково діаграма може бути розміщена у фрейм (прямокутну рамку) і мати заголовок, як це зображено на рис. 2.6.

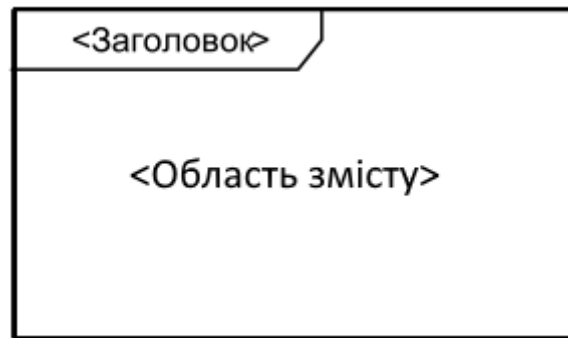


Рисунок 2.6 – Зображення діаграми мови UML 2.0 у вигляді фрейма

2.2.1 Формування вимог до програмного забезпечення

Вимогами називають опис функціональних можливостей і обмежень, що накладаються на створювану програмну систему. Вимоги відображають те, що система повинна робити. Тут не намагаються сформулювати, як добитися виконання цих функцій.

Наприклад, можлива така вимога до банківської системи:

Система повинна надати клієнтові можливості виконання таких операцій над його рахунком: перегляд, зняття грошей, додавання грошей.

А ось такий запис, наприклад, вимогою не є:

Інформація банківського рахунку повинна зберігатися у вигляді трьох таблиць СКБД MySQL.

Тут вказується як має бути побудована система, або що вона повинна робити. Втім, можуть бути і виключення з цього правила. Наприклад, у замовника можуть бути особливі причини для такої реалізації.

Розрізняють дві категорії представлення вимог: вимоги замовника (первинні вимоги) і вимоги розробника (детальні вимоги). Відрізняються вони одна від одної мірою опрацювання описів.

Первинні вимоги документують бажання і потреби замовника і пишуться мовою, зрозумілою замовникові. Роботу із створення первинних вимог називають *збором* або *формуванням вимог*. На стадії проектування вимоги відображаються діаграмою прецедентів (варіантів використання, Use Case).

Детальні вимоги документуються у спеціальній, структурованій формі, вони деталізовані стосовно до первинних вимог. Роботу зі створення детальних вимог називають *аналізом вимог*. На стадії проектування детальні вимоги документуються спеціальною структурованою мовою за допомогою діаграм діяльності та діаграм взаємодії.

Деякі проблеми можуть бути породжені відсутністю чіткого розуміння відмінності між цими категоріями вимог. Пояснимо відмінність між ними. Наприклад, вимога замовника має вигляд:

ВИМОГИ ЗАМОВНИКА

1. ПЗ повинно забезпечити засоби для введення і збереження різноманітних даних абонента-користувача.

А вимоги розробника, що відповідають їй, може записуватися у структурованій формі:

ВИМОГИ РОЗРОБНИКА

1.1. Користувач повинен мати можливість визначати типи даних, що вводяться.

1.2. Для кожного типу даних має бути відповідний засіб, що забезпечує введення і збереження елемента даних цього типу.

1.3. Кожен тип даних повинен представлятися відповідною піктограмою на дисплеї користувача.

1.4. Користувачеві повинна пропонуватися піктограма для кожного типу даних та можливість самостійного вибору піктограми для кожного типу даних.

1.5. При виборі користувачем піктограми типу даних до елемента даних має бути застосований засіб, що асоціюється з вказаним типом.

Розрізняють два види вимог:

1. **Функціональні вимоги** описують поведінку системи і сервіси (функції), які вона повинна виконувати. При цьому виходять з усебічного аналізу проблемної (предметної) області. Розглядаються різноманітні варіанти поведінки, що визначаються різними даними і станами зовнішнього середовища.
2. **Нефункціональні вимоги** належать до характеристик системи та її зовнішнього оточення, критеріїв ефективності й безпеки. Додатково можуть перераховуватися обмеження, що накладаються на дії і функції системи, а також на умови розробки.

Формування вимог є лише початковою фазою роботи з вимогами.

Аналіз вимог є мостом між підготовкою, плануванням і проектуванням ПЗ, який розглядає вимоги замовника як вихідні дані, на виході аналізу – вимоги розробника, які справедливо називають детальними вимогами. Тут відбувається перехід зі світу замовника у світ розробника. Змінюється мова запису вимог. Тепер це не природна мова людини, а мова формалізованих моделей. Вона незрозуміла замовнику, але зрозуміла і близька розробнику.

Реалізація процедур цього етапу проектування відбувається у процесі побудови відповідних діаграм мовою UML або іншими мовами, що використовуються системами автоматизованого проектування.

Результати процедур із формування та аналізу вимог оформляються у вигляді діаграми прецедентів, діаграми діяльності, діаграми взаємодій (діаграми послідовностей і комунікації) та їх описів.

Процес побудови кожної з цих діаграм є відображенням результатів процесу одержання певної частки інформації про вимоги до системи.

Детальніше з особливостями використання процесів побудови діаграм прецедентів, діаграм діяльності та діаграм взаємодій (у формі діаграм послідовностей і комунікації) для реалізації етапу формування та аналізу вимог об'єктно-орієнтованої технології проектування інформаційних систем можна ознайомитись у посібнику з проектування ІС [7].

2.2.2 Формування та аналіз вимог за допомогою діаграм прецедентів

Досвід показує, що моделі формування та аналізу вимог повинні доповнювати, а не замінювати специфікації вимог природною мовою (тобто вимоги в текстовому представленні).

У цьому підрозділі необхідно розглянути основне питання, яке необхідно задати на початку розробки будь-якої системи: *«Що хочуть її користувачі?»*. Тут необхідно якомога докладніше уявити дії користувачів і реакцію системи, оскільки її поведінка зумовлена їхніми вимогами. Іншими словами, робота системи залежить від того, як до неї звертаються і чого хочуть досягти.

Діаграми прецедентів (діаграми варіантів використання, Use Case) дозволяють отримати відмінне візуальне уявлення про вимоги користувачів та поведінку системи з погляду користувача. Діаграма прецедентів розглядається як головний засіб для первинного моделювання динаміки системи, використовується для з'ясування вимог замовника до розроблюваної системи, фіксації цих вимог у формі, яка дозволить проводити подальшу розробку.

Основна ідея в дослідженні ф формуванні функціональних вимог полягає в написанні історій «з життя системи». Ці історії допомагають сформулювати різні задачі та являють собою *сценарії* використання системи – це спеціальна послідовність дій або взаємодій між виконавцями і системою.

У процесі опису сценаріїв необхідно задавати питання: *«Хто є користувачем системи? Які типові сценарії використання системи? Які цілі й задачі користувачів?»*. Такий погляд на систему набагато більше орієнтований на користувача, ніж на звичайний список системних вимог, і дозволяє отримати відмінне візуальне уявлення про вимоги користувачів.

Діаграма варіантів використання є відправною точкою у процесі моделювання. Вона призначена для опису взаємодії проектованої системи з будь-якими зовнішніми або внутрішніми об'єктами – користувачами, іншими системами тощо.

До складу діаграм варіантів використання входять елементи Use Case (варіанти використання), актори, а також відносини залежності, узагальнення та асоціації. Як й інші діаграми UML, діаграми Use Case можуть включати примітки і обмеження.

Актор – це роль, яку виконує користувач або інша система, при взаємодії з проєктованою системою. Кожен актор має унікальне ім'я. Проектування діаграми варіантів використання розпочинається з визначення списку акторів.

Загальноприйнятими графічними позначеннями актора на діаграмах у нотатції мови UML являється фігурка «дротяного чоловічка», під якою записується обов'язкове ім'я актора (рис. 2.7, а). Актор також може зображуватися символом прямокутника з ключовим словом <<actor>>, записаним у формі стереотипу вище за його ім'ям (рис. 2.7, б). Додатково в мові UML 2.0 допускається зображувати актора у формі запропонованої розробником піктограми, зовнішній вигляд якої, на думку самого розробника, наочніше ілюструє характер виконуваної ним ролі (рис. 2.7, в) [5].

Наступним етапом після визначення списку акторів є визначення списку варіантів використання.

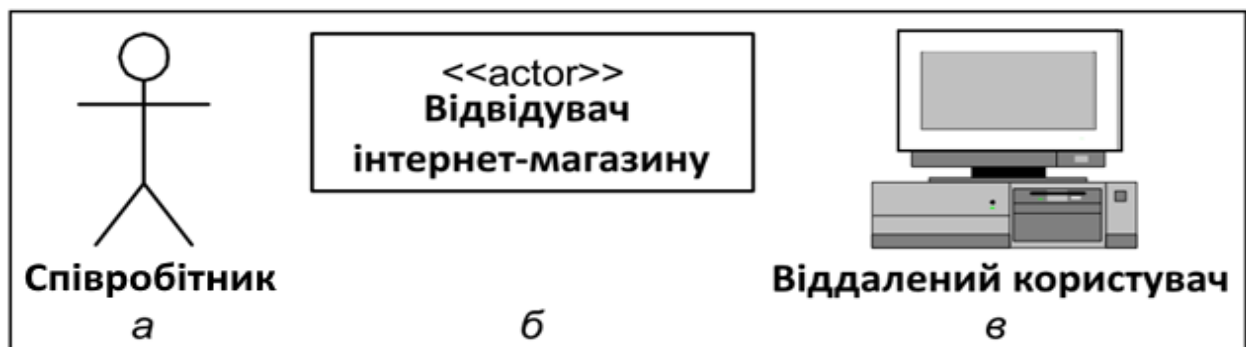


Рисунок 2.7 – Варіанти символів графічного позначення актора:
 а – у вигляді «дротяного чоловічка»; б – у вигляді прямокутника;
 в – у формі піктограми

Варіант використання – це кінцева одиниця взаємодії актора і системи. Сукупність усіх варіантів використання повністю визначає поведінку системи. Кожен варіант використання відноситься до деякого актора. Таке відношення означає, що цей актор ініціює цей варіант використання.

Залежно від мети виконання процедури розрізняють такі варіанти використання [7, розділ 10.3]:

- основні (базові) – забезпечують необхідну функціональність ПЗ, що розробляється;
- допоміжні – забезпечують виконання необхідних налаштувань системи і її обслуговування (наприклад, архівація інформації);
- додаткові – забезпечують додаткові зручності для користувача (як правило, реалізуються, якщо не вимагають серйозних витрат яких-небудь ресурсів ні при розробці, ні при експлуатації).

Загальноприйнятим графічним зображенням варіанту використання в нотації мови UML є еліпс, всередині якого або нижче нього записується ім'я у формі рядка тексту. Ім'я використовується для ідентифікації окремого варіанту використання, яке рекомендується формулювати у формі закінченої пропозиції, що починається, як правило, з іменника або дієслова.

Коментар. Коментар у мові UML призначений для включення в модель довільної текстової інформації у формі примітки, яка може бути приєднана до одного або декількох елементів моделі. Графічно коментар на усіх типах діаграм зображується у формі прямокутника з «загнутим» правим верхнім кутом (рис. 2.8).

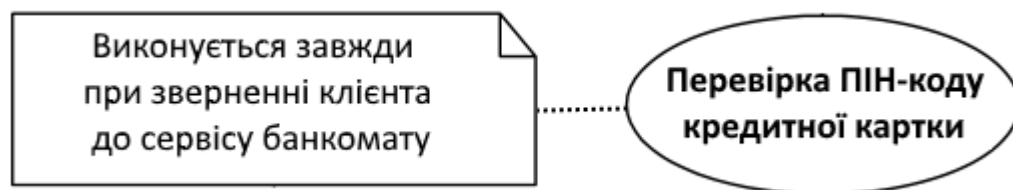


Рисунок 2.8 – Приклад коментарів у мові UML

Відношення асоціації. Асоціація є одним із фундаментальних понять у мові UML і може використовуватися на різних канонічних діаграмах при побудові візуальних моделей. Стосовно діаграм варіантів використання відношення асоціації може служити тільки для позначення взаємодії актора з варіантом використання у вигляді суцільної лінії. Розрізняють асоціації

спрямовані (вказується простою стрілкою у формі букви «V») та не спрямовані (напрям взаємодії актора з варіантом використання для розробника не має принципового значення). Приклад графічного представлення відношення спрямованої і неспрямованої асоціації між актором і варіантом використання наведено на рис. 2.9.



Рисунок 2.9 – Приклад відношення спрямованої і неспрямованої асоціації між актором і варіантом використання

Відношення узагальнення. Відношення узагальнення призначене для специфікації того факту, що один елемент моделі є спеціальним або окремим випадком іншого елемента моделі. Головною особливістю відношення узагальнення є те, що воно може зв'язувати між собою тільки елементи одного типу. Графічно відношення узагальнення позначається суцільною лінією із стрілкою у формі не зафарбованого трикутника, яка вказує на загальний елемент моделі (рис. 2.10).



Рисунок 2.10 – Приклад графічного зображення відношення узагальнення

Відношення включення. Відношення включення між двома варіантами використання в мові UML є окремим випадком загального відношення залежності (dependency), яке визначається як форма взаємозв'язку між двома елементами моделі. Зміна одного елемента моделі у відношенні залежності призводить до зміни деякого іншого елемента.

У загальному випадку залежність є спрямованим бінарним відношенням, яке зв'язує між собою тільки два елементи моделі – незалежний і залежний.

Відношення включення (include) специфікує той факт, що деякий варіант використання містить поведінку, визначену в іншому варіанті використання.

Графічно це відношення позначається як відношення залежності у формі пунктирної лінії з «V»-образною стрілкою, спрямованою від залежного до незалежного варіанту використання. Залежний варіант використання часто називають також *базовим* або *включаючим*, а незалежний – *включаємим* варіантом використання. При цьому лінія зі стрілкою обов'язково позначається ключовим словом (стереотипом) <<include>> (рис. 2.11).

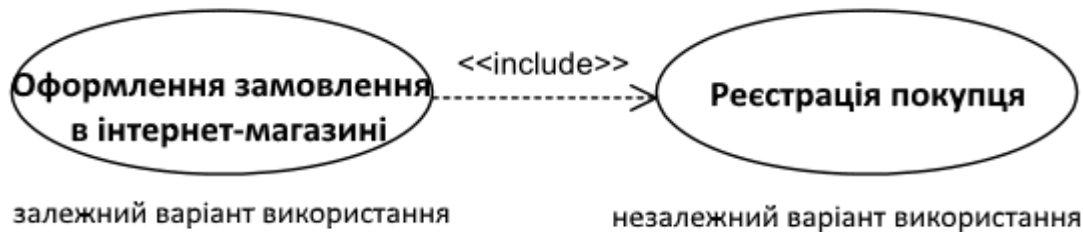


Рисунок 2.11 – Приклад графічного зображення відношення включення

Відношення розширення. Відношення розширення (extend) в мові UML також є окремим випадком загального відношення залежності між двома варіантами використання. Відношення розширення – це спрямоване бінарне відношення, що визначає взаємозв'язок одного варіанту використання з деяким іншим варіантом використання, функціональність або поведінка якого задіюються першим не завжди, а тільки при виконанні деяких додаткових умов.

Графічно це відношення позначається як відношення залежності у формі пунктирної лінії з «V»-образною стрілкою, спрямованою від залежного варіанту використання до незалежного варіанту використання. При цьому залежний варіант використання відносно розширення називається таким, що *розширює*, а незалежний – *базовим* або *розширюваним* варіантом використання. Лінія зі стрілкою обов'язково позначається ключовим словом <<extend>>, яке записується у формі стереотипу (рис. 2.12) [5].

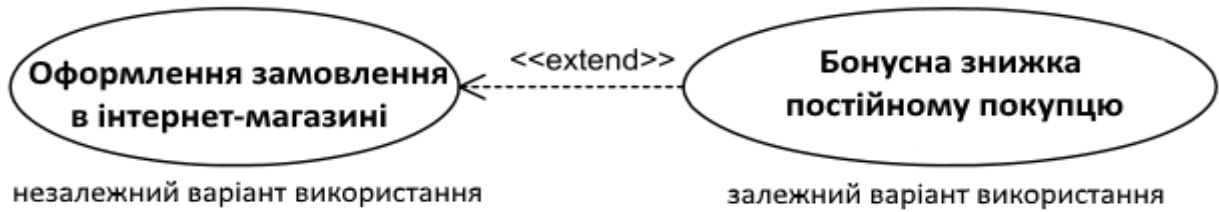


Рисунок 2.12 – Відношення розширення між варіантами використання

Для ілюстрації застосування розглянутих елементів графічної нотації мови UML побудуємо діаграму варіантів використання на прикладі системи продажу товарів в інтернет-магазині (рис. 2.13) [5].

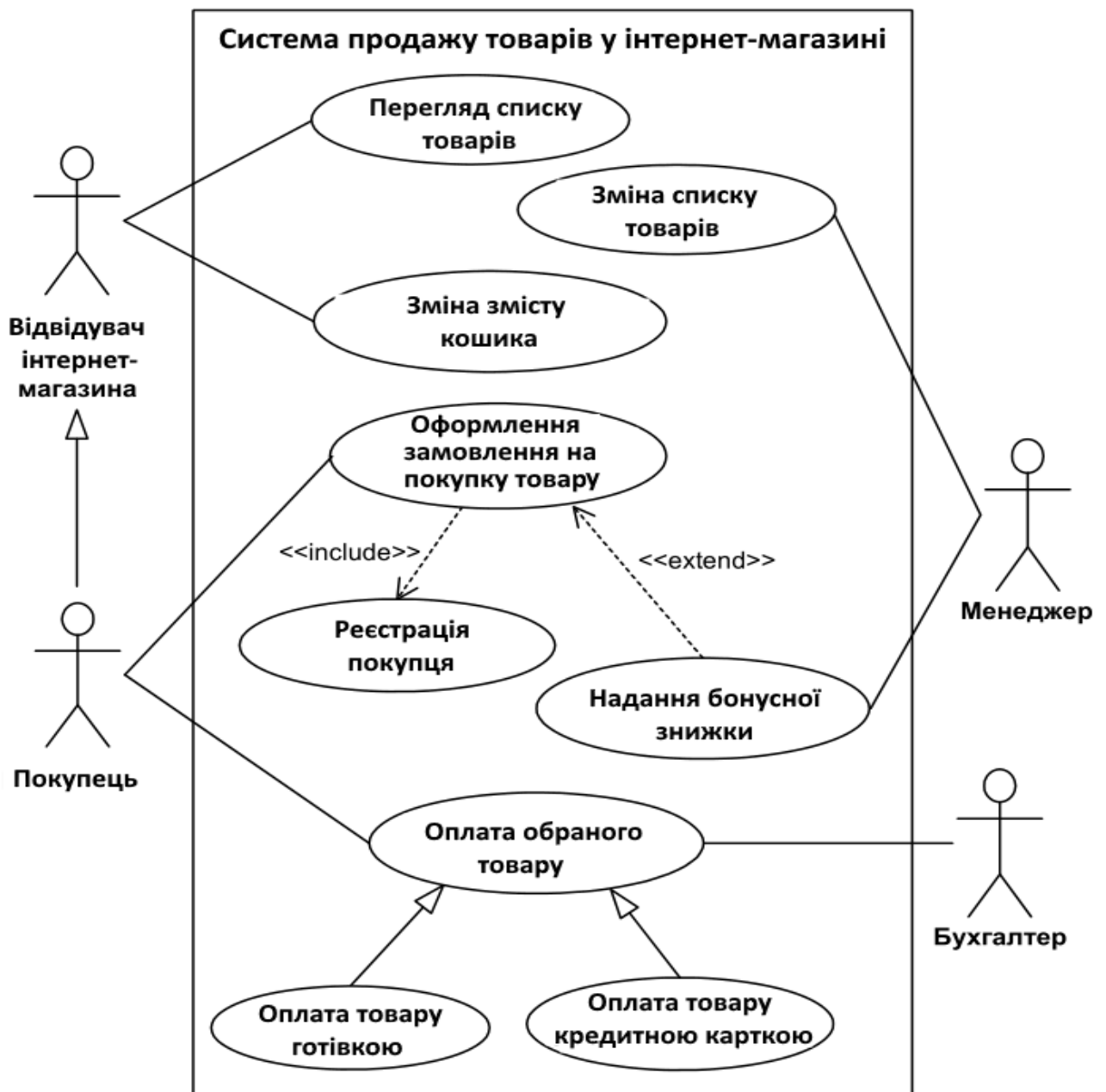


Рисунок 2.13 – Нотація діаграми використання (прецедентів)

В якості основного актора описуваної системи можна розглядати відвідувача інтернет-магазину, який може переглядати список товарів, розміщувати товар у віртуальний кошик і змінювати його вміст. Відвідувач може стати покупцем, якщо він приймає рішення про оформлення замовлення на купівлю обраних ним товарів.

Іншими акторами даної системи можуть виступати менеджер інтернет-магазину і бухгалтер. При цьому менеджер може змінювати список товарів і специфікувати умови для надання бонусної знижки, а бухгалтер – приймати плату за вибраний покупцем товар.

Найбільш значущим прецедентом для цієї системи і її акторів являється прецедент **Замовлення товарів**. Для нього побудуємо додаткову діаграму прецедентів, що пояснює цей варіант використання (рис. 2.14).



Рисунок 2.14 – Діаграма варіантів використання основного прецеденту
Замовлення товарів

Детальний опис усіх варіацій діаграм використання наведений в розділі 10.3 роботи [7].

2.3 Проектування логічної структури системи

Якщо формування та аналіз вимог проектування – це розробка відповідної концептуальної моделі, яка описує усі можливі та прийнятні варіанти розв’язання задач, які вказують, що повинна робити система, то **проектування логічної структури системи** – це розроблення логічної моделі системи у вигляді моделі класів і пакетів, які вказують з чого складається система. Логічне проектування ще називають **детальним проектуванням класів**.

2.3.1 Діаграма класів

Попередні етапи проектування забезпечили **аналіз об’єкта проектування** (формування та аналіз вимог, ідентифікація об’єктів). Але вже на етапі ідентифікації об’єктів вимальовуються архітектурні риси майбутньої системи. Паралельно з аналізом починається етап структурного проектування.

Структурне та архітектурне проектування формують фазу деталізації проектування. Архітектурний скелет доповнюється деталями – класами, що здатні реалізовувати сценарії, які описані діаграмами послідовності.

У процесі реалізації етапу детального проектування орієнтуються на максимальну підготовку до кодування програмної системи. Програмісти повинні отримати детальні проектні рішення, що забезпечать їх повною інформацією для створення програмного коду.

Фундаментальними поняттями ООАП є поняття класу і об’єкту, а також його основні принципи: абстракція, інкапсуляція, поліморфізм, успадкування. Як відомо, класи можуть застосовуватись на етапі аналізу для складання переліку об’єктів (глосарію системи) або для аналізу предметної області. Але ці класи повинні бути деталізовані до одного або кількох проектних класів на етапі детального проектування. Ця потреба викликана відсутністю у класів на етапі аналізу повного набору атрибутів та операцій.

Таким чином, основними будівельними блоками детального проектування є *класи*. Результат етапу детального проектування повинен бути відображений діаграмами класів (що містять опис всіх операцій та атрибутів класів), які призначені для статичного моделювання об'єктів.

Діаграма класів описує типи об'єктів системи і різного роду статичні відношення, які існують між ними. На діаграмах класів відображаються також властивості, операції та обмеження класів, які накладаються на зв'язок між об'єктами.

Діаграма класів не містить інформації про часові аспекти функціонування системи. Вона призначена для представлення тільки статичної структури моделі системи в термінах базових будівельних блоків і відносин між ними. До решти інших структурних діаграм належать діаграми компонентів і розгортання.

Діаграма класів може містити наступні сутності: об'єкти, класи, відношення (залежності, узагальнення, асоціації, кооперації), інтерфейси, коментарі, обмеження, пакети, підсистеми.

2.3.1.1 Основні елементи діаграм класів

Об'єкт – це деяка сутність реального світу або концептуальна (абстрактна) сутність. Прикладами об'єктів можуть служити будинок №5 по вулиці Гоголя, співробітник фірми Степан Хоменко, ваш комп'ютер або щось абстрактне: математична формула, торгове замовлення номер 123654, банківський рахунок клієнта Степана Хоменка. Об'єкт має чіткі певні межі і значення для системи і характеризується станом, поведінкою і індивідуальністю.

Стан об'єкту – це одна з умов, в якій він може знаходитися. Стан зазвичай змінюється з часом і характеризується набором властивостей, які називаються атрибутами. Наприклад, покупець визначається його ім'ям, адресою, телефоном, датою народження.

Поведінка визначає, як об'єкт реагує на запити інших об'єктів і що може робити він сам. Поведінка характеризується операціями об'єкту. Наприклад, покупець може додати товар в кошик, переглядати каталог, вилучати товар з кошику.

Як правило, в системі існує безліч об'єктів, які мають однакову поведінку, приймають однакові стани. Наприклад, співробітники фірми, яких може бути декілька десятків, і дані про яких містяться у базі даних, мають однакові атрибути (з різними значеннями цих атрибутів) – прізвище, ім'я, по батькові, дату народження, посаду тощо, а також можуть мати схожу поведінку. Для угруповання об'єктів використовуються класи.

Клас – це іменованій опис сукупності (групи) об'єктів із загальними властивостями (атрибутами), однаковою поведінкою (операціями) та типами відношень, способів доступу до даних і методів, зв'язками і семантикою.

Кожен клас є шаблоном для створення об'єкту. А кожен об'єкт – це екземпляр класу. Важливо пам'ятати, що кожен об'єкт може бути екземпляром тільки одного класу. Кожен об'єкт має свою індивідуальність, яка передбачає наявність у нього унікального імені, що відрізняється від імен інших можливих об'єктів того ж класу.

Стосовно нашого Internet-магазину можна згрупувати співробітників магазину, описавши загальний для них клас **Співробітник**. Об'єкт цього класу, наприклад, Степан Хоменко, може включати таку інформацію: ім'я, адреса, посада, розмір заробітної плати, крім того цей об'єкт може вийти у відпустку.

У нотації UML графічно клас зображується у вигляді прямокутника, який додатково може бути розділений горизонтальними лініями на три секції, які містять ім'я класу, атрибути (змінні) і операції (методи). Ім'я класу унікальне в межах пакету, вказується по центру в першій верхній секції прямокутника напівжирним шрифтом із заголовної букви (рис. 2.15).



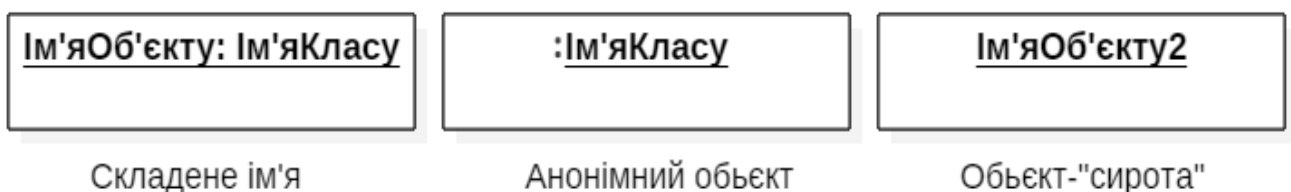
Рисунок 2.15 – Графічне зображення класів і об'єктів

Назви класів вибираються відповідно до понять предметної області. Це повинен бути іменник або словосполучення в однині, що найточніше характеризує предмет. Клас повинен описувати тільки одну сутність. Ім'я класу може бути простим, як це показано на рисунку 2.15, або складеним (рис. 2.16). Складене ім'я класу складається з самого імені класу та з імені пакету, якому належить клас, розділених двокрапкою. Ім'я класу має бути унікальним усередині пакету.



Рисунок 2.16 – Складене ім'я класу

Складене ім'я об'єкту також складається з імені об'єкту та імені класу, розділених двокрапкою. Об'єкт може бути анонімним, якщо невідоме його справжнє ім'я. Тоді на діаграмі об'єкт зображується з ім'ям, яке складається з двокрапки та імені класу, якому належить об'єкт. Якщо використовується доки невідомий клас, екземпляром якого є об'єкт, то зображується ім'я об'єкту після якого йде двокрапка та ім'я невідомого класу. Такий об'єкт називається «сиротою» (рис. 2.17).



Складене ім'я

Анонімний об'єкт

Об'єкт-"сирота"

Рисунок 2.17 – Іменування об'єктів

2.3.1.2 Виявлення класів

Виявлення класів можна розпочати з вивчення потоку подій. Іменники в описі цього потоку можуть виявитися дійовою особою, класом, атрибутом класу або виразом, що не є ні дійовою особою, ні класом, ні атрибутом класу.

Якщо в ході проектування системи вже були побудовані діаграми взаємодії, то, перед тим, як приступати до побудови діаграм класів, пошукайте на цих діаграмах схожі об'єкти. Наприклад, якщо на діаграмі послідовності, що описує оформлення замовлення об'єктами Івановим і Петровим з однаковими властивостями (ім'я, рахунок у банку тощо), то в системі повинен з'явитися клас з ім'ям **Покупець**, який буде шаблоном об'єктів Іванова і Петрова.

Деякі класи будуть виявлені при розгляді трьох стереотипів: сутність (entity), межа (boundary) та управління (control). Вже розглядалися стереотипи відношень на діаграмах прецедентів. Той же принцип створення нового типу на основі вже існуючого застосовується і для класів.

Стереотип – це механізм, що дозволяє розділяти класи за категоріями. Він використовується для створення нового типу елемента, в даному випадку нового типу класу. Наприклад, якщо необхідно виділити усі екранні форми в моделі, для цього треба створити стереотип Form (Форма). Стереотипи допомагають краще зрозуміти відповідальність кожного класу в моделі. В UML для цього застосовують три основні стандартні види стереотипів класів: класи-сутності, граничні класи та управляючі класи.

Клас-сутність містить інформацію, яка зберігається постійно. Використовуються такі класи для моделювання даних і поведінки з довгим життєвим циклом. Вони можуть представляти інформацію про предметну область, а можуть представляти елементи самої системи. Часто будучи абстракціями предметної області, вони мають найбільше значення для користувача, тому в їх назвах застосовуються терміни предметної області. Позначаються класи-сутності стереотипом <<entity>> або спеціальною піктограмою (рис. 2.18, а).

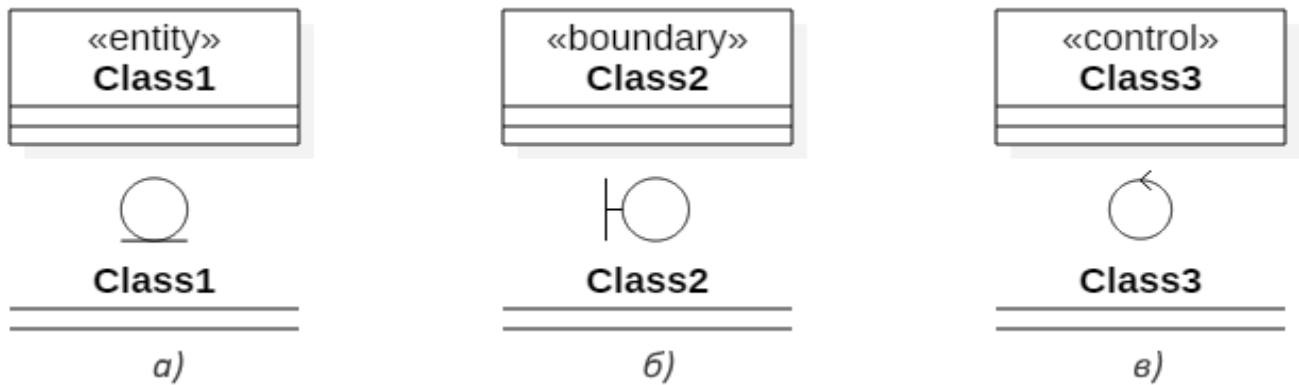


Рисунок 2.18 – Позначення класів-стереотипів

Граничними класами називаються класи, розташовані на межі системи зі всім іншим світом, вони забезпечують взаємодію між довкіллям і внутрішніми елементами системи.

Для виявлення граничних класів необхідно досліджувати діаграми варіантів використання. Для кожної взаємодії між актором і прецедентом треба створити хоча б один граничний клас. Якщо дві дійові особи ініціюють один прецедент, то вони можуть застосовувати один загальний граничний клас для взаємодії з системою. Позначаються граничні класи ім'ям стереотипу <<boundary>> або спеціальною піктограмою (див. рис. 2.18, б).

Управляючий клас відповідає за координацію дій інших класів. Вони служать для моделювання послідовної поведінки одного або декількох прецедентів і координації подій, що реалізують закладену в них поведінку. Позначаються управляючі класи ім'ям стереотипу <<control>> або спеціальною піктограмою (див. рис. 2.18, в).

Управляючий клас делегує відповідальності іншим класам. Сам він може отримувати мало повідомлень, але посилати їх множину. Його називають класом-менеджером. Він запускає альтернативні потоки і знає, як поступити в разі помилки. На початковому етапі проектування управляючі класи створюються для кожної пари актор-прецедент, надалі вони можуть об'єднуватися, розділятися або виключатися.

Розглянемо сценарій Оформлення замовлення, яке є внутрішнім потоком для прецеденту Замовлення товару. Опишемо його класи.

Цей сценарій дозволяє покупцеві оформити замовлення з кошику і провести оплату з використанням банківської карти. Покупець, знаходячись у своєму купівельному кошику, прийнявши рішення про те, що він готовий зробити замовлення в Internet- магазині, вибирає опцію «Оформити замовлення». Запускається сценарій **Оформлення замовлення**. Користувач повинен на спеціальній формі внести свої особисті дані, підтвердити замовлення або ні, і зробити оплату, потім отримати підтвердження замовлення. У системі з'являється новий об'єкт – замовлення покупця.

Необхідно вибрати хоча б один граничний клас, який здійснює зв'язок між дійовою особою **Покупець** та додатковим прецедентом **Оформити замовлення**. Назвемо його **ОформленняЗамовлення(PlaceOrder)**. Цей клас знає, які товари і в якій кількості були в кошику покупця, їх треба перенести в замовлення. Також цей клас може знати, порожній кошик покупця чи ні, і, якщо кошик порожній, то вивести про це відповідне повідомлення.

Для того щоб зробити замовлення, покупець повинен ввести свої особисті дані, електронну адресу, телефон і дані кредитної картки. Для цих цілей введемо ще один клас **ВведенняОсобистихДаних (EnterPersonalInformation)**. Після того як вибрані товари і введена особиста інформація покупця, залишається тільки перевірити деталі замовлення і погодитися з ними чи ні. Для цієї дії введемо клас **ПеревіркаДеталейЗамовлення (ConfirmOrder)**. Нарешті, коли покупець завершить оформлення замовлення, то на екрані він побачить номер замовлення і підтвердження, що замовлення послане покупцеві на електронну адресу. Для виконання цих прецедентів створимо ще один граничний клас **ПідтвердженняЗамовлення (OrderConfirmation)**.

Створимо один управляючий клас, який розподілятиме обов'язки інших класів і викликатиме їх операції при виконанні цього сценарію. Назвемо цей клас **МенеджерОформленняЗамовлення (PlaceOrderManager)**.

Вибір класів-сутностей. У сценарії **Оформлення замовлення** йдеться про покупця, замовлення і товари. Створимо класи-сутності **Покупець (Customer)**, **Замовлення (Order)**, **Товар (Item)**. Можливо, що в ході подальшого

проектування якісь нові класи будуть додані для цього сценарію, а якісь, навпаки, з цієї діаграми будуть вилучені.

Створимо усі ці класи на діаграмі класів (рис. 2.19). Краще виконувати діаграми класів і взаємодії відразу з написами англійською мовою, оскільки ці діаграми надалі будуть брати участь в генерації програмного коду системи. Якщо їх виконувати українською мовою, то згенерований код утримуватиме імена в кирилиці, і для подальшого використання програмного коду такі імена необхідно всі замінити.

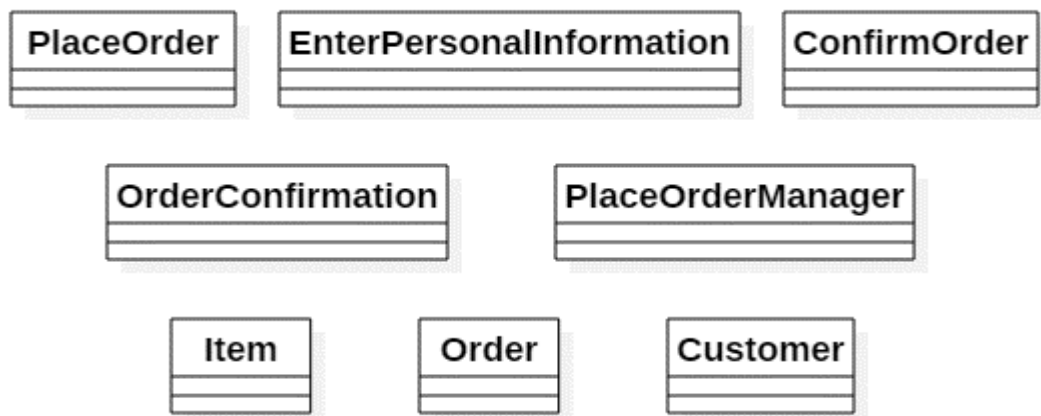


Рисунок 2.19 – Діаграма класів сценарію Оформлення замовлення

Присвоїмо класам один із стереотипів UML (текстові або графічні). Після привласнення класам стереотипів їх зовнішній вигляд зміниться. Поряд з ім'ям класу з'явиться ім'я стереотипу, розміщене в кутові дужки (рис. 2.20).

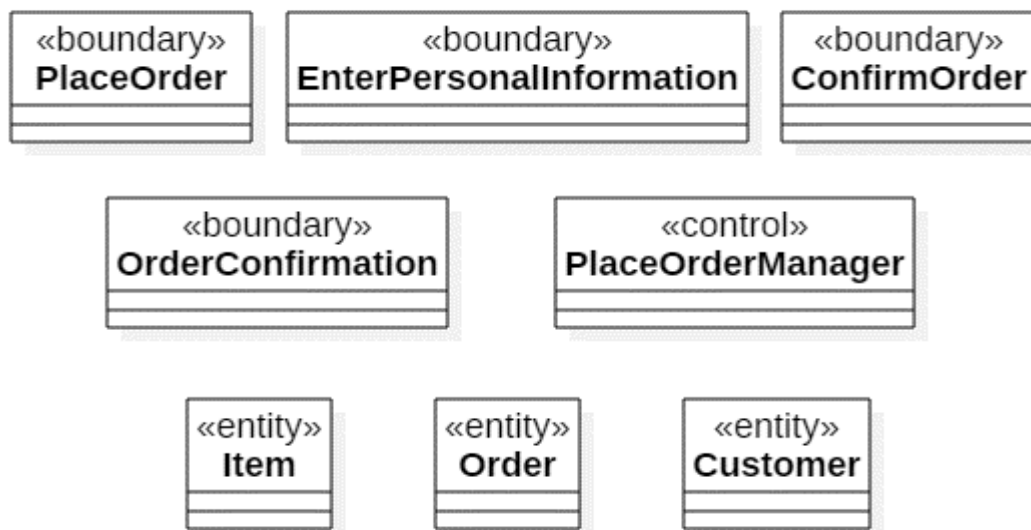


Рисунок 2.20 – Діаграма класів із стереотипами

2.3.1.3 Атрибути та операції класів

Механізм інкапсуляції в UML реалізується за рахунок об'єднання властивостей і поведінки в одному об'єкті. Властивості об'єкту описуються за допомогою атрибутів класу, до якого відноситься об'єкт, а поведінка – операціями класу.

На прямокутнику класу атрибути описуються в другій секції під ім'ям, а операції – в третій, під атрибутами.

Атрибут класу служить для представлення окремої властивості або ознаки, яка є загальною для усіх об'єктів цього класу. Атрибути, таким чином, визначають структуру класу.

Атрибут – змістовна характеристика класу, що описує безліч значень, які можуть приймати окремі об'єкти цього класу. Усім атрибутам має бути присвоєне деяке значення.

Кожному атрибуту класу відповідає окремий рядок тексту, який може складатися з квантора видимості, імені, кратності, типу значень атрибуту і його початкового значення:

**<квантор видимості> <ім'я> [кратність]: <тип>=<початкове значення>
{властивість}**

Квантор видимості може приймати одне з таких значень:

- символ «+» – загальнодоступний (Public), атрибут доступний або видимий з будь-якого іншого класу пакету, в якому визначена діаграма;
- символ «#» – захищений (Protected), атрибут недоступний або невидимий для усіх класів, за винятком підкласів цього класу;
- символ «-» – закритий (Private), атрибут недоступний або невидний для усіх класів без виключення;
- символ «~» – пакетний (Package), атрибут недоступний або невидимий для усіх класів за межами пакету, в якому визначений клас-власник цього атрибуту.

Ім'я атрибуту є текстовим рядком, який використовується як ідентифікатор відповідного атрибуту і тому має бути унікальним в межах цього класу.

Кратність атрибуту характеризує загальну кількість конкретних атрибутів цього типу, що входять до складу окремого класу. Наприклад: 0..1 – нуль або один; 2..* – два або більше; 2..5 – 2,3,4 або 5; * – будь-яке позитивне число або нуль.

Тип атрибуту є виразом, семантика якого обумовлена деяким типом даних, визначеним у пакеті «Типи даних» мови UML або самим розробником. Тип атрибуту може визначається залежно від мови програмування, яку передбачається використати для реалізації цієї моделі. Якщо атрибутом класу виступає інший клас, то типом атрибуту буде цей клас. У простому випадку тип атрибуту вказується рядком тексту, що має осмислене значення в межах пакету або моделі, до яких відноситься даний клас.

Наприклад, атрибут `item` класу `Замовлення (Order)` має бути екземпляром класу `Товар (Item)`. Для інших атрибутів задамо стандартні типи (рис. 2.21).

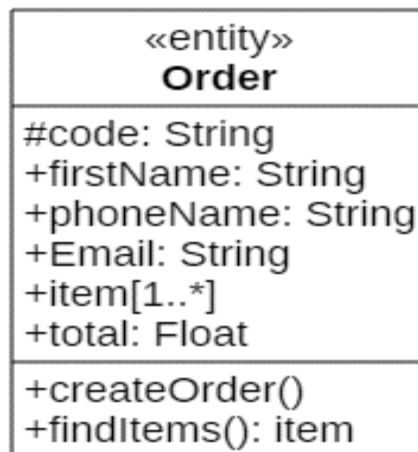


Рисунок 2.21 – Типи атрибутів класу Order

Початкове значення служить для завдання початкового значення відповідного атрибуту в момент створення окремого екземпляра класу.

Операцією або *методом класу* називається іменована послуга, яку можна запросити в будь-якого об'єкту цього класу. Сукупність операцій характеризує функціональний аспект поведінки класу.

Операції класу визначаються в розділі, який розташований нижче розділу з атрибутами. Кожній операції класу відповідає окремий рядок тексту, який може складатися з квантора видимості, імені, списку параметрів операції, типу, поверненого операцією значення:

<квантор видимості> <ім'я> (список параметрів): <тип поверненого значення> = {властивість}

Правила завдання квантора видимості і імені операції аналогічні їх визначенню для атрибутів.

Список параметрів є переліком розділених комою формальних параметрів, кожен з яких може бути представлений в такому виді:

<вид параметра> <ім'я параметра>: <вираз типу>=< значення за замовчанням>,

де *вид параметра* – одне з ключових слів: *in*, *out* або *inout*;

ім'я параметра – ідентифікатор відповідного формального параметра;

вираз типу є залежною від конкретної мови програмування специфікацією типу поверненого значення для відповідного формального параметра;

значення за замовчанням в загальному випадку є виразом для значення формального параметра, синтаксис якого залежить від конкретної мови програмування.

Вираз типу поверненого значення є залежною від мови реалізації специфікацією типів значень параметрів, які повертаються об'єктом після виконання відповідної операції.

Властивість служить для вказівки значень властивостей, які можуть бути застосовані до цього елемента.

Діаграма класів з деякими операціями атрибутами і стереотипами виглядатиме так, як показано на рис. 2.22. Наприклад, на діаграмі класу **Оформлення замовлення** перше повідомлення посилається об'єкту **PlaceOrder**. Цей об'єкт повинний уміти запускати оформлення замовлення, якщо кошик не порожній, тому для нього створили відповідну операцію **placeOrder()**.

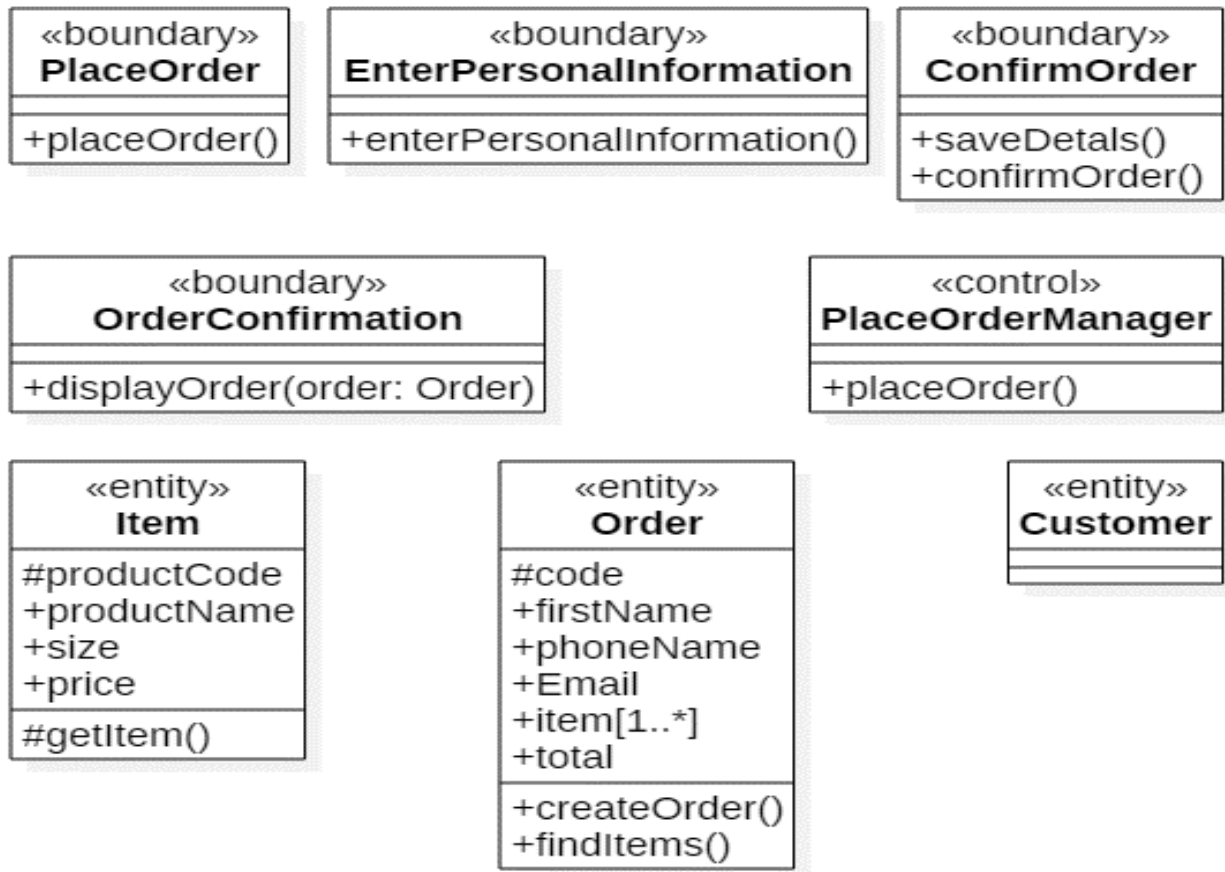


Рисунок 2.22 – Діаграма класів з операціями та атрибутами

Для класу **EnterPersonalInformation** створили відповідну операцію **enterPersonalInformation()** з повідомлення, що посилається об'єктом **PlaceOrder**, а не з такого повідомлення, що посилається **Покупцем**. Для покупця це повідомлення означає заповнення полів форми. У клас **OrderConfirmation** додали операцію **displayOrder()**, створивши її з повідомлення «display order».

2.3.1.4 Відношення між класами

Для того щоб виявити зв'язки класів, досліджуються сценарії і діаграми послідовності: якщо об'єкт посилає повідомлення іншому об'єкту, то між ними існує відношення. Відношення виникають також, якщо один клас використовує інший клас в якості параметра операції. У нотації UML визначені п'ять видів (типів) відношень між класами: *асоціації*, *агрегації*, *композиції*, *залежності* та *узагальнення* (рис. 2.23).

Перераховані типи відношень вже були визначені при побудові діаграм варіантів використання. Тому тут наведені тільки приклади їх графічного зображення на діаграмі класів.

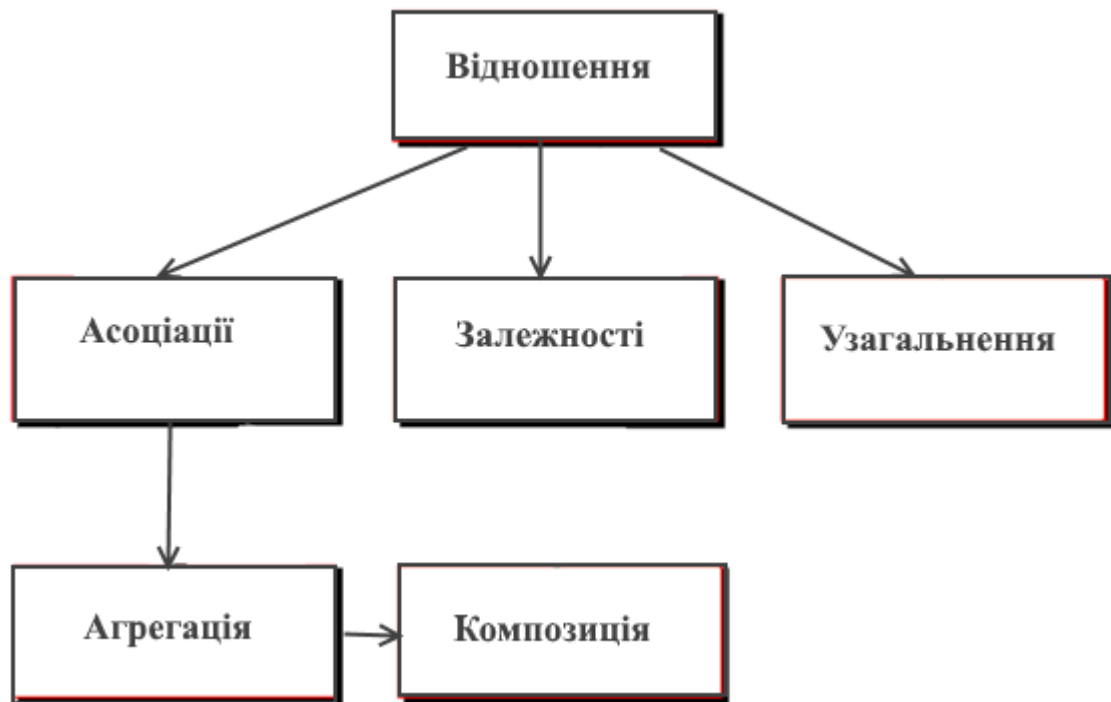


Рисунок 2.23 – Відношення між класами

Асоціація – довільне відношення або семантичний взаємозв'язок між класами, що мають загальною структуру і семантику. **Відношення асоціації** позначається суцільною лінією з додатковими спеціальними символами (ім'я асоціації, імена і кратність класів-ролей асоціації), які характеризують окремі властивості конкретної асоціації (рис. 2.24).

Асоціація може бути однонаправленою або двонаправленою. У першому випадку її зображують у вигляді стрілки, що показує напрям зв'язку (рис. 2.24, в). У другому випадку – у вигляді подвійної стрілки або просто лінія без стрілок (рис. 2.24, а).

Якщо між класами створений двонаправлений зв'язок, то кожен з них бачить відкриті атрибути і операції інших класів.

Якщо між класами встановлена однонаправлена асоціація, то в цьому випадку клас, від якого спрямована стрілка, знає відкриті атрибути і операції іншого класу, а другий клас, до якого йде стрілка, не бачить атрибути і операції

першого класу. Наприклад, клас Покупець знає назву і розмірний ряд Товару, але клас Товар не знає ім'я і адресі Покупця. При такому відношенні між класами Покупець повинен знати про клас Товар і тому не може використовуватися без нього, але клас Товар не залежить від Покупця і може бути повторно використаний самостійно, незалежно від класу Покупець.

Асоціація може бути рефлексивною, це означає, що один екземпляр класу звертається до іншого екземпляра цього класу. Відображається відношення рефлексії як на рис. 2.24, з.

Асоціації можна присвоїти ім'я. Зазвичай в якості імені використовується дієслово або речення, що пояснює причину виникнення зв'язку. За допомогою чорного трикутника, розташованого над лінією зв'язку справа або зліва від імені асоціації, уточнюється сенс імені й вказується напрям імені зв'язку (рис. 2.24, б).

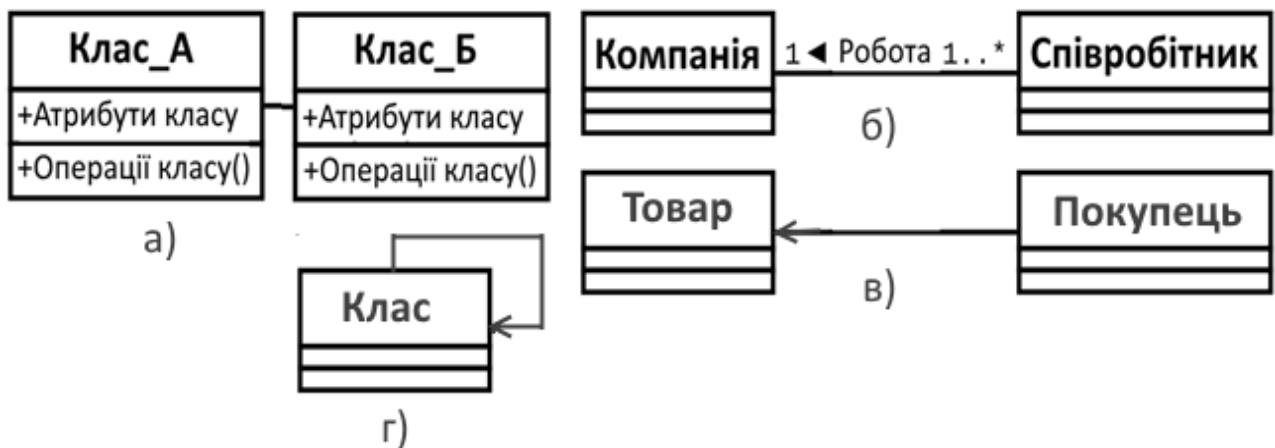


Рисунок 2.24 – Графічне зображення відношення асоціації на діаграмі класів

Закінчення лінії асоціації в місці, де вона з'єднується з класом, називається **роллю асоціації**. Назва ролі асоціації – це іменник, який уточнює, описує роль, в якій один клас бере участь у зв'язку з іншим класом. Роль може бути вказана для обох класів, що беруть участь у зв'язку, або тільки для одного (рис. 2.25).

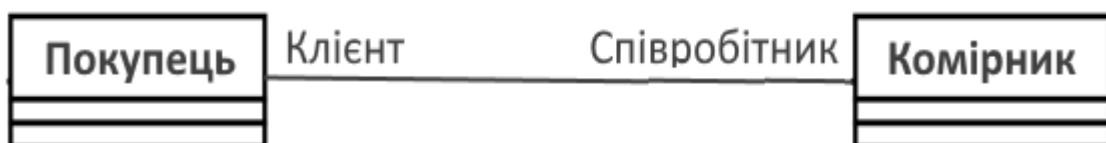


Рисунок 2.25 – Асоціація з ролями

Кратність (потужність) визначається для класів і вказує допустиму кількість об'єктів (екземплярів класу), що беруть участь у відношенні асоціації. Кратність вказує, скільки екземплярів одного класу взаємодіють за допомогою відношення з одним екземпляром цього класу в даний момент.

Приклади індикаторів потужності:

- 0..1 нуль або один;
- 1 рівно один;
- 1..* один або багато;
- 2..5 2,3,4 чи 5.

Наприклад, покупець може оформити багато замовлень або не оформити жодного. Замовлення має бути зроблене тільки 1 покупцем (рис. 2.26).



Рисунок 2.26 – Кратність класів в асоціації

Як видно з прикладу, читати кратність класу треба на протилежному кінці зв'язку. Вказувати ім'я, роль або кратність зв'язку необов'язково. Це треба робити, коли це може допомогти точнішому представленню моделі системи і кращому її розумінню.

Якщо в діаграмі класів асоціація між двома класами має спеціальний вигляд «частина-ціле», то використовується асоціація спеціального виду – **агрегатна асоціація**. У цьому випадку клас «ціле» має більш високий концептуальний рівень, ніж клас «частина». Графічно агрегатні асоціації зображуються у вигляді простої асоціації з незафарбованим ромбом на стороні класу-«цілого» (рис. 2.27).

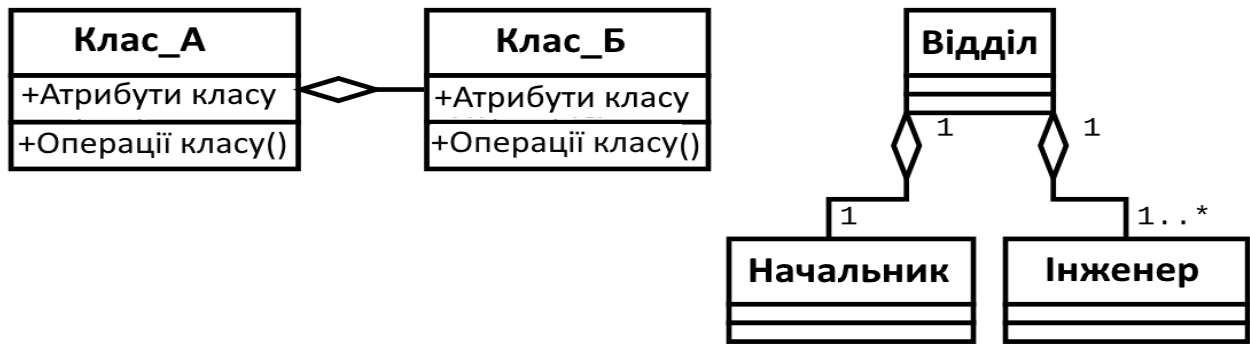


Рисунок 2.27 – Графічне зображення відношення агрегації на діаграмі класів

У нашому прикладі з Internet-магазином будь-яке замовлення складається з товарів (агрегатна асоціація, рис. 2.28).

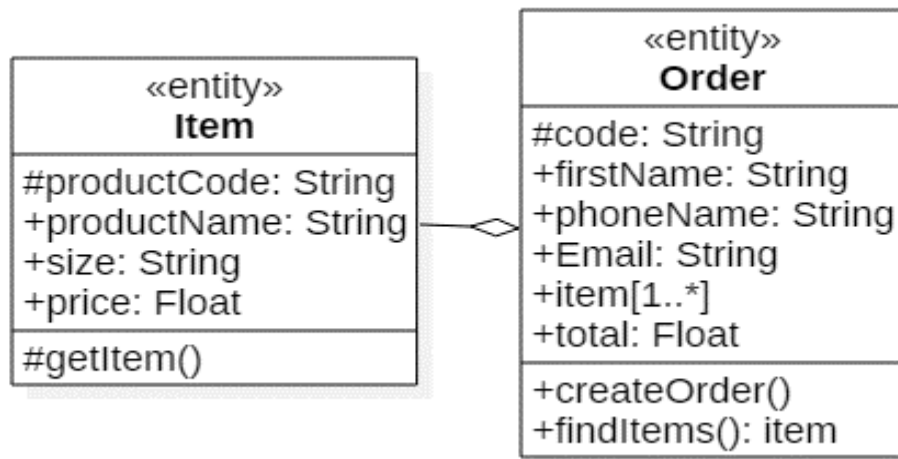


Рисунок 2.28 – Агрегація між класами Товар (**Order**) і Замовлення (**Item**)

Якщо зв'язок «частини» і «цілого» настільки сильний, що знищення «цілого» приводить до знищення усіх його «частин», то такі агрегатні асоціації називаються *композиційними* або просто *композиціями*. За наявності композиції об'єкт-частина може бути частиною тільки одного об'єкту-цілого (композиту), причому година життя частин і цілого співпадають. Як тільки буде знищений об'єкт-ціле, так разом з ним буде знищений об'єкт Частина. При звичайній агрегатній асоціації «частина» може одночасно належати декільком «цілим». Графічно композиція зображується у вигляді простої асоціації, доповненої зафарбованим ромбом з боку «цілого» (рис. 2.29).

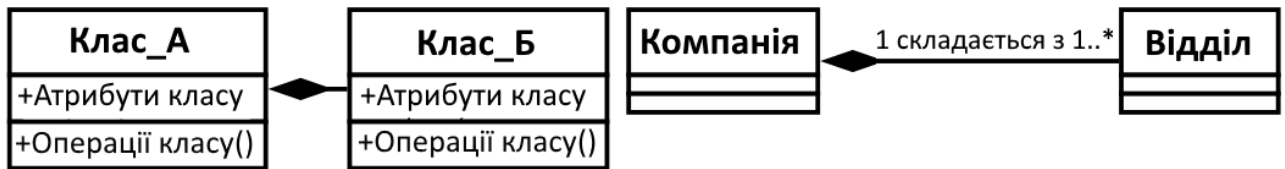


Рисунок 2.29 – Графічне зображення відношення композиції на діаграмі класів

Для відношень композиції і агрегації можуть використовуватися додаткові позначення, які застосовуються для відношення асоціації. А саме можуть вказуватися кратності окремих класів, які в загальному випадку не є обов'язковими.

Проте кратність агрегованого кінця композитної агрегації не повинна мати верхньої межі більше ніж 1. Розглянемо приклад – вікно графічного інтерфейсу програми, яку складається із заголовка, піввісі прокрутки, головного меню і робочої області (рис. 2.30). Як тільки буде закрито вікно програми, усі інші елементи вікна перестануть існувати. Подібне вікно є класом, де його елементи також є класами.

Оскільки агрегація і композиція є відношеннями асоціації, то для них допустимо, але не обов'язково, вказувати ім'я, роль і кратність.

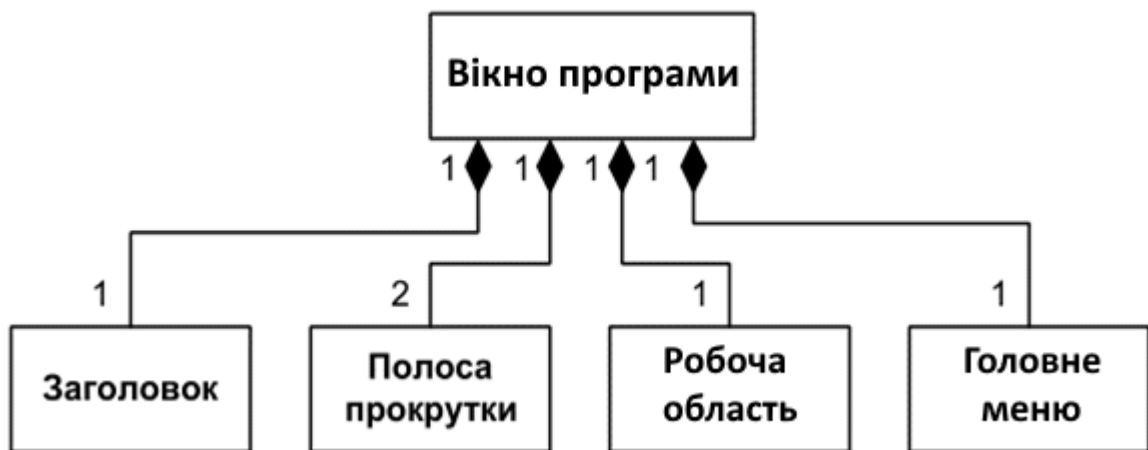


Рисунок 2.30 – Діаграма класів для ілюстрації відношення композиції

Залежністю називається відношення використання, яке визначає, що зміна в специфікації однієї сутності може вплинути на іншу сутність, яка її використовує, причому зворотне в загальному випадку не вірно.

Відношення залежності графічно зображується пунктирною лінією між відповідними елементами зі стрілкою на одному з її кінців ("->" або "<-"). На діаграмі класів дане відношення зв'язує окремі класи між собою, при цьому стрілка спрямована від класу-клієнта залежності до незалежного класу. На рис. 2.31 зображені два класи: Клас_А и Клас_Б, при цьому Клас_Б – незалежний, а Клас_А – клієнт.



Рисунок 2.31 – Графічне зображення відношення залежності

Наприклад, якщо використати клас **Order** (Замовлення) як вхідний параметр операції **displayOrder** (відобразитиЗамовлення) класу **OrderConfirmation** (ПідтвердженняЗамовлення), то для виконання цієї операції клас **OrderConfirmation** (ПідтвердженняЗамовлення) використовує клас **Order** (Замовлення), тобто вони пов'язані відношенням залежності (рис. 2.32).

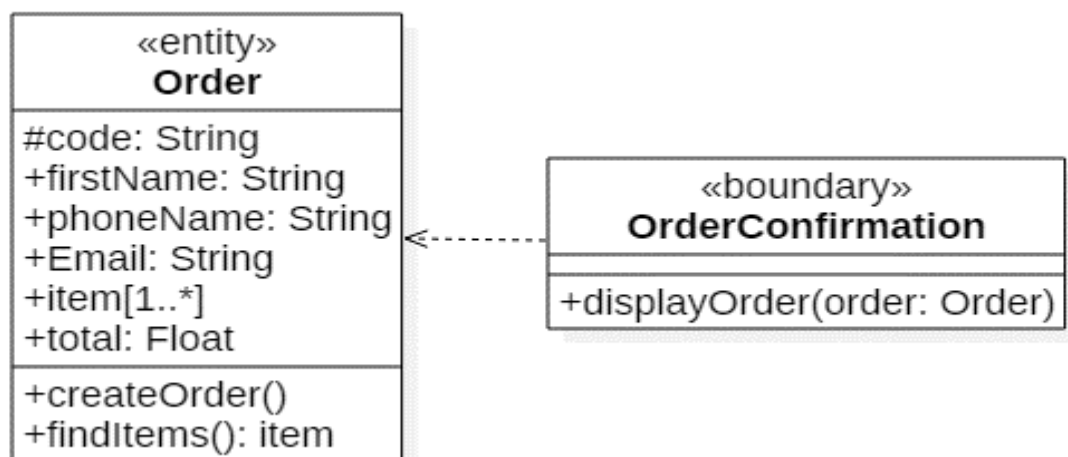


Рисунок 2.32 – Залежність між класами Order і OrderConfirmation

Створення екземпляра класу **ПідтвердженняЗамовлення** не спричинить за собою створення екземпляра класу **Замовлення**. Проте ці два класи зможуть обмінюватися повідомленнями на діаграмах взаємодії.

Узагальнення – це відношення успадкування між двома елементами моделі. Воно дає класу можливість успадковувати відкриті або захищені атрибути і операції суперкласу (класу від якого успадковуються атрибути і операції). Крім успадкованих кожен клас може мати свої атрибути і операції.

На діаграмах узагальнення зображується у вигляді стрілки з не зафарбованим трикутником у суперкласі, що йде від нащадка.

Наприклад, у магазині можуть працювати різні співробітники: співробітник відділу продажів, комірник, директор. Усі вони мають загальні властивості: ім'я, адреса, телефон, дата народження, посада, тому можна розглядати узагальнюючу сутність **Співробітник**, атрибути і операції якої сутності **Директор** і **Комірник** будуть наслідувати (рис. 2.33).



Рисунок 2.33 – Відношення узагальнення між класами

Якщо **Співробітник** має в якості атрибутів ім'я, адресу, телефон, дату народження, посаду, то сутності **Директор** і **Комірник**, звичайно, наслідують ці атрибути зі своїми значеннями. Крім того вони можуть мати і власні атрибути або операції. Наприклад, у **Директора** може бути операція **звільнити Співробітника**, якої не може бути у **Комірника**, а у останнього операція – **видати Товар**.

Закриті атрибути і операції не можуть успадковуватися нащадками.

2.3.1.5 Приклади діаграм класів

На рис. 2.34 представлена діаграма класів сценарію Оформлення замовлення в Internet-магазині. Клас **PlaceOrder** пов'язаний з **EnterPersonalInformation**, а об'єкт **ConfirmOrder** посилає повідомлення об'єкту класу **PlaceOrderManager**. **PlaceOrderManager** пов'язаний з об'єктами класів **Order** і **OrderConfirmation**. Усі перераховані зв'язки визначимо відношенням асоціації. Клас **OrderConfirmation** використовує клас **Order** як параметр своєї операції: між ними визначимо відношення залежності. Екземпляри класу **Order** складаються з екземплярів класу **Item**. Між ними створено відношення агрегації.

Для того щоб клас **ConfirmOrder** міг виконувати операцію підтвердження замовлення, він має бути пов'язаний з класом **EnterPersonalInformation**, тому між ними створено відношення асоціації.

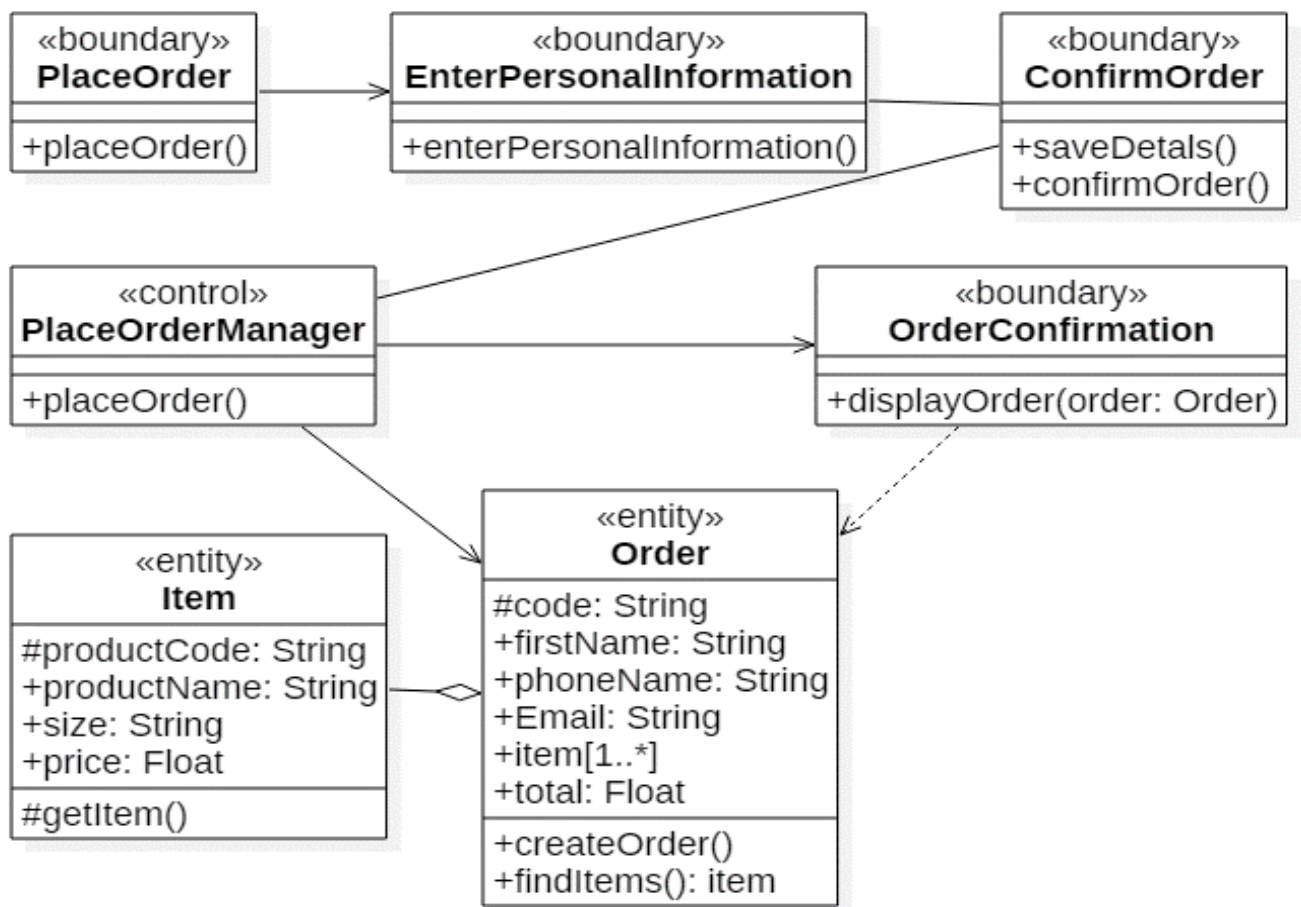


Рисунок 2.34 – Діаграма класів сценарію Оформлення замовлення

На рис. 2.35 показана сукупність класів, узятих з ІС закладу вищої освіти (ЗВО). У нижній частині діаграми розташовані класи **Студент**, **Курс** і **Викладач**. Між класами **Студент** і **Курс** є асоціація, що показує, що студенти можуть відвідувати курси. Більше того, кожен студент може відвідувати будь-яку кількість курсів і на кожен курс може записатися будь-яке число студентів.

Два класи (**ЗВО** і **Факультет**) містять операції, що дозволяють маніпулювати їх частинами. Ці операції включені із-за їх важливості для підтримки цілісності даних (наприклад, додавання або видалення класу **Факультет** зачіпає цілий ряд інших об'єктів).

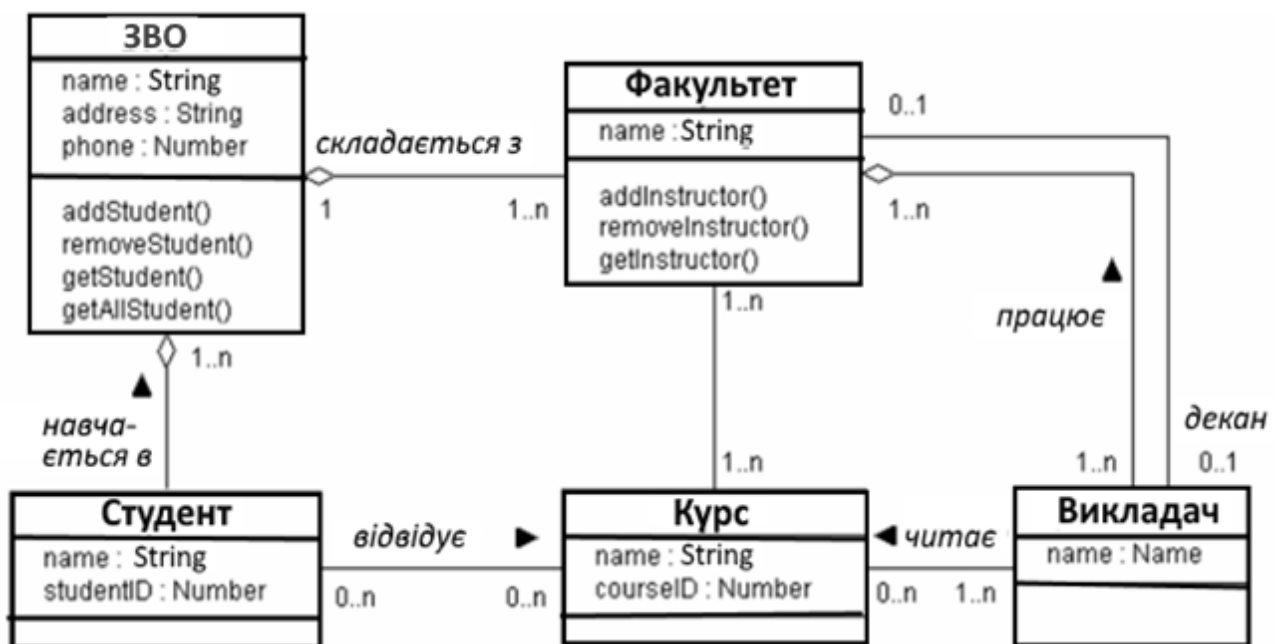


Рисунок 2.35 – Діаграма класів для інформаційної системи ЗВО

Оскільки діаграма класів є однією з головних діаграм моделі проектованої системи, то побудуємо ще одну UML-діаграму класів. В якості прикладної області візьмемо відділ кадрів деякого підприємства (наприклад, internet-магазину) і почнемо будувати його модель. Для прикладів атрибутів і методів класів використовуватимемо мову Java.

Узагальнення. Розглянемо два класи: абстрактний клас «**Man**» (людина) і більш спеціалізований клас «**Employee**» (співробітник). Оскільки клас «**Employee**» наслідує властивості та методи класу «**Man**», то між ними встановимо зв'язок у вигляді відношення узагальнення (рис. 2.36). Відношення

узагальнення – це спадкоємність. Це відношення у будь-якій ООП мові програмування має явну реалізацію через розширення (extends) одного класу іншим.

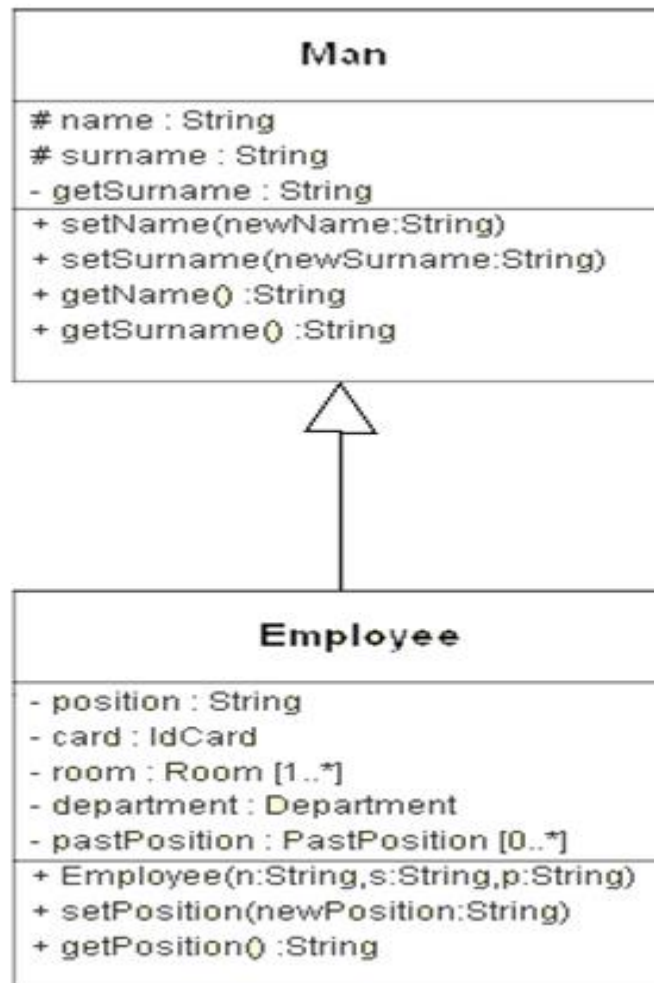


Рисунок 2.36 – Відношення узагальнення

Асоціація. Асоціація показує відношення між об'єктами-екземплярами класу. У модель додаємо клас «**IdCard**», що представляє ідентифікаційну картку (пропуск) співробітника. Кожному співробітнику може відповідати тільки одна ідентифікаційна картка, потужність зв'язку 1 до 1 (бінарна асоціація).

Клас **Employee** має поле `card`, в якого тип `IdCard`, клас також має методи для надання значення (`setIdCard`) цьому полю і для отримання значення (`getIdCard`). З екземпляра об'єкту **Employee** можна дізнатися про пов'язаний з ним об'єкт типу **IdCard**, а це означає, що навігація (стрілка на лінії) спрямована від класу **Employee** до класу **IdCard** (рис. 2.37).

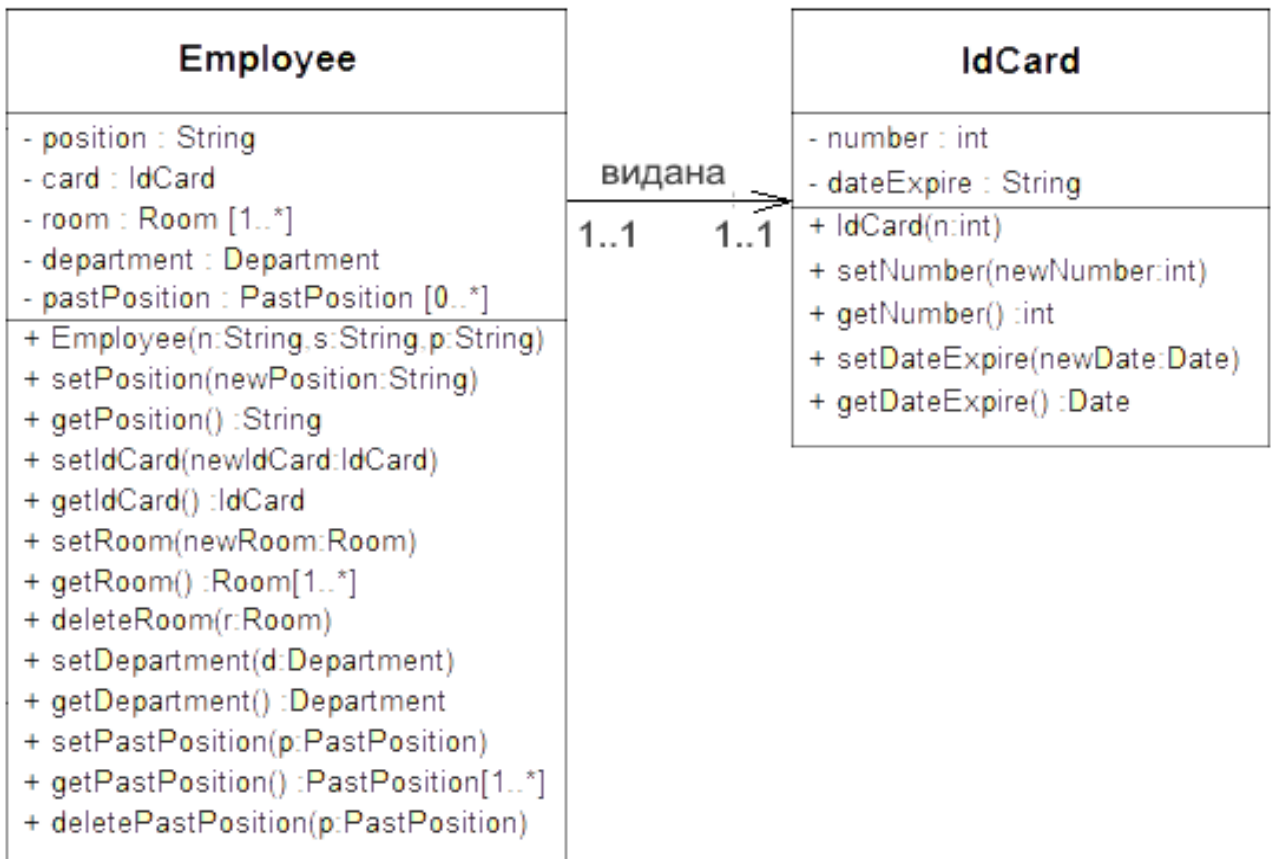


Рисунок 2.37 – Бінарна асоціація

Якщо в організації закріплюють за працівниками приміщення, то додаємо новий клас **Room**. Кожному об'єкту працівник (**Employee**) може відповідати декілька робочих приміщень. Потужність зв'язку один-до-багатьох. Додаємо в клас **Employee** поле і методи для роботи з класом **Room**. Навігація від **Employee** до **Room** (рис. 2.38).

Агрегація. Якщо підприємство структуроване по відділах, то введемо в модель клас **Department** (відділ). У кожному відділі може працювати один або більше людей. Відділ включає одного або більше співробітників і, таким чином, їх агрегує. На підприємстві можуть бути співробітники, які не належать жодному відділу, наприклад, директор підприємства (рис. 2.39).

Клас **Department**, окрім конструктора і методу зміни імені відділу, має методи для занесення у відділ нового співробітника, для видалення співробітника і для отримання всіх співробітників, що входять у цей відділ.



Рисунок 2.38 – N-арна асоціація

Навігація на діаграмі не показана, це означає, що вона є двонаправленою: від об'єкту типу «**Department**» можна дізнатися про співробітника і від об'єкту типу «**Employee**» можна дізнатися до якого відділу він відноситься.

Для того щоб дізнатися до якого відділу відноситься який-небудь співробітник, то додамо у клас **Employee** поле і методи для призначення і отримання відділу.

Композиція. Припустимо, що однією з вимог до розробленої системи є вимога про те, щоб зберігати дані про колишню займану посаду на підприємстві. Введемо новий клас «**pastPosition**». У нього, окрім властивості «ім'я» (name), введемо і властивість «department», яка зв'яже його з класом «**Department**».

Дані про минулі займані посади є частиною даних про співробітника, таким чином, між ними зв'язок ціле-частина. У той же час, дані про минулі посади не можуть існувати без об'єкту типу «**Employee**».

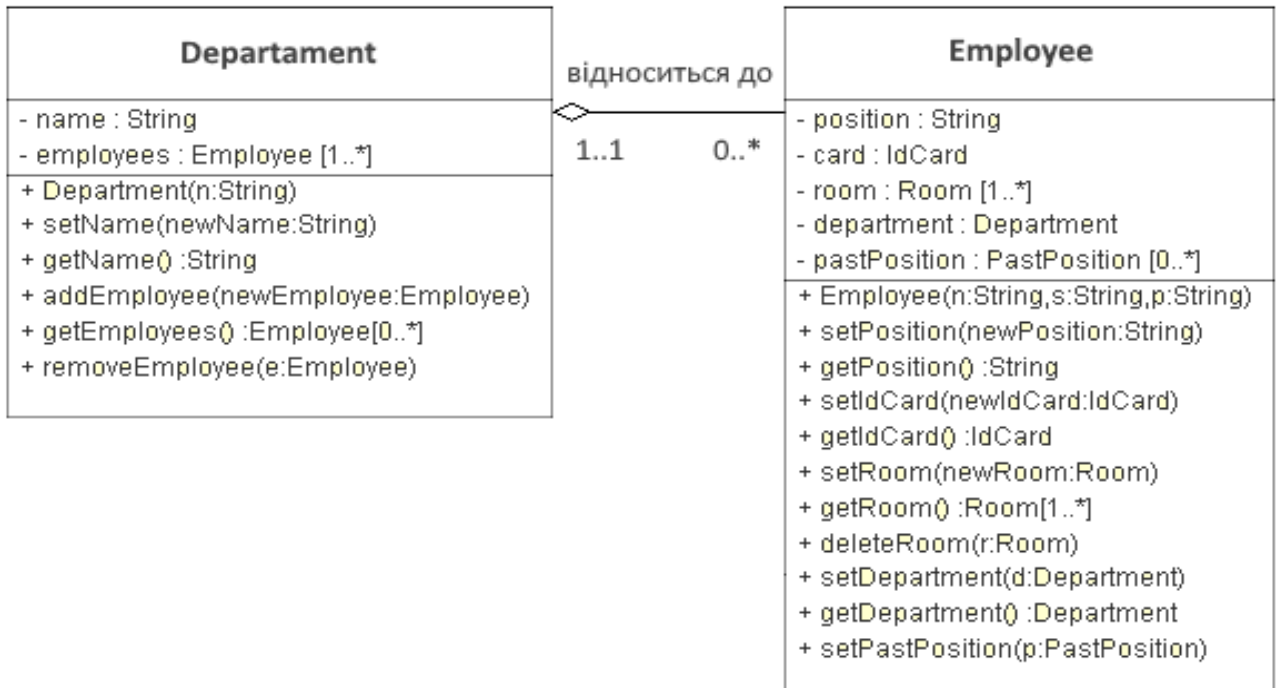


Рисунок 2.39 – Відношення агрегації

Знищення об'єкту «Employee» повинне привести до знищення об'єктів «pastPosition» (відношення композиції, рис. 2.40).

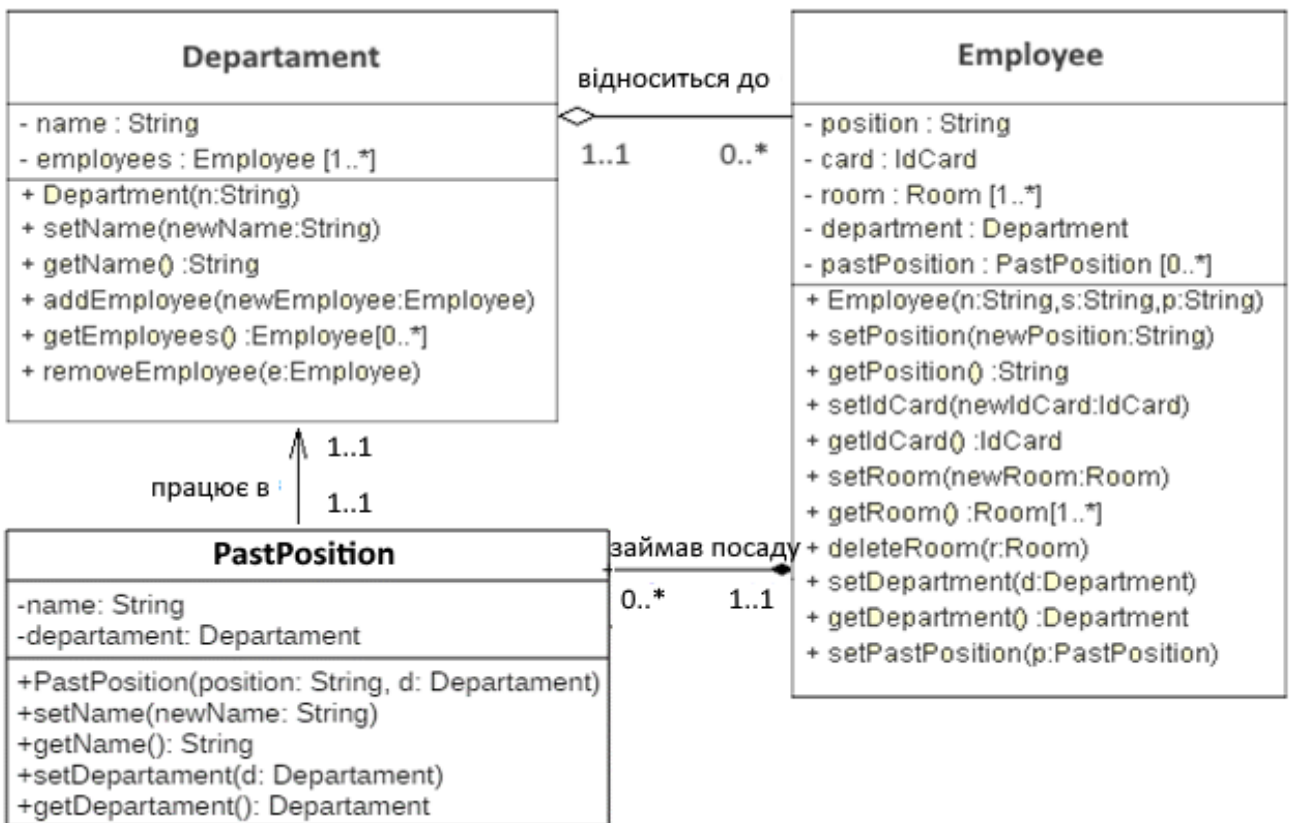


Рисунок 2.40 – Відношення композиції

У клас **Employee** додамо властивості й методи для роботи з даними про минулу посаду (`setPastPosition()`, `getPastPosition()`, `deletePastPosition()`).

Залежність. Для організації діалогу з користувачем введемо в систему клас «**Menu**» з методом «`showEmployees`», який показує список співробітників та їх посади. Параметром для методу є масив об'єктів «**Employee**». Таким чином, зміни внесені у клас «**Employee**» можуть спричинити і зміни класу «**Menu**» (відношення залежності, рис. 2.41).

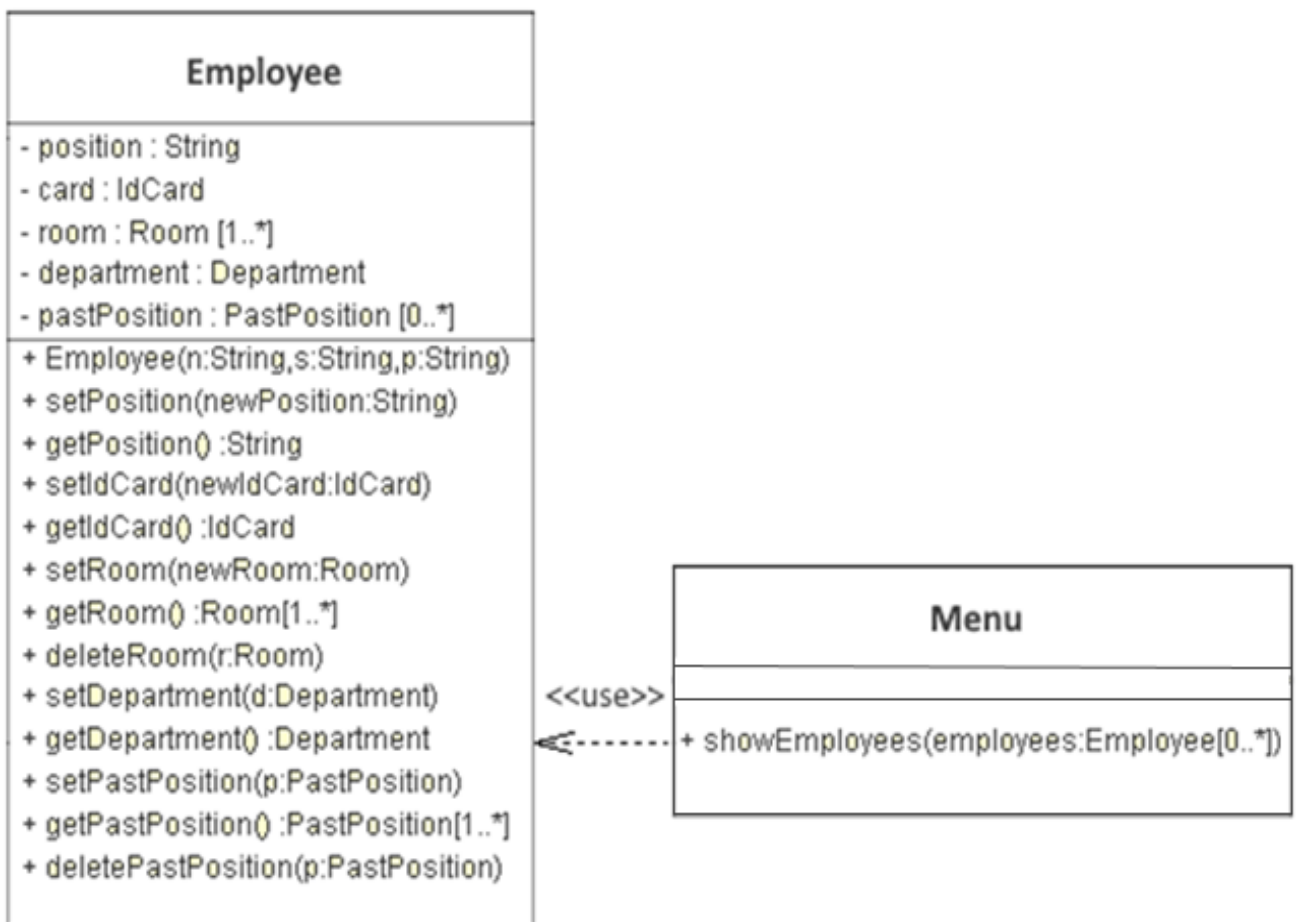


Рисунок 2.41 – Відношення залежності

На даній діаграмі клас «**Menu**» не відноситься до прикладної області, а є «системним» класом уявного застосування.

Таким чином, у результаті моделювання отримаємо таку загальну діаграму класів відділу кадрів (рис. 2.42).

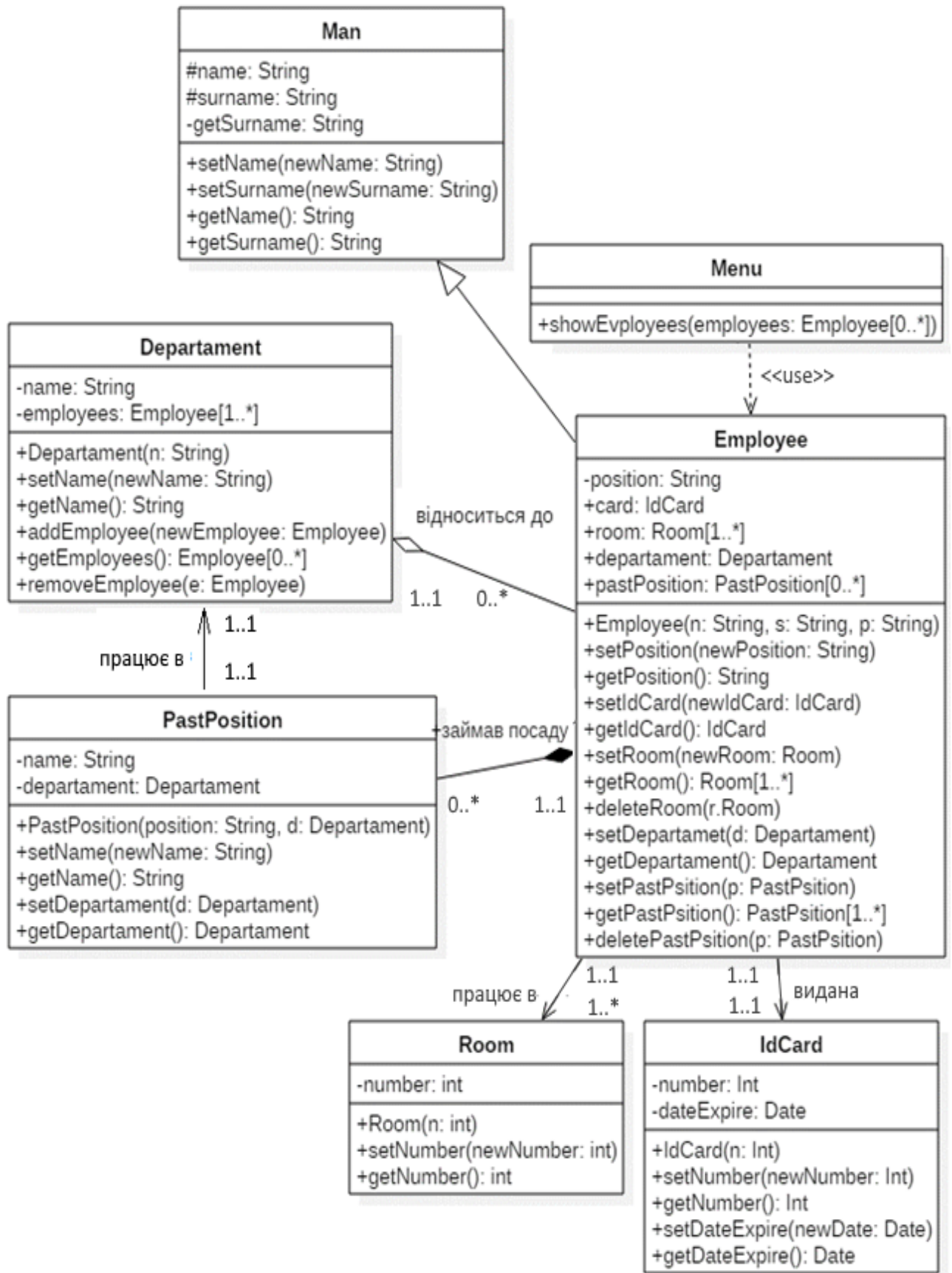


Рисунок 2.42 – Діаграма класів відділу кадрів

2.3.2 Діаграма пакетів

Якщо в моделі небагато класів, то ними легко управляти, проте більшість систем містять велику кількість класів, тому потрібний механізм, що дозволяє групувати класи і полегшувати їх повторне використання. Таким механізмом в UML є пакети.

Пакети – це спосіб угруповання будь-яких елементів мови UML: діаграм, варіантів використання, класів тощо. Пакети застосовують, щоб розбивати великі моделі на частини, групуючи класи, що мають деяку спільність. Це значною мірою спрощує розуміння і підтримку проекрованої моделі.

Пакети є життєво необхідним засобом для великих проектів. Їх слід використати в тих випадках, коли діаграма класів, що охоплює усю систему в цілому і розміщена на єдиному аркуші паперу формату А4, стає нечитабельною.

Існує декілька найпоширеніших підходів до угруповання.

Угруповання по стереотипу. Стереотип – це механізм, що дозволяє розділяти класи на категорії. В UML визначені три основні стереотипи: межа, об'єкт та управління.

Пограничні класи розташовані на межі проекрованої системи. Вони включають усі форми взаємодії з довкіллям: звіти, форми документів, інтерфейси з апаратурою, користувачами й іншими системами.

Класи-сутності містять інформацію про класи внутрішнього устрою системи, тобто є сутністю цієї системи.

Управляючі класи відповідають за координацію дій інших класів, їх взаємодію між собою.

Угруповання по функціональності. Окрім згаданих вище стереотипів можна створювати і свої власні (теми пакетів), тобто тема пакету може бути досить довільною. Нею може бути основний клас, основні відношення, найважливіші аспекти функціональності. Наприклад, модель класів компілятора можна було б розділити на пакети лексичного аналізу, розбору, семантичного аналізу, генерації коду та оптимізації.

Для графічного зображення пакетів на діаграмах застосовується спеціальний графічний символ – великий прямокутник з невеликим прямокутником, приєднаним до лівої верхньої частини першого. Візуально символ пакету нагадує піктограму теки на екрані монітора (рис. 2.43).



Рисунок 2.43 – Графічне зображення пакетів

Пакет може бути зображений без своїх членів. У цьому випадку ім'я пакету можна вказати у великому прямокутнику, воно має бути унікальним у межах даної моделі. Якщо зображуються класи-члени пакету, то тип пакету пишеться на закладці, а всередині великого прямокутника перераховуються класи-члени пакету за допомогою символу класу.

У UML для встановлення взаємодій між пакетами в основному використовується *відношення залежності*.

Залежність – це однонаправлене відношення, що показує, як клас залежить від визначень, зроблених в іншому класі. Залежність означає відношення типу «постачальник-клієнт», коли модифікація постачальника може вплинути на одного або декілька клієнтів. Клієнт – елемент моделі, залежний в деякому контексті від елемента або елементів постачальника.

Залежність між двома пакетами існує в тому випадку, якщо між двома будь-якими класами з цих пакетів існує будь-який взаємозв'язок.

Відношення залежності зображують у вигляді пунктирної лінії зі стрілкою, спрямованою від залежного класу до незалежного класу (рис. 2.44).

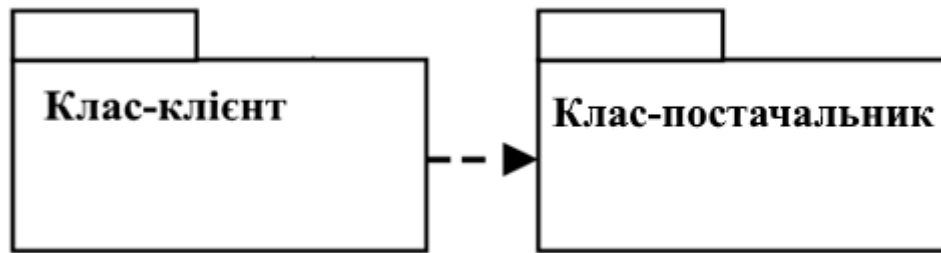


Рисунок 2.44 – Графічне зображення відношення залежності

Іншим типом відношень між пакетами (без використання ліній залежності) є відношення вкладеності або включення пакетів один в одного. Це відношення може бути зображене простим розміщенням одного пакету усередині іншого пакету (рис. 2.45, а). Так, у даному випадку **Пакет А** містить у собі два підпакети: **Пакет В** і **Пакет С**. З іншого боку, це ж відношення може бути зображене за допомогою відрізків ліній аналогічно графічному представленню дерева (рис. 2.45, б). Пакет-контейнер з'єднується з підпакетами суцільною лінією, на кінці якої зображується спеціальний символ – \oplus .

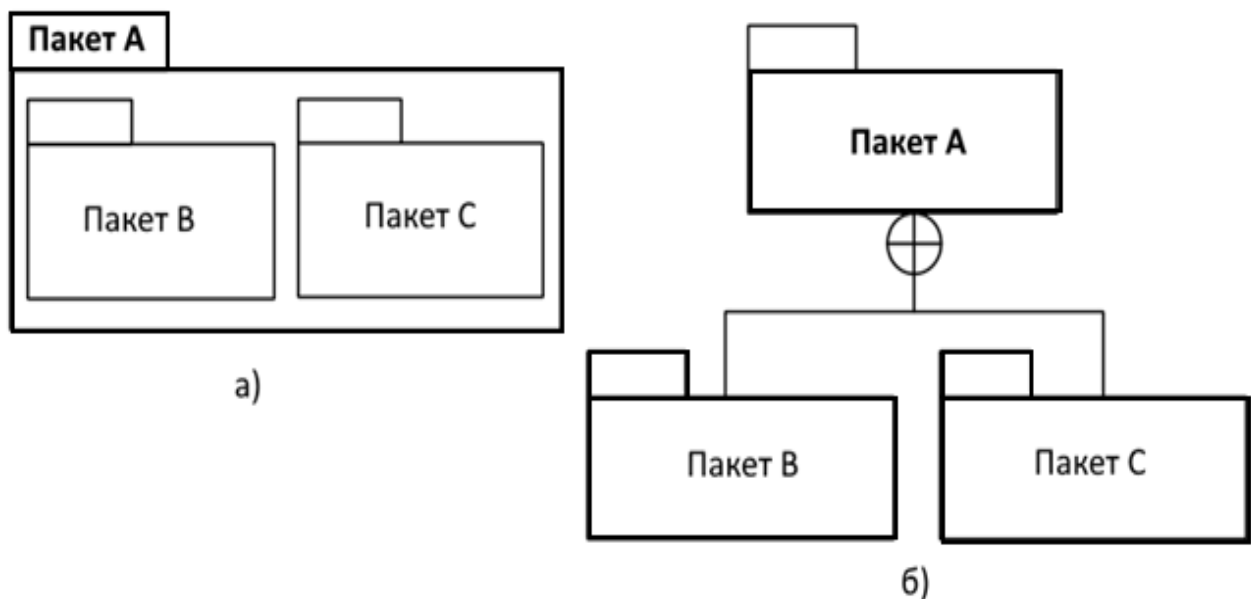


Рисунок 2.45 – Приклади графічного зображення пакетів

Таким чином, *діаграма пакетів* – це структурна діаграма, призначена для представлення розміщення елементів моделі в пакетах і специфікації залежностей між пакетами та їх елементами (рис. 2.46).

У контексті архітектурного проектування пакет є зручним засобом упаковки підсистеми, яка є частиною архітектурної організації, а діаграма пакетів – моделлю структури архітектурного рівня.

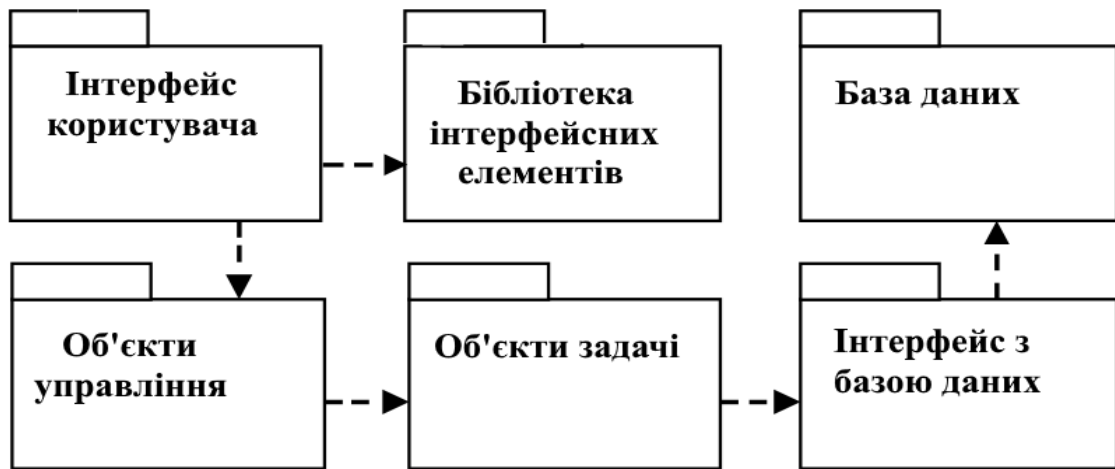


Рисунок 2.46 – Проста діаграма пакетів

2.4 Архітектурне проектування

Основне призначення логічного представлення полягає в аналізі структурних і функціональних відношень між елементами моделі системи. Проте для створення конкретної фізичної системи необхідно деяким чином реалізувати усі елементи логічного представлення в конкретні матеріальні сутності. Для опису таких реальних сутностей призначений інший аспект модельного представлення, а саме – фізичне представлення моделі.

Створення архітектури – це проектування на найвищому рівні. Логічна архітектура описує систему в термінах її принципової організації у вигляді пакетів, програмних класів і підсистем. Вона називається логічною, оскільки не визначає способи розгортання цих елементів у різних операційних системах або на фізичних комп'ютерах у мережі (це відноситься до архітектури розгортання).

Етап архітектурного проектування ІС повинен бути відображений шляхом опису процесу об'єктно-орієнтованої декомпозиції системи до рівня переліку підсистем та їх зв'язків, опису її логічної структури у вигляді пакетів компонентів (**діаграма компонентів**) [7].

2.4.1 Діаграма компонентів

Ще одним будівельним блоком для створення архітектури об'єктно-орієнтованої системи на фізичному рівні вважається компонент програмного забезпечення. Діаграма компонентів показує визначення, внутрішню структуру і залежності набору компонентів. У якості фізичних компонент можуть виступати файли, бази даних, бібліотеки, модулі, пакети, інтерфейси ІС тощо.

Компонент – елемент моделі, що представляє деяку модульну частину системи з інкапсульованим вмістом. У мові UML в якості компонента може розглядатися будь-який автономний елемент системи або підсистеми.

Внутрішні частини компонентів приховані та недоступні ззовні, окрім тих, які надаються за допомогою його інтерфейсів. Компоненти можуть бути пов'язані між собою за допомогою певних відношень. Ці відношення визначені так, щоб розглядати компонент якомога незалежнішими від свого оточення.

Для графічного представлення компонента в UML 1 використовується спеціальний символ – прямокутник зі вставленими ліворуч двома прямокутниками меншого розміру (рис. 2.47, а, б). В середині великого прямокутника записується ім'я компонента і, можливо, деяка додаткова інформація.

У мові UML 2 компонент зображується у формі прямокутника з не обов'язковим ключовим словом «component», записаним у формі стереотипу. Додатково у правому верхньому кутку символу класифікатора може бути зображена піктограма компонента у формі невеликого прямокутника з двома меншими прямокутниками (рис. 2.47, в). У середині символу прямокутника вказується ім'я компонента. Зображення компонента на діаграмі може містити також символи інтерфейсів, які приєднуються до прямокутника компонента за допомогою з'єднувачів (рис. 2.47, г).

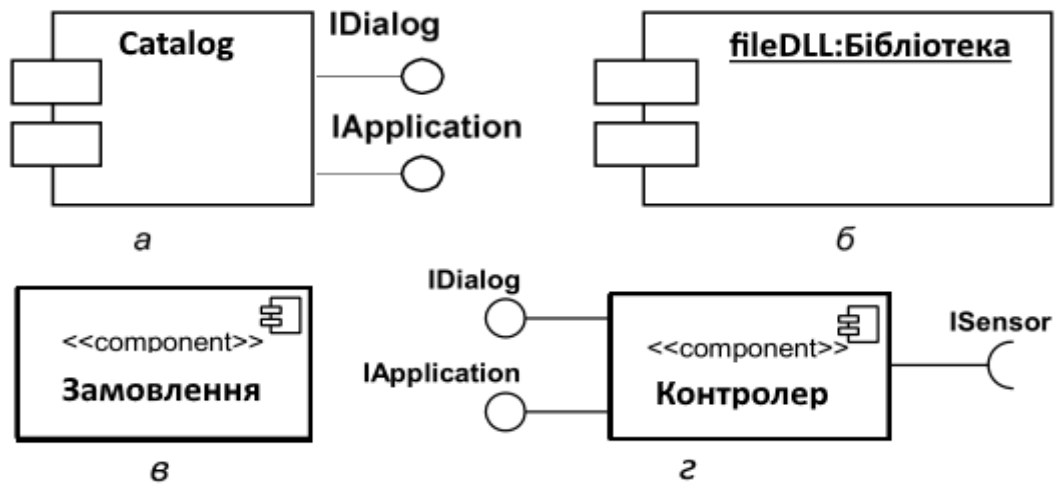


Рисунок 2.47 – Приклади графічного зображення компонент

Компонент може мати два різновиди *інтерфейсів*: інтерфейси, що надаються, і необхідні інтерфейси, які утворюють основу для взаємозв'язків компонентів між собою з використанням залежностей та з'єднувачів.

Інтерфейс, що надається, – це інтерфейс, який компонент пропонує для свого оточення.

Необхідний інтерфейс – це інтерфейс, який потрібний компоненту від свого оточення для виконання заявленої функціональності або поведінки.

Інтерфейси, що надаються, і необхідні інтерфейси, у мові UML 2 зображуються з використанням спеціальної нотації, яка дістала назву «Кулі і шарніра» або «льодяника на паличці».

Разом з інтерфейсами, які відносяться до зовнішнього представлення, на діаграмі може бути зображене внутрішнє представлення або представлення компонента у формі білої скрині. У цьому випадку усередині символу компонента зображуються класи або інші класифікатори, які реалізують поведінку цього компонента. Так, наприклад, компонент **Замовлення** має два класи, які реалізують його функціональність, а саме: **ЗаголовокЗамовлення** і **РядокТовару** (рис. 2.48) [5]. Причому ці класи пов'язані відношенням композиції.

Діаграма компонентів, на відміну від раніше розглянутих діаграм, описує особливості фізичного представлення системи та дозволяє визначити

архітектуру розроблюваної системи, встановивши залежності між програмними компонентами. У багатьох середовищах розробки модуль або компонент відповідає файлу.

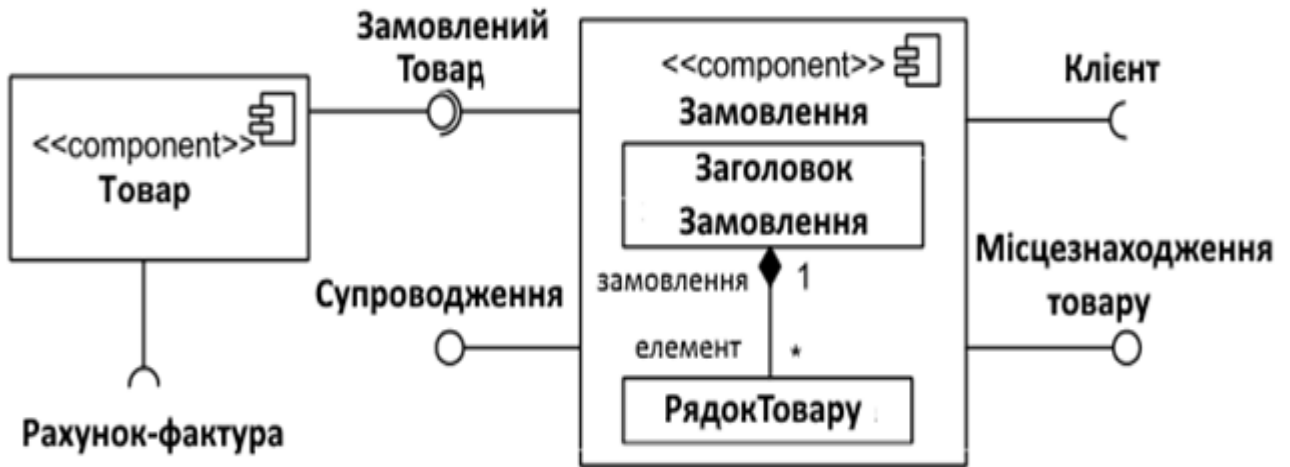


Рисунок 2.48 – Зображення компонент с інтерфейсами в нотації «кулі і шарніра»

Пунктирні стрілки, що з'єднують модулі, показують відношення взаємозалежності аналогічні тим, які мають місце при компіляції вихідних текстів програм. Основними графічними елементами діаграми компонентів є компоненти, інтерфейси і залежності між ними.

Інтерфейси також можуть бути зображені у формі звичайних прямокутників класу із стереотипом <<interface>> і секцією операцій. У цьому випадку необхідний інтерфейс позначається за допомогою *відношення залежності*, спрямований від одного з класів, що входять до його складу. Додатково це відношення може мати стереотип <<use>> (рис. 2.49, *ліворуч*). Інтерфейс, що надається, позначається за допомогою *відношення реалізації*, спрямованого від компонента або від одного з класів, що входить до його складу, до цього інтерфейсу (рис. 2.49, *справа*). Неважко помітити, що ця нотація практично ідентична нотації, яка використовується на діаграмі класів [5].

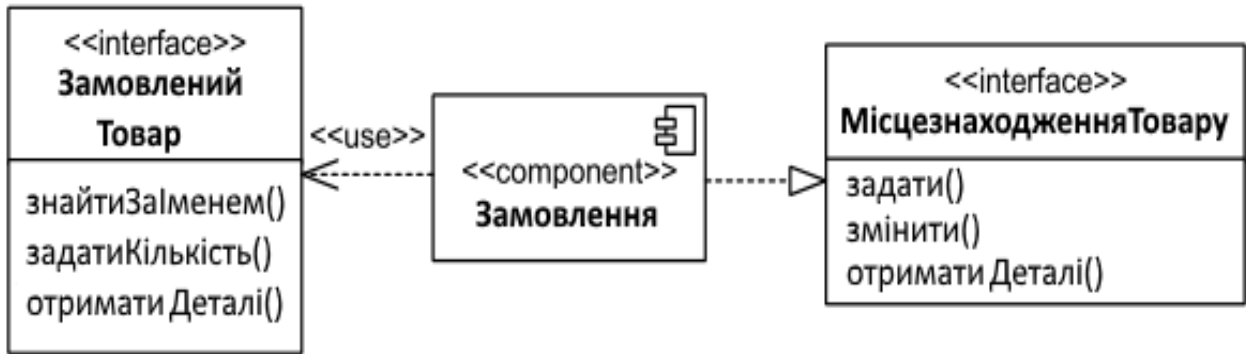


Рисунок 2.49 – Представлення інтерфейсів у формі символу класифікатора з відношеннями залежності і реалізації

На рис. 2.50 показаний приклад простої діаграми компонентів. У даному прикладі компонент **Till** (Каса) може взаємодіяти з компонентом **Sales Server** (Сервер продаж) за допомогою інтерфейсу **sales message** (повідомлення про продажі).

Оскільки мережа може бути ненадійною, то компонент **Message Queue** (Черга повідомлень) встановлений так, щоб каса могла спілкуватися з сервером, коли мережа працює, та з чергою повідомлень, коли мережа відключена.

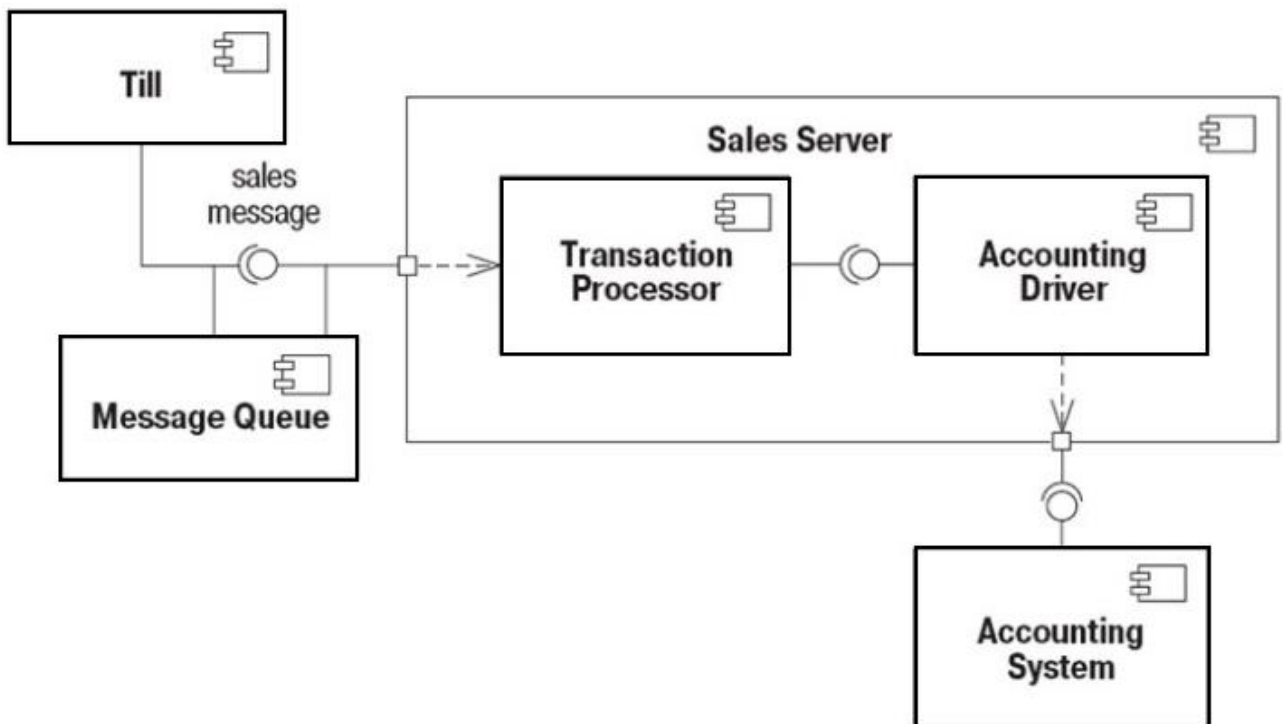


Рисунок 2.50 – Приклад простої діаграми компонентів

Тоді черга повідомлень зможе взаємодіяти з сервером, коли мережа знову стане доступною. У результаті черга повідомлень надає інтерфейс для спілкування з касою, і вимагає такий самий інтерфейс для спілкування з сервером. Сервер розділений на два основні компоненти: **Transaction Processor** (Процесор транзакцій) реалізує інтерфейс повідомлень, а **Accounting Driver** (Драйвер рахунків) спілкується з **Accounting System** (Система ведення рахунків).

Компонент у своєму складі має інтерфейсний клас, через який здійснюється доступ до інших класів об'єктів.

2.4.2 Розгортання програмної системи на апаратних засобах

Фізичне представлення програмної системи не може бути повним, якщо відсутня інформація про те, на якій платформі та на яких обчислювальних засобах вона реалізована. Якщо розробляється проста програма, яка може виконуватися локально на комп'ютері користувача, яка не використовує ніяких периферійних пристроїв і ресурсів, то в цьому випадку немає необхідності в розробці додаткових діаграм. Однак при розробці корпоративних додатків ситуація виглядає зовсім інакше.

По-перше, складні програмні системи можуть реалізовуватися в мережевому варіанті на різних обчислювальних платформах і технологіях доступу до розподілених баз даних. Наявність локальної корпоративної мережі потребує розв'язання цілого комплексу додаткових завдань щодо раціонального розміщення компонентів по вузлах цієї мережі, що визначає загальну продуктивність програмної системи.

По-друге, інтеграція програмної системи з мережею Інтернет визначає необхідність розв'язання додаткових завдань при проектуванні системи, таких як забезпечення безпеки, криптозахищеності та стійкості доступу до інформації для корпоративних клієнтів. Ці аспекти залежать від реалізації проекту у формі фізично існуючих вузлів системи, таких як сервери, робочі станції, брандмауери, канали зв'язку і сховища даних.

Нарешті, технології доступу і маніпулювання даними в рамках загальної схеми «клієнт-сервер» вимагають розміщення великих баз даних у різних сегментах корпоративної мережі, їх резервного копіювання, архівування, кешування для забезпечення необхідної продуктивності системи в цілому. Ці аспекти також вимагають візуального представлення з метою специфікації програмних і технологічних особливостей реалізації розподілених архітектур.

Розподіл елементів (результатів розробки) програмного додатку за апаратними вузлами комп'ютерної системи зручно показати за допомогою *діаграм розгортання* (у російській (українській) літературі їх називають також *діаграмами розміщення*).

З точки зору реалізації проєктована система складається з компонентів (представлених на діаграмах компонентів), розподілених по обчислювальним вузлам (представленим на діаграмах розгортання).

Діаграма розгортання дозволяє зобразити «фізичну» структуру програмної системи. На етапі проєктування діаграма розгортання використовується для представлення фізичної сукупності вузлів як основи для реалізації системи. Діаграма розгортання складається з трьох основних елементів: артефактів, вузлів і відносин між ними.

Діаграма розгортання відображає відповідність конкретних програмних артефактів (наприклад, виконуваних файлів) обчислювальним вузлам (виконуючим обробку). Вони показують розміщення програмних елементів у фізичній архітектурі системи і взаємодію (зазвичай мережеву) між фізичними елементами. Діаграма розгортання дозволяє краще зрозуміти фізичну архітектуру (або архітектуру розгортання).

Тобто, на діаграмі розгортання розміщаються тільки ті елементи фізичного представлення моделі, які існують під час виконання програмної системи. Ті елементи, які не використовуються на етапі виконання, на діаграмі розгортання, як правило, не показуються. Так, наприклад, компоненти з текстами програм можуть бути присутніми на діаграмі компонентів, а на діаграмі розгортання вони не вказуються.

Слід зазначити, що на відміну від діаграм логічного представлення і діаграм компонентів, діаграма розгортання проектується, як правило, в єдиному екземплярі, оскільки вона повинна цілком представляти особливості топології і реалізації усієї системи.

Графічно діаграма розгортання – це граф з вузлів, які з'єднані асоціаціями, що показують існуючі комунікації. Вузли можуть містити артефакти (виконуваних компонентів і динамічних бібліотек), що працюють у цих вузлах.

Вузол (node) є деяким фізично існуючим елементом системи, що має деякий обчислювальний ресурс. В якості обчислювального ресурсу вузла може розглядатися наявність щонайменше деякого об'єму електронної або магнітооптичної пам'яті і/або процесора. Поняття вузла також може включати і інші механічні або електронні пристрої, такі як датчики, принтери, модеми, цифрові камери, сканери і маніпулятори.

Графічно на діаграмі розгортання вузол зображується у формі тривимірного куба, який має власне ім'я, що вказується усередині цього графічного символу. Самі вузли можуть представлятися як в якості типів (ім'я вузла без підкреслення, рис. 2.51, *а*), так і в якості екземплярів (рис. 2.51, *б*, *в*) і стереотипів у формі рисунку (рис. 2.51, *г*). На цьому типі вузлів не можуть розміщуватися виконувані компоненти програмної системи.

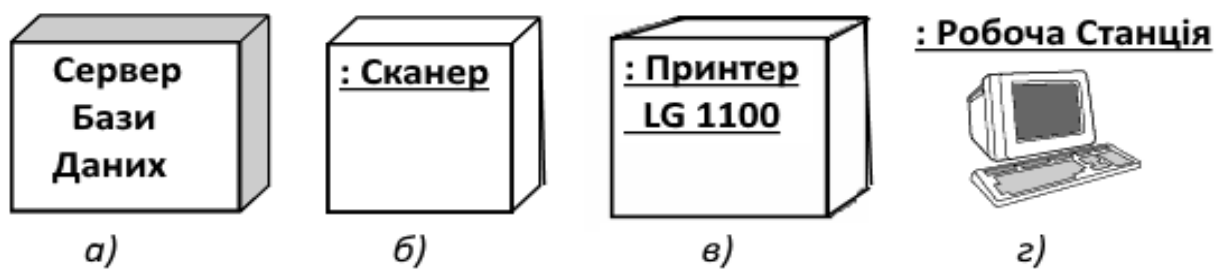


Рисунок 2.51 – Графічне зображення вузла на діаграмі розгортання

При необхідності явно вказати компоненти, які розміщуються або виконуються на окремому вузлі, можна зробити двома способами. Перший з них дозволяє розділити графічний символ вузла на дві секції горизонтальною лінією. У верхній секції записують ім'я вузла, а в нижній – розміщені на цьому вузлі компоненти (рис. 2.52, *а*).

Другий спосіб дозволяє показувати на діаграмі розгортання вузли з вкладеними зображеннями компонентів (рис. 2.52, б). Такими вкладеними компонентами можуть виступати тільки виконувані компоненти і динамічні бібліотеки.

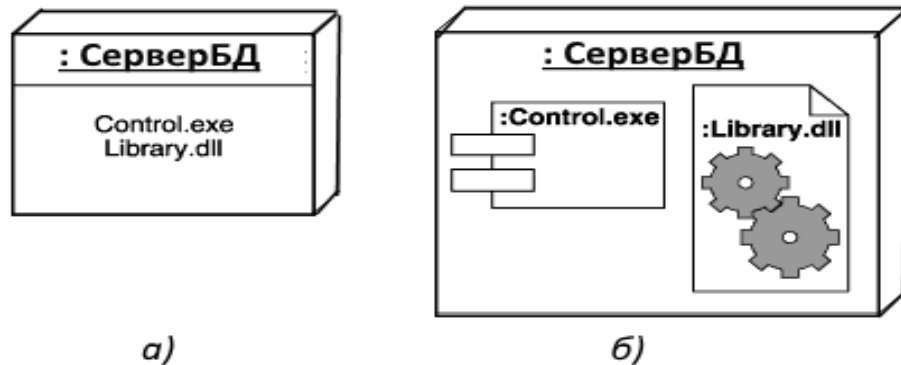


Рисунок 2.52 – Варіанти графічного зображення вузлів-екземплярів з розміщуваними на них компонентами

З'єднання (шляхи комунікації) вказують відношення між вузлами, які є різновидом асоціації. Зображаються вони відрізками ліній без стрілок. Наявність такої лінії вказує на необхідність організації фізичного каналу для обміну інформацією між відповідними вузлами.

Характер з'єднання може бути додатково специфікований приміткою, поміченою значенням або обмеженням, які визначаються згідно із загальним визначенням асоціації (рис. 2.53). У розглянутому прикладі явно визначені не лише вимоги до швидкості передачі даних у локальній мережі за допомогою поміченого значення, але і рекомендації за технологією фізичної реалізації з'єднань у формі примітки.



Рисунок 2.53 – Фрагмент діаграми розгортання зі з'єднаннями між вузлами

Окрім з'єднань на діаграмі розгортання можуть бути присутніми відношення залежності між вузлом і розгорнутими на ньому компонентами. Подібний спосіб є альтернативою вкладеному зображенню компонентів усередині символу вузла, що не завжди зручно, оскільки робить цей символ надмірно об'ємним (рис. 2.54).

Розглянемо фрагмент фізичного представлення системи віддаленого обслуговування клієнтів банку (рис. 2.55). На діаграмі розгортання вузлами системи є віддалений термінал (вузол-тип) і сервер банку (вузол-екземпляр). Вказана залежність компонента реалізації діалогу «dialog.exe» на віддаленому терміналі від інтерфейсу IAuthorise (Авторизація), реалізованого компонентом «main.exe».

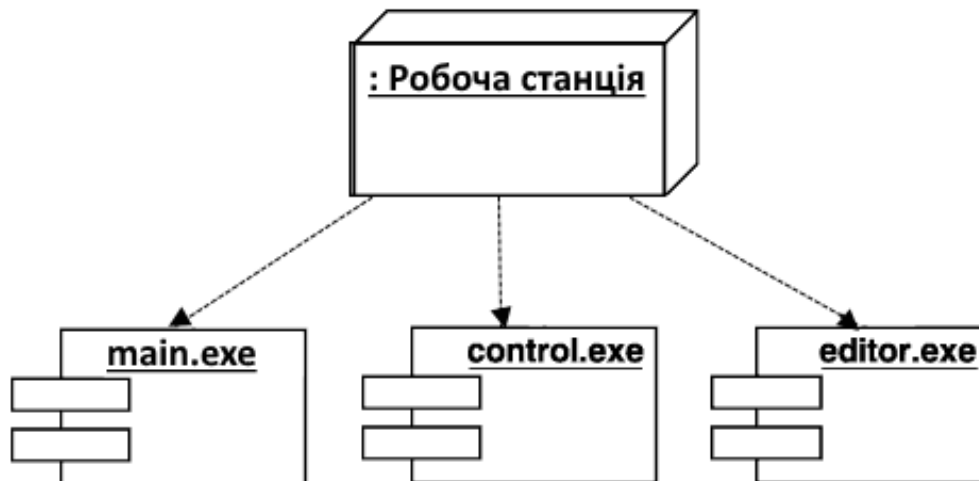


Рисунок 2.54 – Діаграма розгортання з відношенням залежності між вузлом і розгорнутими на ньому компонентами

У свою чергу, компонент «main.exe» розгорнутий на анонімному вузлі-екземплярі «Сервер банку». Останній залежить від компонента бази даних «Клієнти банку», який розгорнутий на цьому ж вузлі.

Примітка вказує на необхідність використання захищеної лінії зв'язку для обміну даними в цій системі. Інший варіант запису цієї інформації полягає в доповненні діаграми вузлом із стереотипом «закрита мережа».

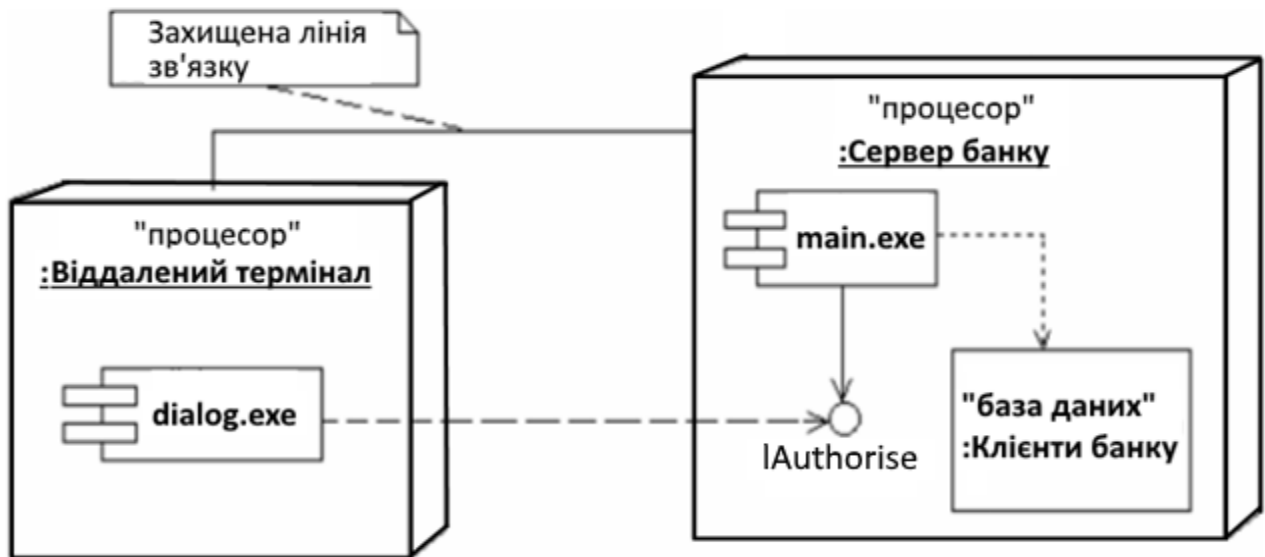


Рисунок 2.55 – Діаграма розгортання для системи віддаленого обслуговування клієнтів банку

2.5 Моделювання поведінки системи

При створенні програмної системи недостатньо відповісти на питання *що робить система?* і *з чого вона складається?* – потрібно відповісти на питання *як працює система?*. Відповідь на це питання дає *модель поведінки*. Поведінка реальної програми цілком і повністю визначається її кодом – як програма складена, так вона і виконується. Але програма – це просто запис алгоритму. Таким чином, **модель поведінки** (behavior model) – це опис алгоритму роботи системи [7, розділи 12, 18].

Поведінка системи визначається множиною об'єктів, що обмінюються повідомленнями, і задається діаграмами діяльності, діаграмами взаємодії, діаграмами кінцевого автомата.

Для опису потоку управління, тобто послідовності виконуваних елементарних кроків при виконанні окремої операції або реалізації складного варіанта використання, зручно використовувати **діаграму діяльності**.

Взаємодія декількох програмних об'єктів між собою описується **діаграмами взаємодії** в одній з двох еквівалентних форм (**діаграми послідовності** або **діаграми комунікації**).

У UML передбачено декілька різних засобів для опису поведінки. Вибір того чи іншого засобу диктується типом поведінки, яку потрібно описати.

Для опису життєвого циклу конкретного об'єкта, поведінка якого залежить від історії цього об'єкта, або обробки асинхронних стимулів, використовується кінцевий автомат у формі **діаграми станів**. При цьому стани кінцевого автомата відповідають станам об'єкта, тобто різним наборам значень атрибутів, а переходи відповідають виконанню операцій.

Як і діаграми видів діяльності, діаграми кінцевих автоматів (станів) UML відображають динамічну модель системи. За їх допомогою ілюструються події і стани об'єктів – транзакцій, прецедентів, людей тощо.

2.5.1 Діаграма діяльності

Опис функціональних вимог у вигляді варіантів використання, безсумнівно, є дуже важливим етапом у життєвому циклі розробки ПЗ. Адже саме на цьому етапі відбувається узгодження із замовником того, як буде виглядати і функціонувати система. Тепер на підставі затверджених вимог необхідно систему спроектувати і реалізувати. Завдання аналітика на цьому етапі – перетворити опис варіантів використання в технічно грамотний опис структури й поведінки системи, ясну архітекторам і розробникам. Таким чином, перехід від моделювання використання до інших видів моделювання полягає в уточненні, деталізації та конкретизації варіантів використання.

Якщо діаграма варіантів використання дає «вид зверху» на функціональність системи, то діаграма діяльності, навпаки, дозволяє докладно ілюструвати окремий варіант використання та його сценарії.

Діаграми діяльності, так само як і блок-схеми, – загальний та потужний засіб опису алгоритмів. Їх з успіхом також можна застосовувати для моделювання поведінки людей, пристроїв і організацій при виконанні бізнес-процесів.

Під діяльністю в даному випадку розуміють завдання (операцію), які необхідно виконати вручну або за допомогою засобів автоматизації. У теоретичному плані діаграма діяльності – це узагальнене уявлення алгоритму, що реалізує аналізований варіант використання. У загальному випадку діяльність є самостійним елементом поведінки, яка, у свою чергу, може включати інші діяльності або окремі дії.

Діаграми діяльності в UML забезпечені додатковими синтаксичними засобами, які різко підвищують їх виразність і область застосування.

Дія. Дія є іменованим елементом, якому відповідає один крок або етап при виконанні деякої діяльності. У результаті виконання дії може змінитися стан системи. Кожна дія в діяльності може виконуватися довільну кількість разів.

Вузол діяльності. Вузол діяльності є абстрактним класом для окремих точок в потоці діяльності, сполучених дугами. Вузли діяльності можуть відноситися тільки до окремої діяльності або групи діяльностей. Існує декілька видів конкретних вузлів діяльності, для кожного з яких є власна графічна нотація: вузол дії (рис. 2.56, а), вузол виклику діяльності (рис. 2.56, б), вузол об'єкту (рис. 2.56, в) і вузли управління (рис. 2.56, г).

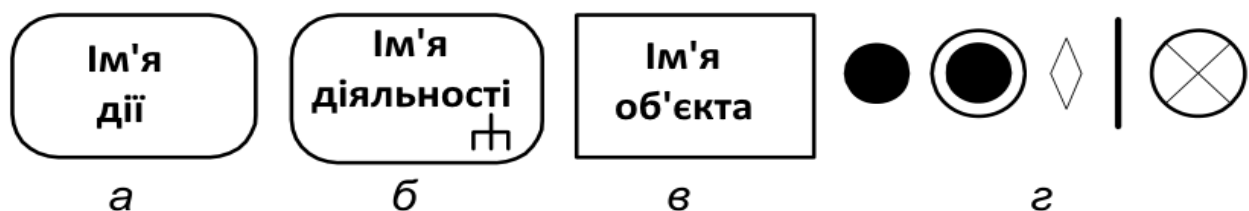


Рисунок 2.56 – Нотація для різних вузлів діяльності

Вузол виклику діяльності зображується розміщенням символу «граблі», які мають подібність з символом ієрархії, ілюструючи той факт, що цей виклик діяльності починає іншу діяльність. Остання може бути представлена у формі додаткової вкладеної діаграми діяльності.

Дуга діяльності. На діаграмі вузли діяльності можуть бути пов'язані між собою за допомогою дуг діяльності. Дуга діяльності є спрямованим відношенням і має в якості джерела та мети деякі вузли.

Дуга діяльності зображується суцільною лінією з тонкою стрілкою, при цьому ім'я дуги може бути відсутнім (рис. 2.57, *a*). Якщо ж дуга має ім'я, то воно записується з рядкової букви і вказується поблизу стрілки (рис. 2.57, *б*). Дуга діяльності може бути також зображена з використанням з'єднувача, який представляється у формі невеликого круга з ім'ям мітки (цифри або букви) для цієї дуги (рис. 2.57, *в*). Це позначення прийняте виключно з метою зручності й не робить впливу на модель.

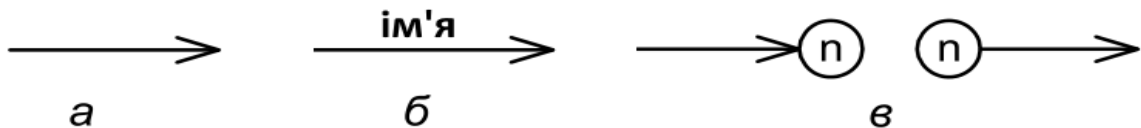


Рисунок 2.57 – Нотація для різних дуг діяльності

Вузли розгалуження і злиття. Вузол розгалуження має одне ребро, що входить, і два й більше альтернативних ребер, що виходять (рис. 2.58, *a*). Маркер, що поступає по ребру буде запропонований усім ребрам, що виходять, але пройде тільки по одному з них. Вузол розгалуження – це перехрестя потоків, на якому маркер повинен вибрати тільки один шлях. Кожне вихідне ребро захищене сторожовою умовою, яка означає, що ребро прийме маркер тільки в разі виконання сторожової умови. У вузлах злиття сходяться два або більше ребер і виходить лише одне ребро. Вони об'єднують усі потоки, що входять, в один потік, що виходить (рис. 2.58, *б*).

Вузли розгалуження і об'єднання (паралелізм). Вузол розгалуження розділяє потік на декілька паралельних потоків (рис. 2.58, *в*). Вузол об'єднання синхронізує і об'єднує декілька потоків в єдиний потік, що виходить (рис. 2.58, *г*).

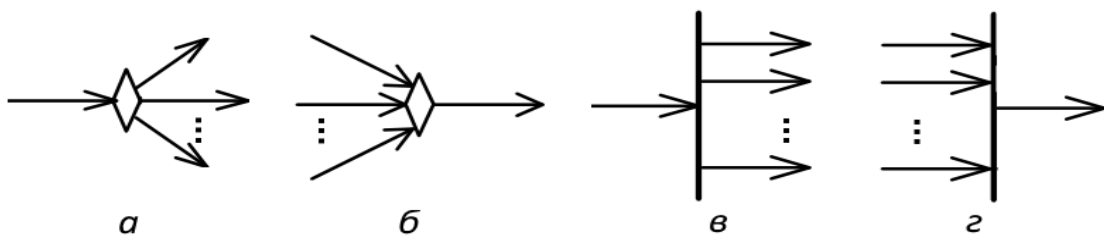


Рисунок 2.58 – Вузли розгалуження (*a*), злиття (*б*), паралельне розгалуження (*в*) і об'єднання (*г*)

На рис. 2.59 наведено діаграму діяльності, побудовану на прикладі процесу обробки звернення клієнта до менеджера інтернет-магазину і подальшого оформлення замовлення.

У загальному випадку дії на діаграмі діяльності виконуються над тими або іншими об'єктами. Ці об'єкти або ініціюють виконання дій, або визначають деякий результат цих дій. При цьому дії специфікують виклики, які передаються від одного об'єкту графа діяльності до іншого.

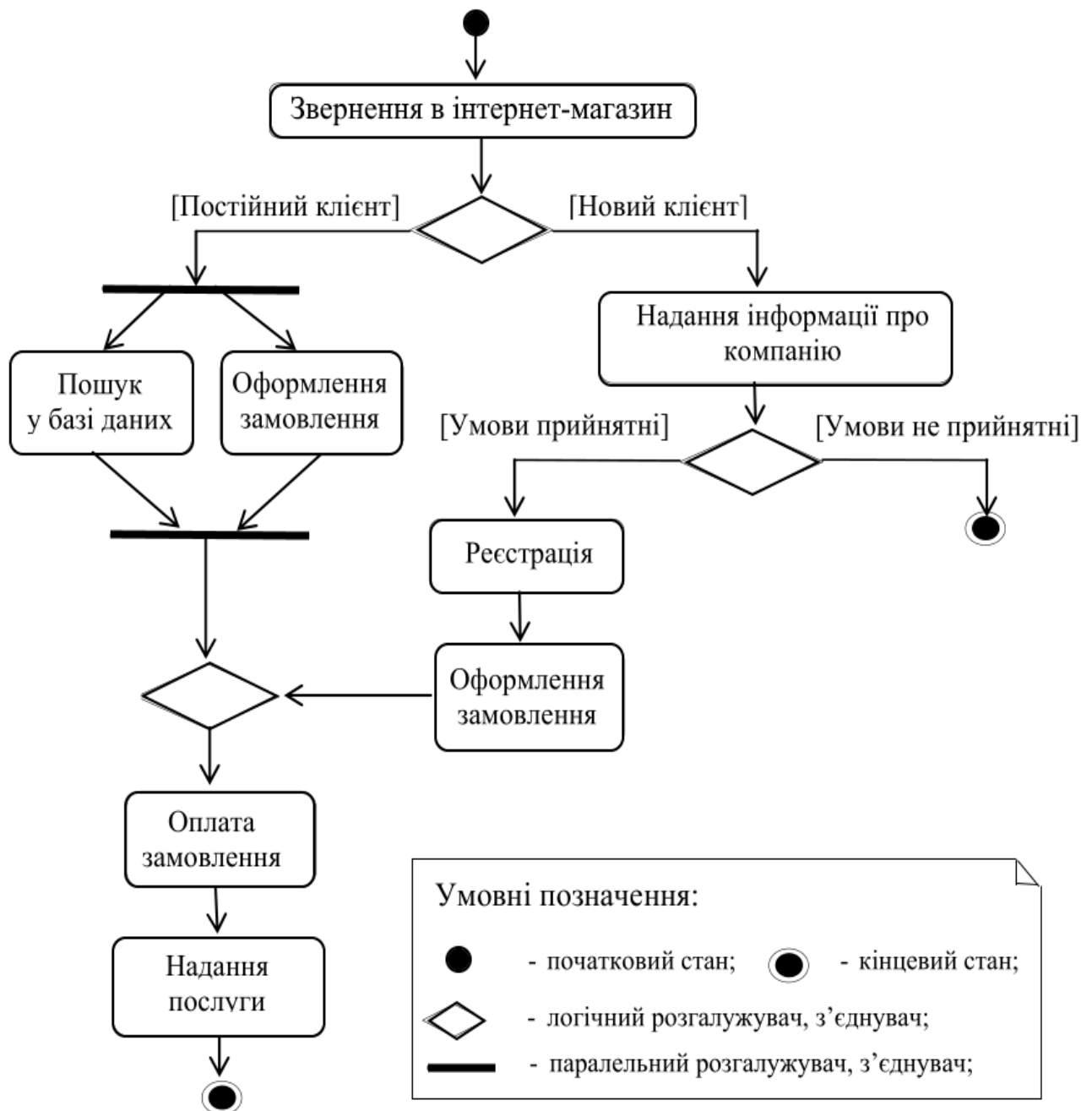


Рисунок 2.59 – Приклад діаграми діяльності

Для детальної ілюстрації особливості зображення розгалуження і паралельних підпроцесів розглянемо ще один приклад з вибором напою і приготуванням кави. Особливість цього прикладу полягає в тому, що він не вимагає ніяких додаткових пояснень у силу очевидності свого контексту (рис. 2.60) [5]. Паралельність у даному прикладі проявляється в тому, що пошуком чашки бармен може зайнятися під час приготування кави.

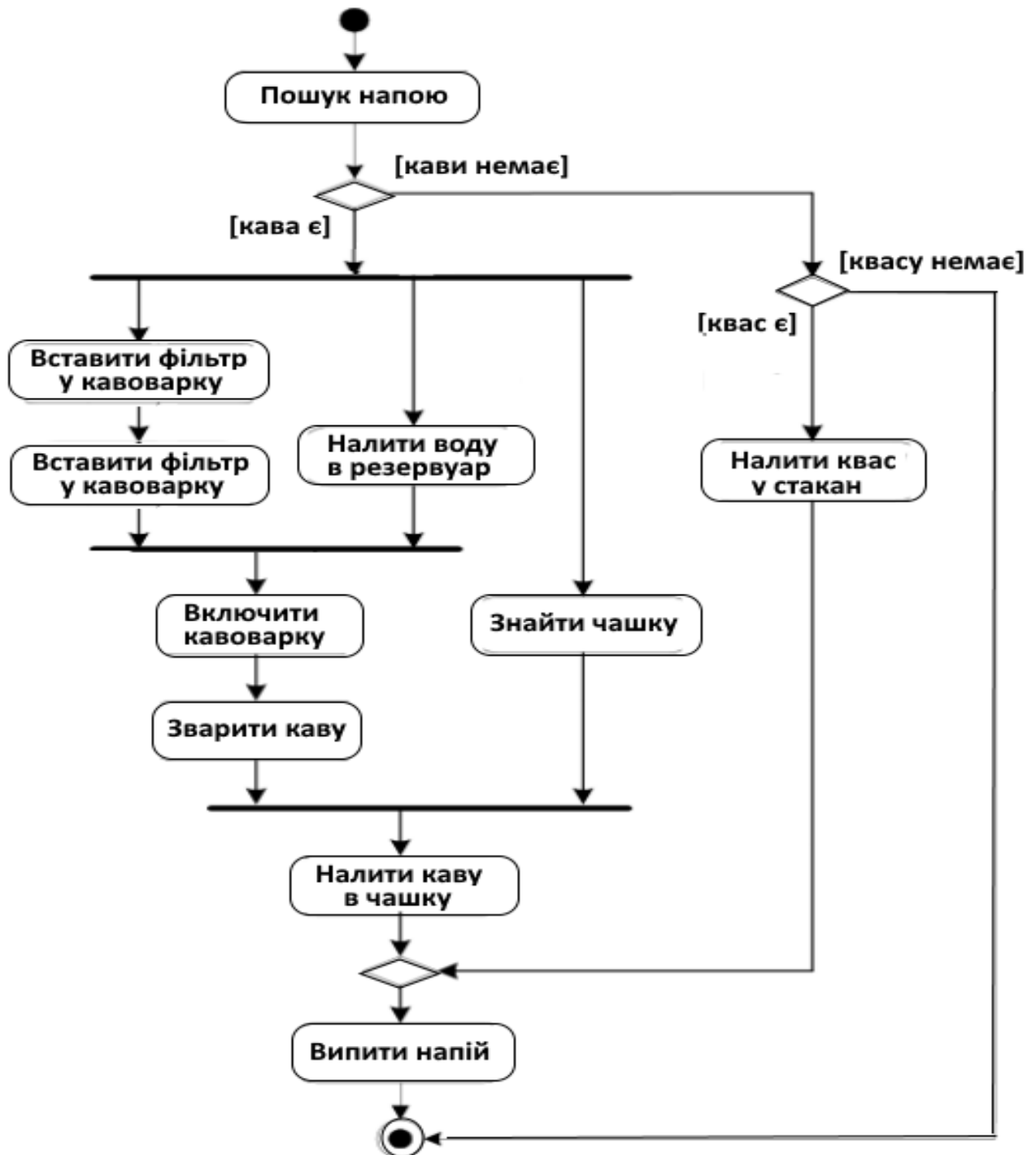


Рисунок 2.60 – Діаграма діяльності для прикладу з приготуванням напою

Діаграми діяльності можуть бути використані не лише для специфікації алгоритмів обчислень або потоків управління у програмних системах. Не менш важлива сфера їх застосування пов'язана з моделюванням бізнес-процесів. Для моделювання цих особливостей в мові UML використовується спеціальна конструкція, що отримала назву *доріжки* (рис. 2.61).

При цьому усі стани дії на діаграмі діяльності діляться на окремі групи, які відділяються один від одного вертикальними лініями. Дві сусідні лінії і утворюють доріжку, а група станів між цими лініями виконується окремим підрозділом компанії (Відділ замовлень, Відділ продажів, Склад).

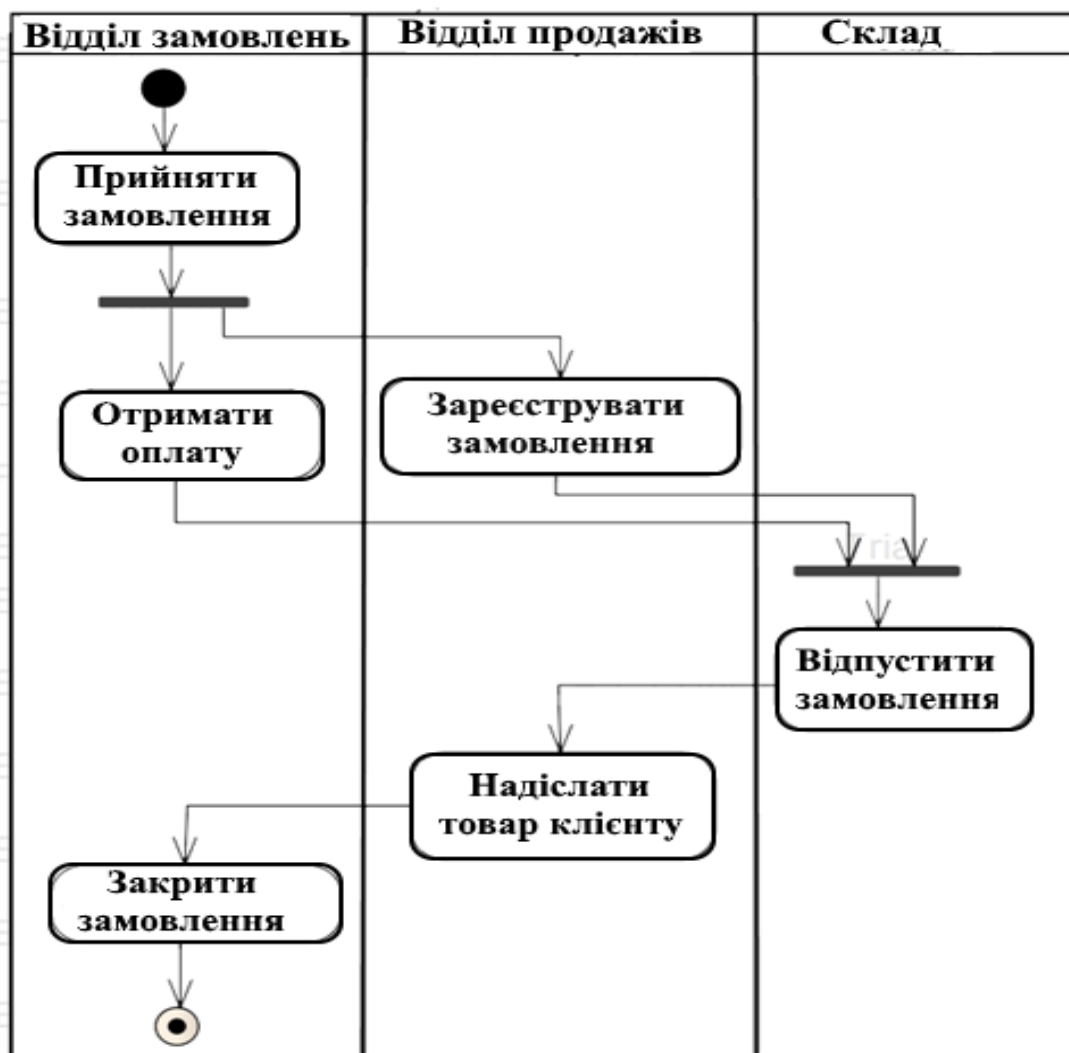


Рисунок 2.61 – Варіант діаграми діяльності з доріжками

Підрозділами компанії є відділ прийому й оформлення замовлень, відділ продаж і склад. Цим підрозділам відповідатимуть три доріжки на діаграмі

діяльності, кожна з яких специфікує зону відповідальності підрозділу. У цьому випадку діаграма діяльності містить у собі не лише інформацію про послідовність виконання робочих дій, але й про те, який підрозділ торгової компанії повинен виконувати ту або іншу дію.

Відділом прийому і оформлення замовлень фіксується отримане від клієнта замовлення, після чого він реєструється у відділі продажів. Потім здійснюється розпаралелювання діяльності на два потоки. Перший з них залишається в цьому ж відділі та пов'язаний з отриманням оплати від клієнта за замовлений товар. Другий ініціює виконання дії з реєстрації замовлення у відділі продаж (модель товару, розміри, рік випуску та ін.) і передачі товару на склад.

Проте видача товару зі складу починається тільки після того, як від клієнта буде отримана плата за товар (перехід-злиття). Потім виконується підготовка товару до відправки і його відправка клієнтові у відділі продаж. Після завершення цих дій замовлення закривається у відділі прийому і оформлення замовлень.

2.5.2 Діаграма послідовності

Діаграми взаємодії призначені для моделювання поведінки шляхом опису взаємодії об'єктів для виконання деякої задачі або досягнення певної мети. У програмній системі об'єкти повинні взаємодіяти один з одним, посилаючи повідомлення. **Взаємодія** – це така поведінка на основі обміну повідомленнями між набором об'єктів, яка забезпечує реалізацію вимог до системи.

Діаграми взаємодії застосовуються на різних рівнях моделювання: як для опису поведінки окремих операцій, так і цілих варіантів використання. Даний тип діаграм дозволяє описувати не тільки взаємодію програмних об'єктів (примірників класів), але й взаємодію примірників інших класифікаторів: дійових осіб, варіантів використання, компонентів тощо.

Діаграми взаємодії зображуються в декількох різних графічних формах, з яких найважливішими є **діаграми послідовності** і **діаграми комунікації**.

Діаграма послідовності та діаграма комунікації семантично еквівалентні, хоча графічно виглядають зовсім по-різному. Семантично ці діаграми еквівалентні тому, що описують одне і те ж: послідовність передачі повідомлень між об'єктами у процесі їх взаємодії.

Виглядають по-різному вони тому, що в діаграмі послідовності графічно підкреслюється впорядкованість у часі переданих повідомлень, у той час як в діаграмі комунікації на передній план висувається структура зв'язків між об'єктами, до яких передаються повідомлення.

Діаграма послідовності призначена для моделювання поведінки у формі опису протоколу сеансу обміну повідомленнями між взаємодіючими екземплярами класифікаторів під час виконання одного з можливих сценаріїв.

На діаграмі послідовності вважається виділеним один напрямок, відповідно до напрямку часу. За замовчанням прийнято, що час протікає зверху донизу, але це не обов'язково, наприклад, можна вважати, що час протікає зліва направо, обумовивши це спеціальним коментарем.

Паралельно вісі часу від усіх учасників взаємодії об'єктів відходить пряма пунктирна лінія, яка називається **лінією життя**. Лінія життя представляє об'єкт у взаємодії. Якщо стрілка відходить від лінії життя об'єкта, то це означає, що даний об'єкт відправляє повідомлення, а якщо стрілка повідомлення входить у лінію життя, то це означає, що даний об'єкт отримує повідомлення. Якщо ж стрілка перетинає лінію життя об'єкта, то це нічого не значить – повідомлення пролетіло повз. Якщо у процесі взаємодії об'єкт припиняє своє існування, то лінія життя обривається і в цьому місці ставиться косий хрест. Над стрілкою повідомлення вказується текстова частина.

Фокус управління призначений для виділення активності об'єктів і зображується на діаграмі у вигляді вузького прямокутника, верхня сторона якого є початком отримання фокусу управління об'єкту (початок активності), нижня сторона – закінченням фокусу управління (закінчення активності). Періоди активності можуть чергуватися з періодами пасивності або очікування (декілька фокусів управління).

Ініціатором взаємодії в системі може бути *актор* або зовнішній користувач. Актор може мати власне ім'я або залишатися анонімним. Найчастіше актор і його фокус управління існують у системі постійно.

Об'єкт може ініціювати рекурсивну взаємодію з самим собою. На діаграмі послідовності *рекурсія* – це невеликий прямокутник, приєднаний до правої сторони фокусу управління того об'єкту, для якого зображується ця рекурсивна взаємодія (рис. 2.62).



Рисунок 2.62 – Рекурсія на діаграмі послідовності

Повідомлення – закінчений фрагмент інформації, що направляється одним об'єктом іншому. Прийом повідомлення ініціює виконання певних дій, спрямованих на вирішення окремої задачі тим об'єктом, якому це повідомлення відправлене. Повідомлення не лише передає деяку інформацію, але й вимагає або припускає виконання очікуваних дій від приймаючого об'єкту.

Повідомлення можуть мати аргументи або параметри, залежно від конкретних значень яких може бути отриманий різний результат. Значення параметрів окремих повідомлень можуть містити умовні вирази – розгалуження або альтернативні шляхи основного потоку управління.

Повідомлення зображаються горизонтальними стрілками, що сполучають лінії життя або фокуси управління двох об'єктів на діаграмі послідовності.

Різні типи повідомлень зображені на рис. 2.63 (*a* – синхронний виклик – відправка повідомлення і очікування реакції на нього; *b* – асинхронний виклик – відправка повідомлення і подальше продовження роботи без очікування реакції; *c* – повідомлення реакції – показує повідомлення у відповідь для повідомлення синхронного виклику; *d* – повідомлення вилучення – використовується для вилучення лінії життя іншого об'єкту).

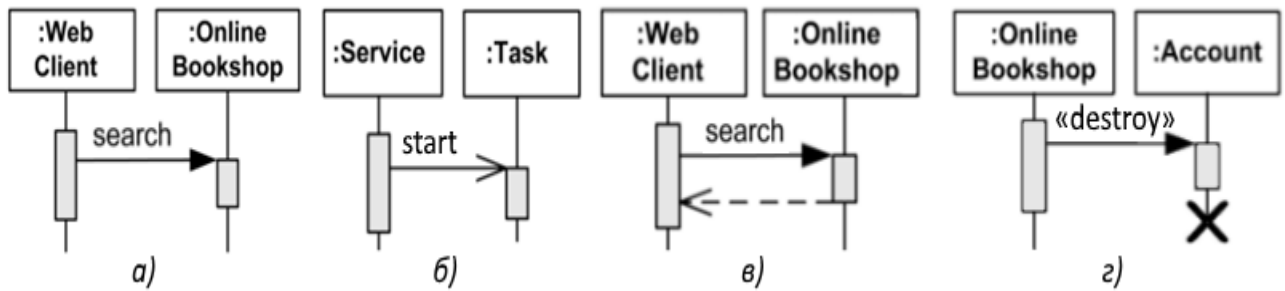


Рисунок 2.63 – Типи повідомлень на діаграмі послідовності

Коментарі або примітки можуть включатися в діаграми послідовності, асоціюючись з окремими об'єктами або повідомленнями. Використовується стандартне позначення для коментаря у вигляді прямокутника із загнутих правим верхнім кутом.

Для запису часових обмежень використовуються фігурні дужки, які розміщуються поряд з початком стрілки відповідного повідомлення. Часові обмеження можуть відноситися до виконання певних дій об'єктами і до повідомлень, специфікуючи умови їх передачі або прийому, наприклад:

- { час_прийому_повідомлення, час_відправки_повідомлення < 1 с };
- { час_очікування_відповіді < 5 с };
- { час_передачі_паketу < 10 с };
- { об'єкт_1. час_подання_сигналу_тривоги > 30 с }.

Графічно діаграма послідовності – це двовимірна схема, яка показує узагальнені об'єкти (ролі), які розміщені вздовж горизонтальної осі, і повідомлення, впорядковані за часом, уздовж вертикальної осі (лінії життя). Прецеденти визначають, як зовнішні виконавці взаємодіють з програмною системою. У процесі цієї взаємодії виконавцем генеруються системні події, які являють собою запити на виконання деякої системної операції.

На діаграмі послідовності зображуються виключно ті об'єкти, які безпосередньо беруть участь у взаємодії і не показують можливі статичні асоціації з іншими об'єктами.

Для діаграми послідовності ключовим моментом є саме динаміка взаємодії об'єктів у часі та має як би два виміри.

Один вимір – зліва направо у вигляді вертикальних ліній, кожна з яких зображує лінію життя окремого об'єкта, що бере участь у взаємодії. Графічно кожен об'єкт зображується прямокутником і розташовується у верхній частині своєї лінії життя.

Другий вимір діаграми послідовності – це вертикальна часова вісь, спрямована зверху вниз. Початковому моменту часу відповідає сама верхня частина діаграми. При цьому взаємодія об'єктів реалізується за допомогою повідомлень, які надсилаються одними об'єктами іншим (рис. 2.64).

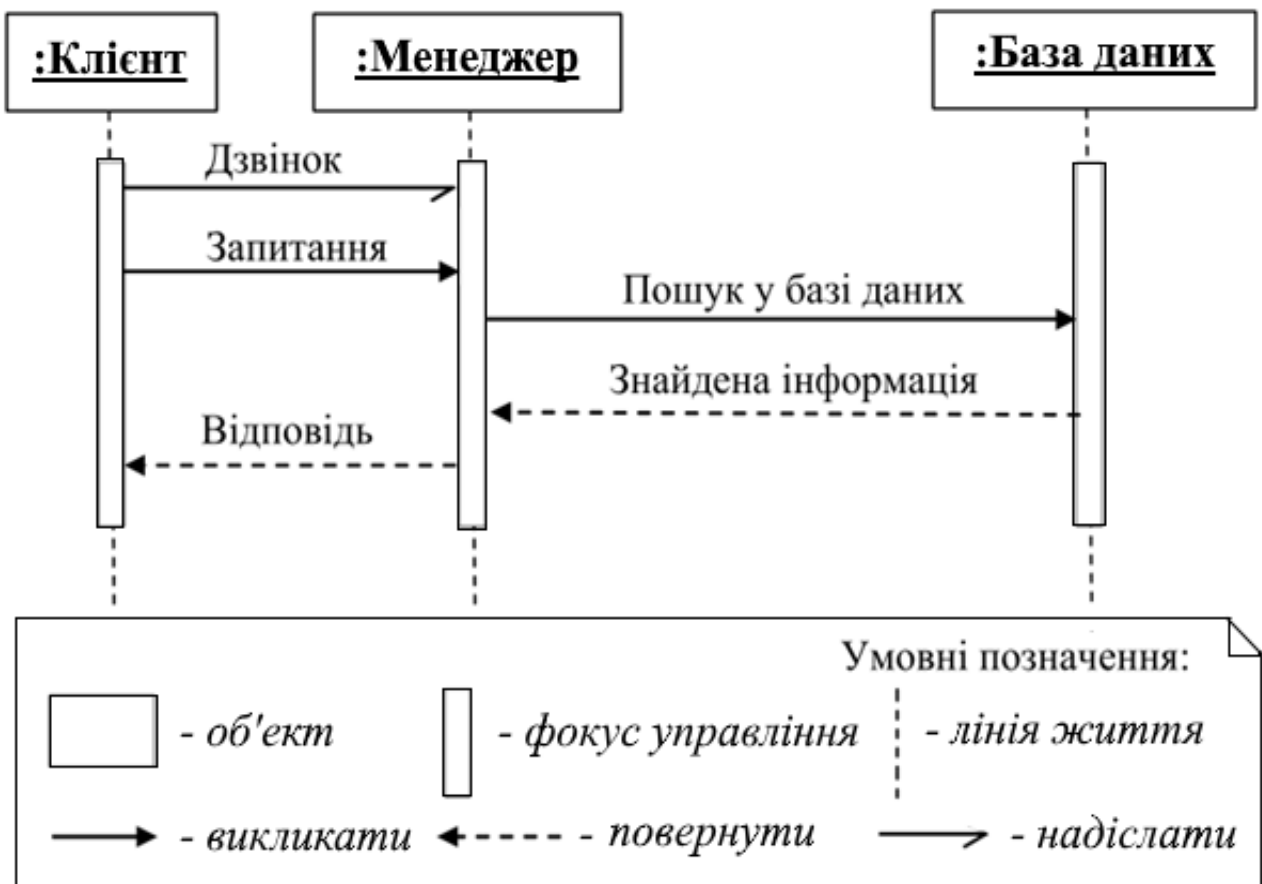


Рисунок 2.64. Фрагмент діаграми послідовності

Вже було визначено класи сценарію **Оформлення замовлення** з наведеного прикладу на діаграмі класів, тепер за допомогою діаграми послідовності покажемо, як взаємодіють об'єкти цих класів в часі.

Складемо діаграму послідовності для випадку, коли покупець успішно оформляє замовлення (рис. 2.65).

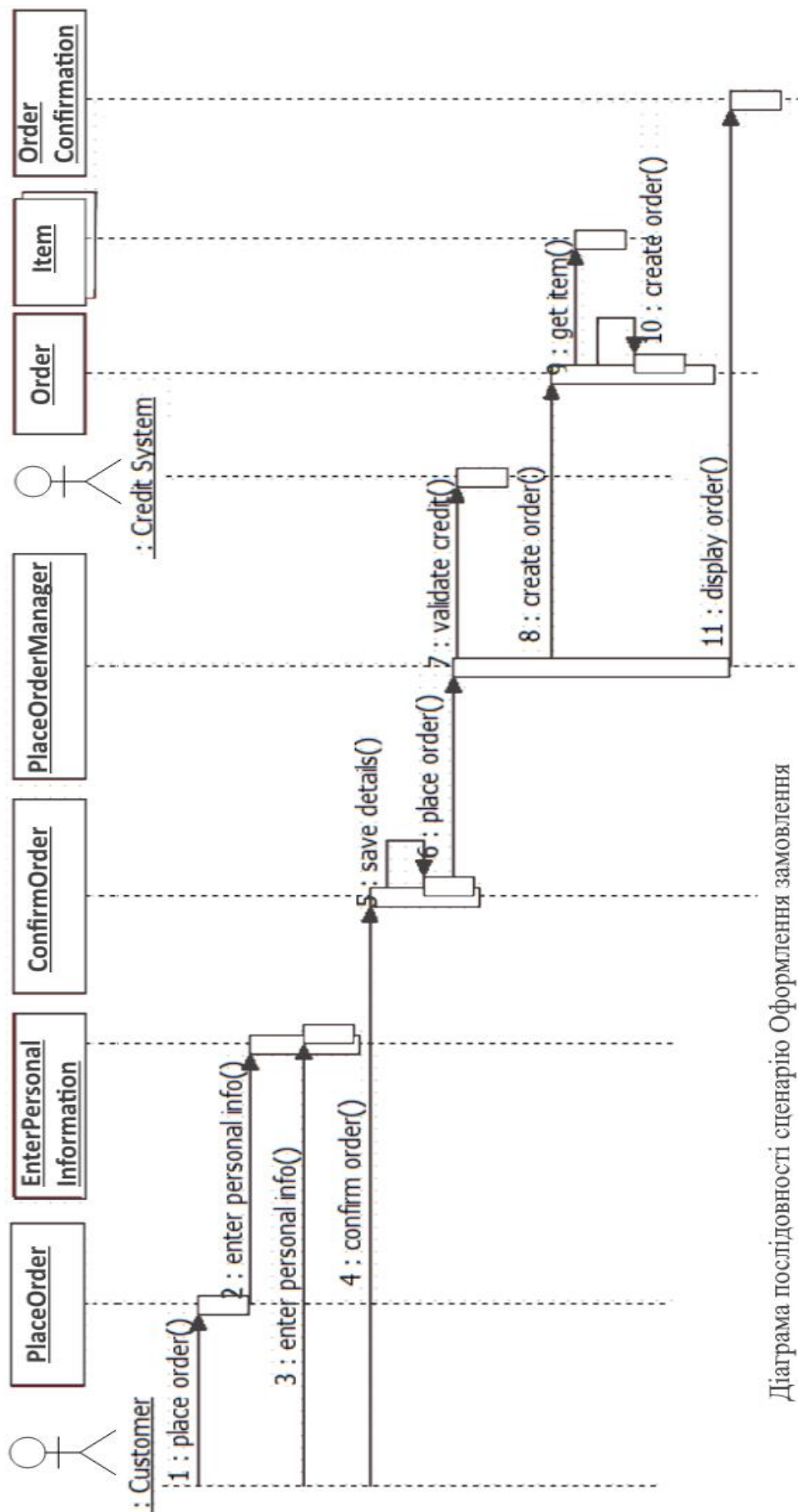


Рисунок 2.65 – Діаграма послідовності сценарію Оформлення замовлення

Покупець вибирає опцію «Оформити замовлення» (place order), при цьому викликається деякий об'єкт **PlaceOrder** (це буде граничний об'єкт, що належить відповідному граничному класу). Далі відкривається форма введення особистих даних покупця і його кредитної картки (**EnterPersonalInformation**), на ній покупець вводить своє ім'я, адресу, телефон, адресу електронної пошти (enter personal info) і кредитні дані.

Інформація приймається і відкривається форма підтвердження замовлення (**ConfirmOrder**), покупець підтверджує, що згоден з реквізитами замовлення (confirm order), деталі замовлення зберігаються для подальшого використання (save the details). Фокус управління передається деякому управляючому об'єкту (**PlaceOrderManager**), який звертається до зовнішньої кредитної системи (**Credit System**) для проведення платежу. Якщо платіж пройшов успішно (а саме такий сценарій ми зараз і розглядаємо), то **PlaceOrderManager** посилає повідомлення (create order) створити об'єкт **Замовлення (Order)**, потім викликає форму підтвердження замовлення (**OrderConfirmation**). Об'єкт **Замовлення (Order)** звертається до об'єктів **Товар (Item)** для того, щоб отримати інформацію про товари і створює замовлення. Процес завершується.

Зверніть увагу, що символ об'єкту **Товар(Item)** на діаграмі послідовності відрізняється від символів інших об'єктів, оскільки був завданий множинний екземпляр класу. Дійсно, замовлення може складатися з декількох товарів, це означає, що об'єкту **Замовлення(Order)** необхідно отримати інформацію про декілька об'єктів **Товар(Item)**. Для цього використовуємо нотацію UML множинного екземпляра класу, який представляє одним значком декілька об'єктів.

Діаграми послідовності вважають найпопулярнішим засобом представлення детальних вимог на етапі аналізу. Мета перетворення – підвищити рівень точності, формалізувати вимоги, привести їх до виду, необхідного для розв'язання задач проектування. На рис. 2.66 приведений приклад ще одної детальної діаграми послідовності – «Зняття грошей через банкомат».

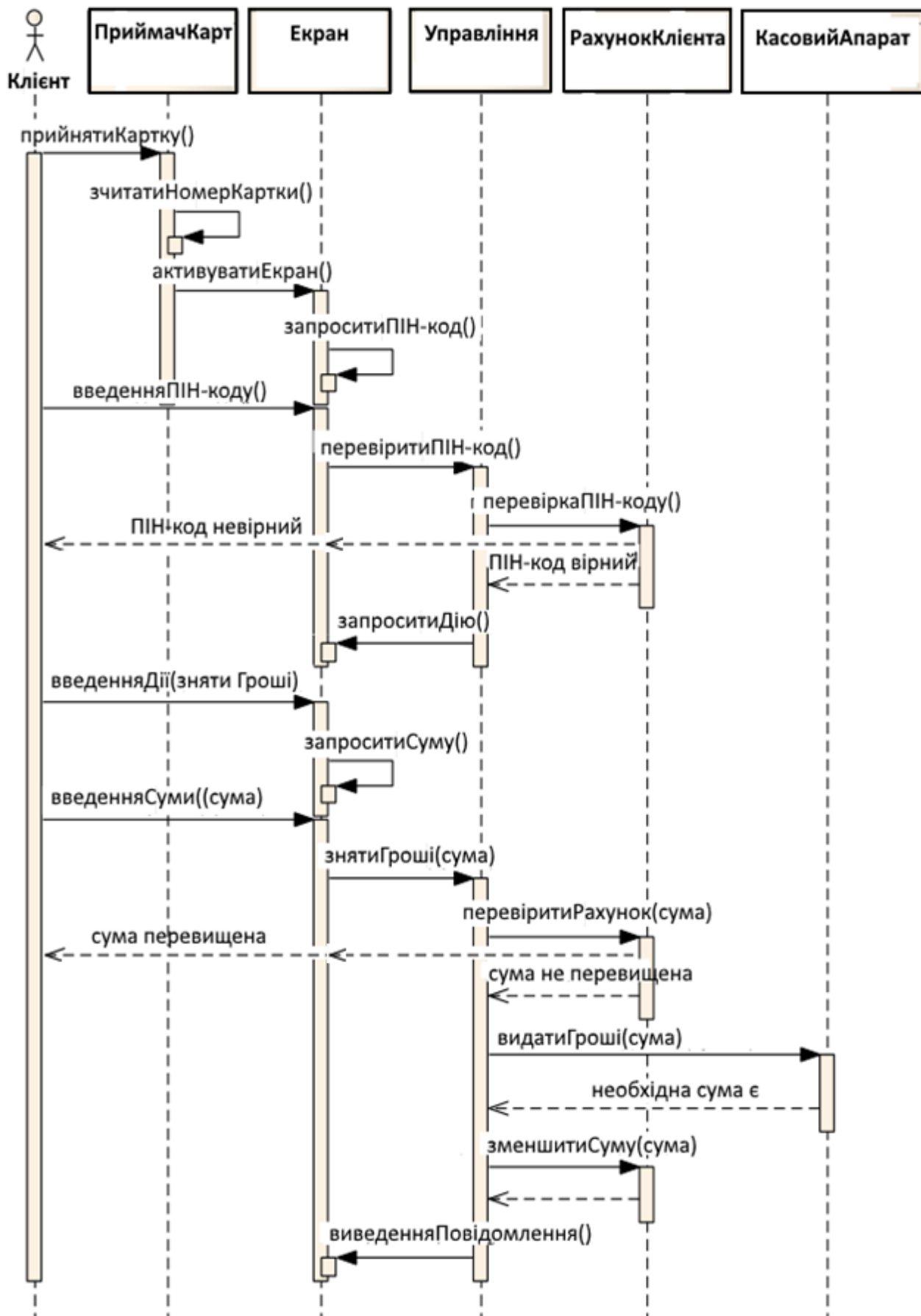


Рисунок 2.66 – Діаграми послідовності «Зняття грошей через банкомат»

2.5.3 Діаграма комунікації

Діаграма комунікації (в UML 1 – **діаграма кооперації**), так само як і діаграма послідовності описує поведінку як взаємодію, тобто як протокол обміну повідомлень між об'єктами. Один і той же самий об'єкт може брати участь у різних взаємодіях, граючи в них різні ролі. Таким чином, взаємодія завжди відбувається в певному контексті, який визначається множиною учасників у взаємодії об'єктів і зв'язків.

Замість того щоб малювати кожного учасника у вигляді лінії життя і показувати послідовність повідомлень, розташовуючи їх по вертикалі, як це робиться в діаграмах послідовності, комунікаційні діаграми допускають довільне розміщення великої кількості об'єктів, дозволяючи малювати зв'язки, що показують відношення об'єктів, і використовувати нумерацію для представлення послідовності повідомлень.

Для створення діаграми комунікації треба розташувати об'єкти, що беруть участь у взаємодії, у вигляді вершин графа. Потім зв'язки, що з'єднують ці об'єкти, зображаються у вигляді дуг цього графа. Зв'язки доповнюються повідомленнями, які об'єкти приймають і посилають повідомлення.

Діаграма комунікації характеризуються двома особливостями, що відрізняють їх від діаграм послідовностей: *шляхом (зв'язком)* і порядковим номером повідомлення. Числа перед іменами повідомлень вказують на відносну послідовність повідомлень.

Шлях або зв'язок (Link) використовується для опису зв'язку одного об'єкту з іншим. Зв'язки не мають власних імен і кратності, але можуть використовувати такі стереотипи:

- «association» – вказує, що зв'язок є асоціацією;
- «parameter» – вказує, що зв'язок є параметром деякого методу;
- «local» – вказує, що зв'язок є локальною змінною методу, зона видимості якої обмежена тільки сусіднім об'єктом;

- «global» – вказує, що зв'язок є глобальною змінною, зона видимості якої поширюється на усю діаграму комунікації;
- «self» – вказує, що зв'язок є зв'язком рефлексії об'єкту з самим собою (зображується петлею у верхній частині прямокутника об'єкту).

Порядковий номер повідомлення використовується для позначення тимчасової послідовності повідомлень.

Для сценарію **Оформлення замовлення**, для якого вже було складено діаграму послідовності, побудуємо діаграму кооперації, помістивши на неї усі ті ж самі об'єкти (рис. 2.67).

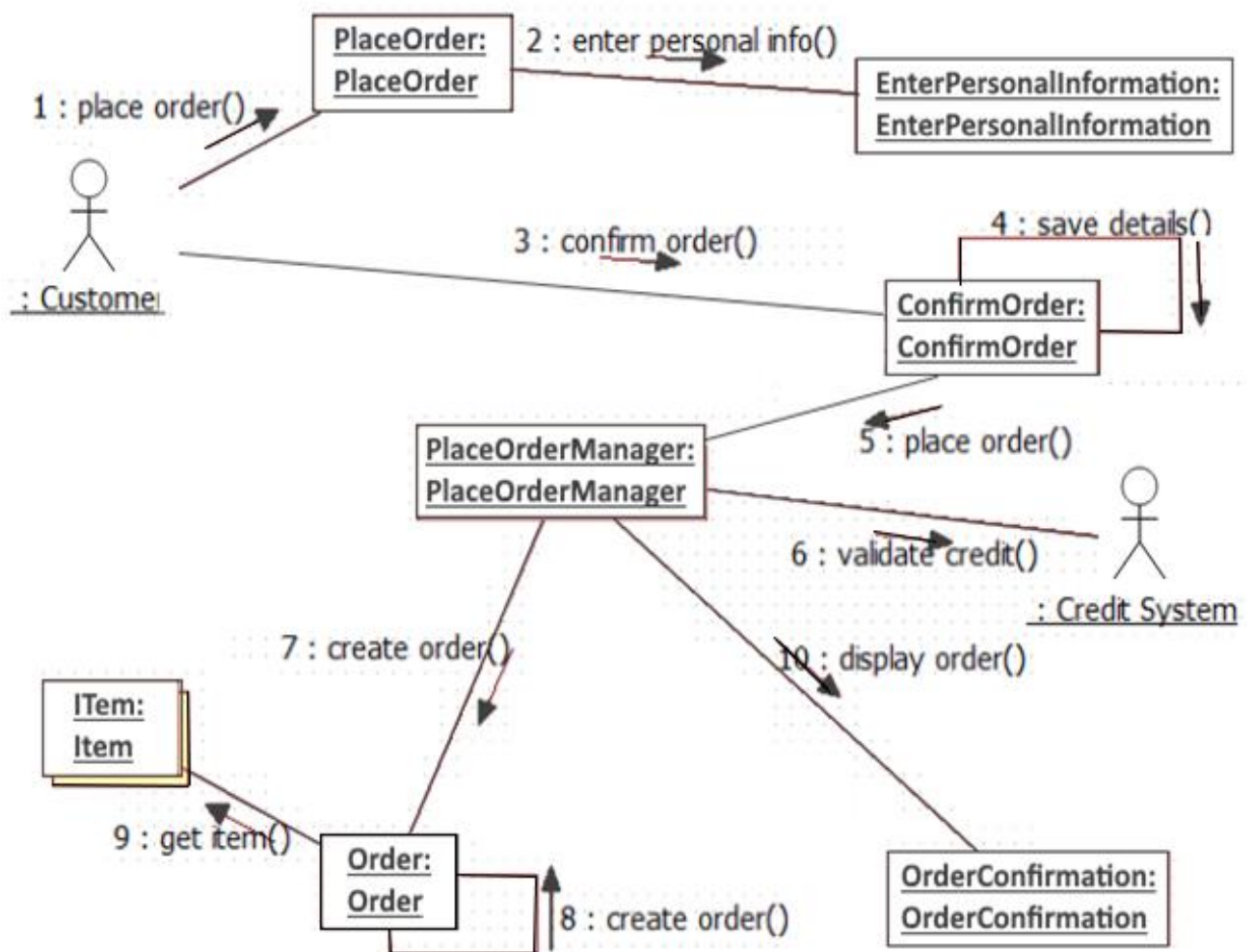


Рисунок 2.67 – Діаграма комунікації сценарію Оформлення замовлення

Розглянемо ще один приклад діаграми комунікації, як варіант моделі комунікації внутрішніх частин банкомату при знятті готівки (рис. 2.68) [5].

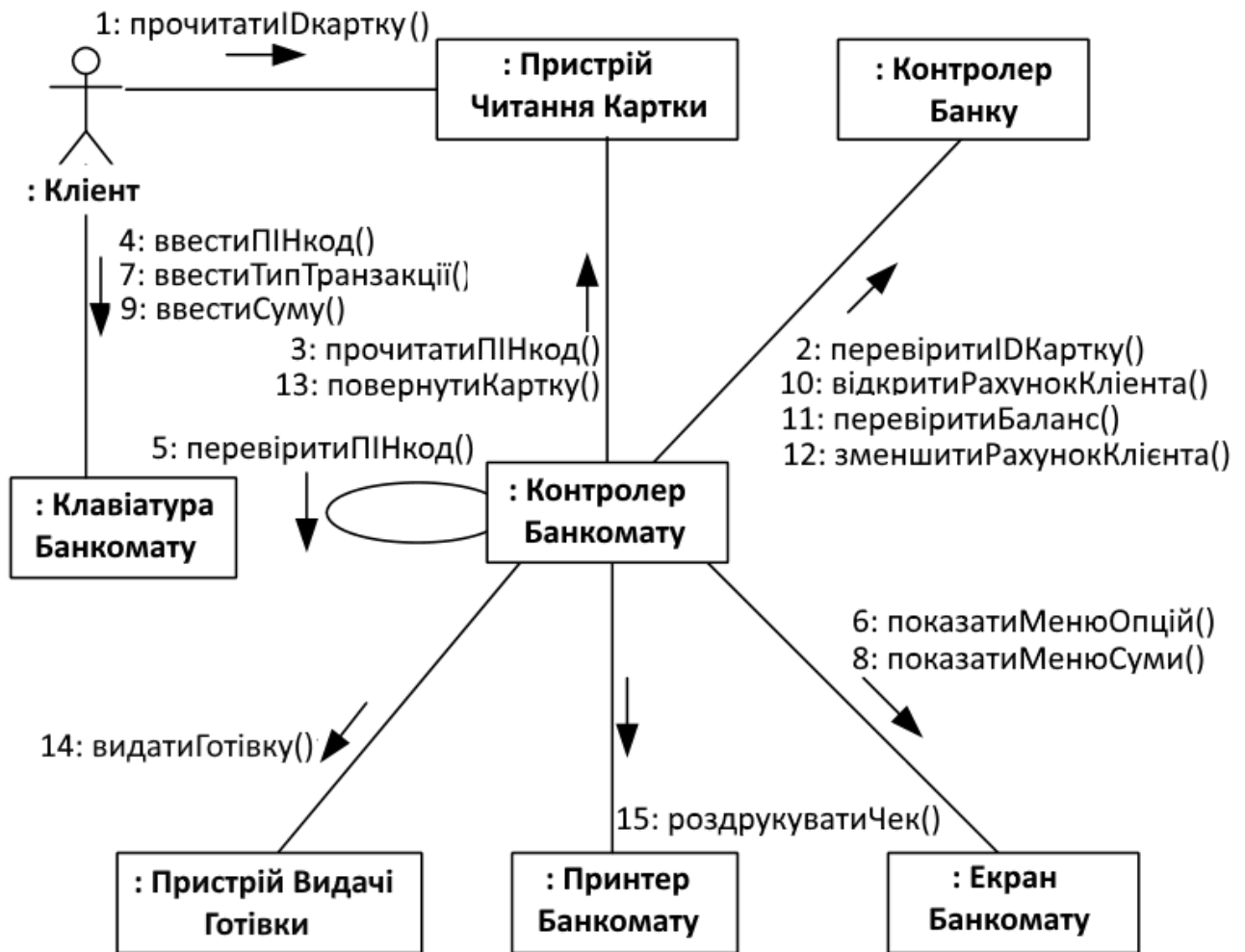


Рисунок 2.68 – Діаграма комунікації для банкомату

Усі зв'язки на діаграмах комунікації специфіковані, на їх кінцях вказана інформація у формі ролей зв'язків. Щоб відобразити структурну організацію потоків управління, на діаграмі зображені усі повідомлення з вказівкою їх порядку і семантичних особливостей.

2.5.4 Діаграма кінцевого автомату

Діаграма кінцевого автомата (діаграма станів) відображає кінцевий автомат, виділяючи потік управління, наступний від стану до стану. Кінцевий автомат – поведінка, яка визначає послідовність зміни станів у ході існування об'єкту. Більш точно, діаграма визначає для кожного стану об'єкту дію, яка виконується об'єктом при отриманні ним сигналу про подію. Один і той же

самий об'єкт може виконувати різні дії у відповідь на одну і ту саму подію в залежності від стану об'єкту. Виконання дії, як правило, викликає зміну стану. Тобто, у цілому моделювання станів ґрунтується на поняттях «подія» і «стан».

Модель станів може включати декілька діаграм станів по одному на кожен клас, поведінка якого в часі важлива для проєктованого додатку. Клас, що має декілька станів, має важливу поведінку в часі. Якщо клас має тільки один стан, то його поведінку в часі можна ігнорувати.

Діаграми станів UML більш наочні й виразні в порівнянні з класичними уявленнями автоматів, але їх застосування вимагає більшої підготовленості користувача і пред'являє більш високі вимоги до «кмітливості» і «уважності» інструментів моделювання.

Діаграма кінцевого автомата задає поведінку системи як цілісної, єдиної сутності, вона моделює життєвий цикл єдиного об'єкту. У силу цього автоматний підхід зручно застосовувати для формалізації динаміки окремого, важкого для розуміння блоку системи.

Діаграма станів має схожу семантику з діаграмою діяльності, тільки діяльність тут замінена станом, переходи символізують дії. Таким чином, якщо для діаграми діяльності відмінність між поняттями «Діяльність» і «Дія» полягає в можливості подальшої декомпозиції, то на діаграмі станів діяльність символізує стан, в якому об'єкт знаходиться тривалий час, у той час як подія моментальна.

Зазвичай діаграми кінцевих автоматів застосовуються для моделювання поведінки подієво-керованих об'єктів.

Застосовувати діаграму станів необхідно тільки для тих класів, які проявляють цікаву поведінку, коли побудова діаграми станів допомагає зрозуміти, як все відбувається. Автомати станів можна також використовувати при моделюванні поведінки графічного інтерфейсу, як реакції на дії користувача.

У загальному випадку кінцевий автомат представляє динамічні аспекти модельованої системи у вигляді орієнтованого графа, вершини якого відповідають станам, а дуги – переходам. При цьому поведінка моделюється як

послідовне переміщення по графу станів від вершини до вершини по дугах, що зв'язують їх, з урахуванням їх орієнтації.

На діаграмі станів час знаходження системи в тому або іншому стані явно не враховується, проте передбачається, що послідовність зміни станів впорядкована в часі. Іншими словами, кожен подальший стан може настати пізніше передуючого йому стану.

Стан і його графічне зображення. Стан – умова або ситуація в ході життєвого циклу об'єкту, впродовж якої він задовольняє логічній умові, виконує певну діяльність або чекає на подію. У формулюванні задач стани часто відповідають дієсловам або дієприслівникам (чекає, додзвонюється) або виконанню деякої умови (включений, нижче запланованої ціни).

Об'єкти класу мають кінцеве число можливих станів. У конкретний момент часу кожен об'єкт може знаходитися рівно в одному стані. Впродовж часу свого існування об'єкти можуть проходити через один або декілька станів.

Стан на діаграмі зображується округленим прямокутником (рис. 2.69), який може бути розділений на дві секції горизонтальною лінією. Якщо вказана лише одна секція, то в ній записується тільки ім'я стану (рис. 2.69, *a*). Інакше в першій з них записується ім'я стану, а у другій – список деяких внутрішніх дій або переходів в цьому стані (рис. 2.69, *б*). При цьому під дією в мові UML розуміють деяку атомарну операцію, виконання якої призводить до зміни стану або повернення деякого значення (наприклад, «істина» або «хиба»).

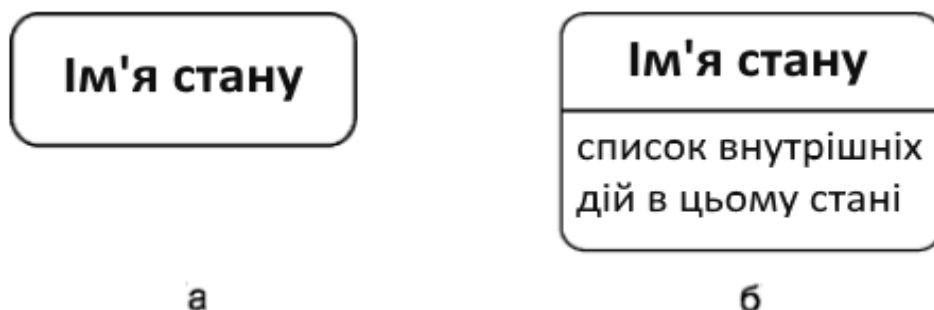


Рисунок 2.69 – Графічне зображення станів на діаграмі кінцевого автомата

Дія. Дія призводить до зміни стану системи, і може бути реалізоване за допомогою передачі повідомлення об'єкту, модифікацією зв'язку або значення

атрибуту. Кожна дія записується у вигляді окремого рядка і має наступний формат: **мітка дії / вираз дії**.

Якщо список виразів дії порожній, то мітка дії вказується без роздільника «/». Перелік міток дій в мові UML фіксований, причому ці мітки не можуть бути використані в якості імен подій:

1. **Вхідна дія (entry action)** – дія, яка виконується в момент переходу в цей стан.
2. **Дія виходу (exit action)** – дія, яка вироблена при виході з цього стану.
3. **Внутрішня діяльність (do activity)** – виконання об'єктом операцій або процедур, які вимагають певного часу.

Як приклад стану розглянемо аутентифікацію клієнта для доступу до ресурсів інформаційної системи (рис. 2.70, б).

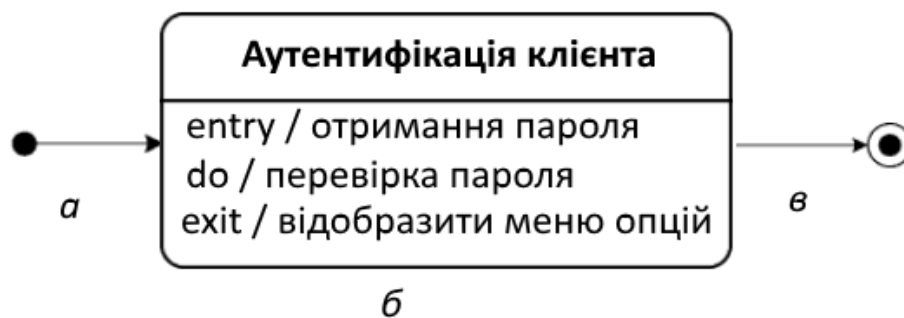


Рисунок 2.70 – Приклад стану з непорожньою секцією внутрішніх дій

Окрім звичайних станів на діаграмі станів можуть розміщуватися псевдостани.

Псевдостан – це вершина в кінцевому автоматі, яка має форму стану, але не має поведінки. Прикладами псевдостанів, які визначені в мові UML, є початковий і кінцевий стани.

Початковий стан – різновид псевдостану, який позначає початок виконання процесу зміни станів кінцевого автомата або знаходження модельованого об'єкту у складеному стані (див. рис. 2.70, а).

Кінцевий стан – різновид псевдостану, який позначає припинення процесу зміни станів кінцевого автомата або знаходження модельованого об'єкту у складеному стані (див. рис. 2.70, в).

Подія і перехід. Перебування модельованого об'єкту або системи в першому стані може супроводжуватися виконанням деяких внутрішніх дій або діяльності. При цьому зміна поточного стану об'єкту буде можлива або після завершення цих дій (діяльності), або при виникненні деяких зовнішніх подій. У обох випадках говорять, що відбувається перехід об'єкту з одного стану в інший.

Подія (event) – це подія, що трапилася в певний момент часу. Іншими словами, події – це зовнішні дії. Наприклад, натиснення кнопки або виліт рейсу. Події відбуваються в результаті виконання деякої дії в системі або її оточенні, тому в описі задачі вони часто відповідають дієсловом у минулому часі (живлення включене) або виконанню деякої умови (температура підвищилася).

Формально, подія є специфікацією факту, що має місце у просторі й в часі. Про події говорять, що вони «відбуваються», при цьому окремі події мають бути впорядковані в часі. Після настання події не можна вже повернутися до попередніх, якщо така можливість явно не передбачена в моделі.

Перехід – відношення між двома станами, яке вказує на те, що об'єкт у першому стані повинен виконати певні дії і перейти у другий стан.

В окремих випадках виникає необхідність явно показати ситуацію, коли перехід може мати декілька початкових станів або цільових станів. Такий перехід дістав назву – *паралельний перехід*.

Введення в розгляд паралельних переходів може бути обумовлене необхідністю синхронізувати та/або розділити окремі процеси управління на паралельні потоки (нитки) без специфікації додаткової синхронізації в паралельних кінцевих *підавтоматах* (submachine state).

Графічно такий перехід зображується вертикальною рисою, аналогічно позначенню переходу в мережах Петрі. Якщо паралельний перехід має дві або більше дуг, що виходять з нього, то його називають *розділенням*. Якщо ж він має дві або більше дуг, що входять, то його називають *злиттям*. Текстовий рядок специфікації паралельного переходу записується поряд з рисою і стосується всіх дуг, що входять або виходять.

У загальному випадку поведінка паралельних кінцевих підавтоматів відбувається незалежно один від одного, що дозволяє, наприклад, моделювати багатозадачність у програмних системах.

В окремих ситуаціях може виникнути необхідність врахувати в моделі синхронізацію настання окремих подій та спрацьовування відповідних переходів. Для цієї мети в мові UML є псевдостан, який називається *синхронізуючим станом* або *станом синхронізації*.

Стан синхронізації – це псевдостан у кінцевому автоматі, який використовується для синхронізації паралельних областей кінцевого автомата. Синхронізуючий стан позначається невеликим колом, усередині якого розміщений символ зірочки «*». Він використовується спільно з переходом-злиттям або переходом-розділенням для того, щоб явно вказати події в інших кінцевих підавтоматах, що роблять безпосередній вплив на поведінку цього підавтомату.

Окрім простих станів на діаграмі можуть відображатися **складені стани** (composite state), що складаються з вкладених в них підстанів (substate).

Складений стан може бути розбитий на зони, що називаються також **паралельними підавтоматами** (concurrent substates). Якщо на діаграмі є складений стан з вкладеними паралельними підавтоматами, то екземпляр сутності може одночасно знаходитися в декількох підстанах, але не більше ніж по одному з кожного підавтомата. У UML також визначені декілька псевдостанів, які перераховані нижче.

Початковий стан. З початкового стану автомата, початкового підстану складеного стану або паралельного підавтомату можуть виходити тільки переходи. У кінцевий стан автомата або кінцевий підстан складеного стану чи паралельного підавтомата можуть входити тільки переходи (рис. 2.71).

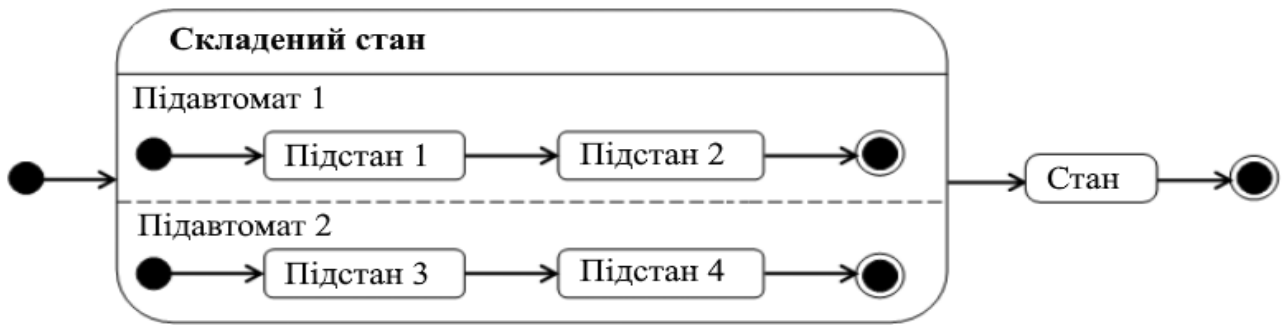


Рисунок 2.71 – Приклад стану з непорожньою секцією внутрішніх дій

Точки входу/виходу. Точок входу в автомат або складений стан (\circ) може бути декілька. Допускається кріплення до межі складеного стану. Точок виходу з автомату або складеного стану (\otimes) також може бути декілька. Допускається кріплення до межі складеного стану (рис. 2.72).

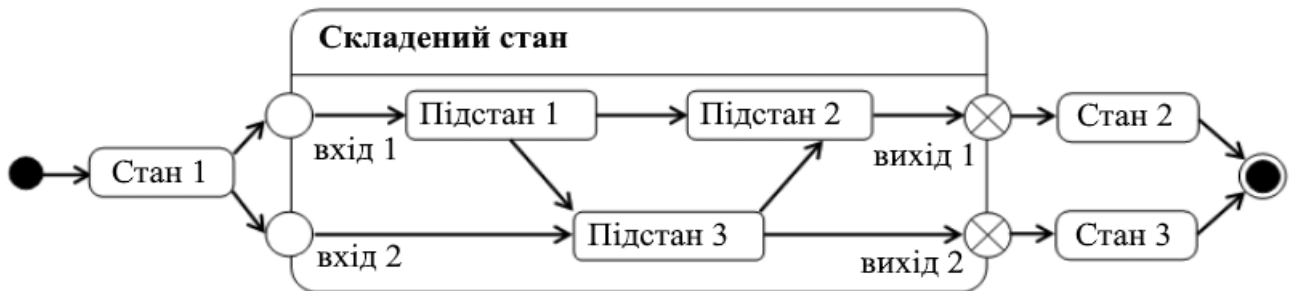


Рисунок 2.72 – Приклад стану з непорожньою секцією внутрішніх дій

Розгалуження і з'єднання переходів. Розгалуження переходів (\rightarrow) в паралельні підавтомати і з'єднання виходу з (\rightarrow) паралельних підавтоматів виконують функцію синхронізації підавтоматів складеного стану (рис. 2.73).

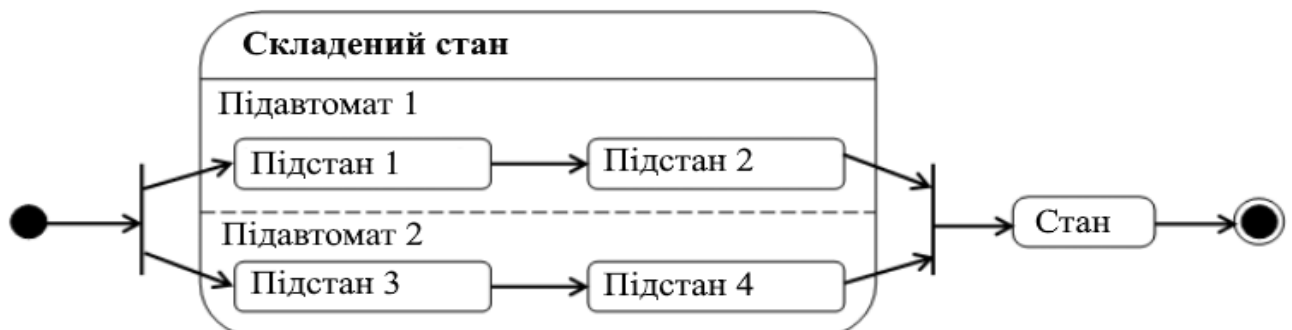


Рисунок 2.73 – Приклад стану з непорожньою секцією внутрішніх дій

При розробці діаграм автоматів слід дотримуватися наступних правил і рекомендацій.

1. Наявність у екземпляра сутності декількох станів, що відрізняються від простої схеми, наприклад, «*справний – несправний*» або «*активний – неактивний*», служить ознакою необхідності побудови діаграми автомата.
2. Діаграма автомата обов'язково повинна починатися знаком початкового стану і закінчуватися знаком кінцевого. Початковий стан в діаграмі автомата вказується тільки один раз, а кінцевих може бути декілька в цілях мінімізації перетинів переходів. Для підавтоматів рекомендується дотримуватися цього ж правила або використати точки входу / виходу.
3. Для полегшення сприйняття діаграми рекомендується використати декомпозицію з прихованням складених станів.
4. Діаграма не повинна містити ізольованих станів і переходів. Переходи і їх специфікації мають бути задані так, щоб на графі кожен стан був потенційно досяжний з початкового стану, і з будь-якого стану було потенційно досяжний кінцевий стан.

У кожен момент часу автомат або підавтомат повинні знаходитися тільки в одному стані. Це означає, що специфікація переходів з одного стану не повинна допускати потенційної можливості переходу в два і більше станів.

Повернемося до нашого прикладу Internet-магазину. Покупець оформляє замовлення. Клас **Замовлення**, окрім інших атрибутів має атрибут «статус». Простежимо динаміку руху замовлень в системі за допомогою діаграми кінцевого автомату, складеної для класу **Замовлення**.

За умовою завдання дані про зроблене замовлення поступають співробітникам відділу продаж, який перевіряє оплату, реквізити замовлення і передає його комірникові на комплектацію. Комірник, перевібивши наявність замовлених товарів і зібравши замовлення, якщо це можливо, робить відмітку про готовність.

Замовлення видається зі складу комірником. Комірник видає замовлення і відмічає в системі, що замовлення видане. Покажемо на діаграмі перехід замовлення між станами (рис. 2.74).



Рисунок 2.74 – Діаграма станів об'єкту класу Замовлення

При нормальному завершенні внутрішньої діяльності генерується відповідна подія. У даному прикладі при оформленні замовлення він має бути сплачений (вхідна дія **Сплатити замовлення**). Обробка замовлення передбачає перевірку оплати і наявності товарів (діяльність **Перевірити оплату і наявність**), перехід в один із станів (**На комплектації**, **Укомплектований**, **Виданий**) означає зміну статусу замовлення. Умова **[Покупець не забрав замовлення]** викликає перехід в стан **Розформований** при цьому виконується дія **Повернути гроші на картку** (рис. 2.75).



Рисунок 2.75 – Остаточна діаграма станів об'єкту Замовлення

Розглянемо ще одну діаграму станів з паралельними процесами. Так, наприклад, при включенні комп'ютера з деякою мережевою ОС паралельно починається виконання декількох процесів (рис. 2.76). Зокрема, на діаграмі кінцевого автомату після включення комп'ютера паралельно відбувається перевірка пароля користувача і запуск різних служб.

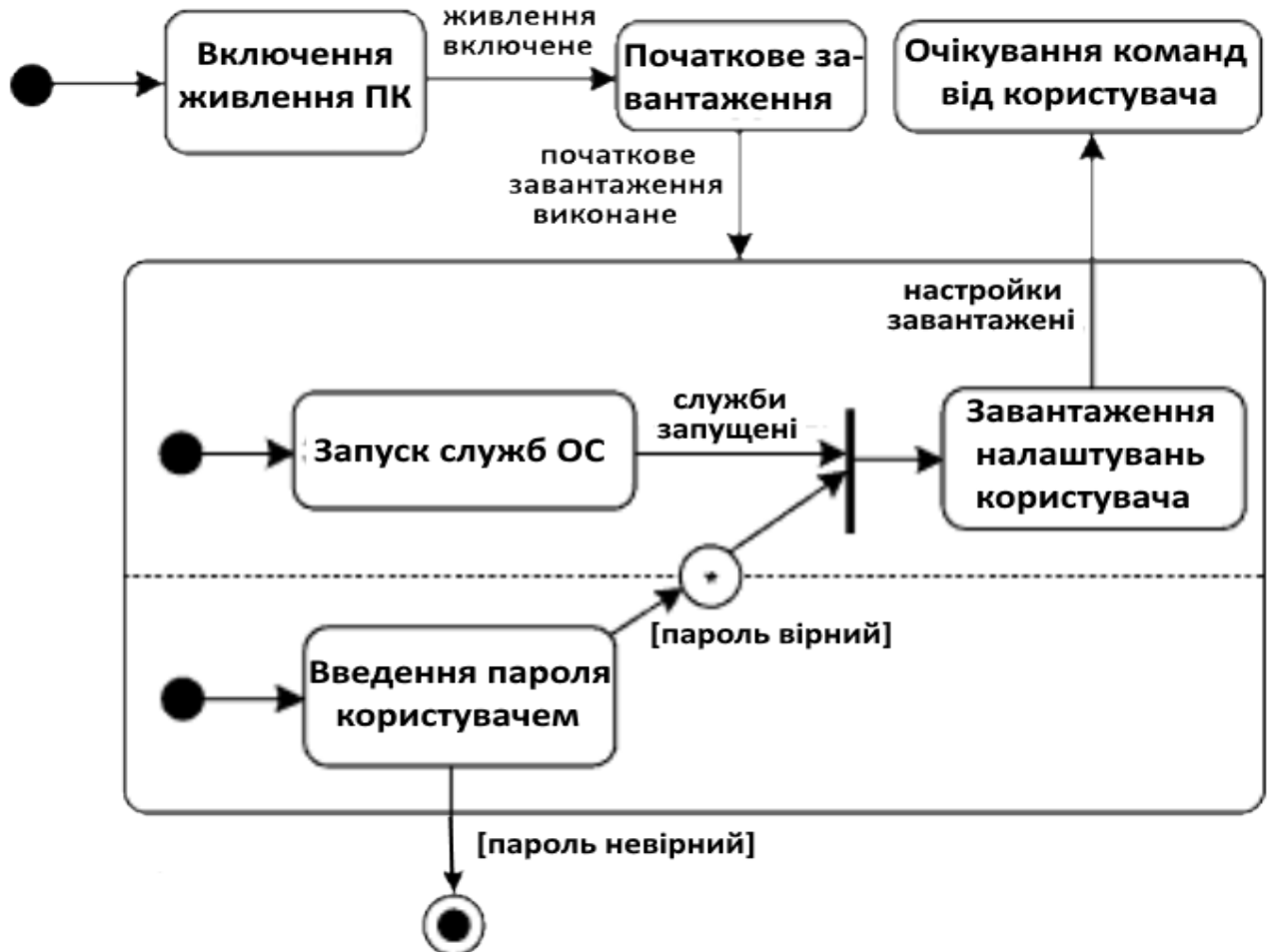


Рисунок 2.76 – Діаграма кінцевого автомату включення комп'ютера

При цьому робота користувача на комп'ютері стане можливою тільки в разі успішної його аутентифікації, інакше комп'ютер може бути вимкнений. Розглянуті особливості синхронізації цих паралельних процесів враховані на відповідній діаграмі станів за допомогою синхронізуючого стану.

Іноді розробку діаграми станів, особливо в умовах дефіциту часу, опускають, оскільки часто відбувається дублювання інформації, представленої на діаграмах послідовності та комунікації (кооперації).

2.6 Проектування і розробка бази даних

При проектуванні інформаційних систем з базами даних, що входять до їх складу, зручно користуватися класифікацією моделей, зображеною на рис. 2.77. Усі моделі даних, які використовуються на трьох етапах проектування, діляться на три види. На першому етапі досліджується предметна область, виявляються в ній об'єкти і процеси, які треба буде відобразити в ІС при розв'язанні задач, для яких розробляється ІС. Модель, яка використовується на цьому етапі, служить для наочного представлення семантичних зв'язків у предметній області. Строга формалізація структури даних на цьому етапі не обов'язкова. Такі моделі називаються *інфологічними (концептуальними)*. Нині найпоширенішою інфологічною моделлю є модель «сутність-зв'язок».

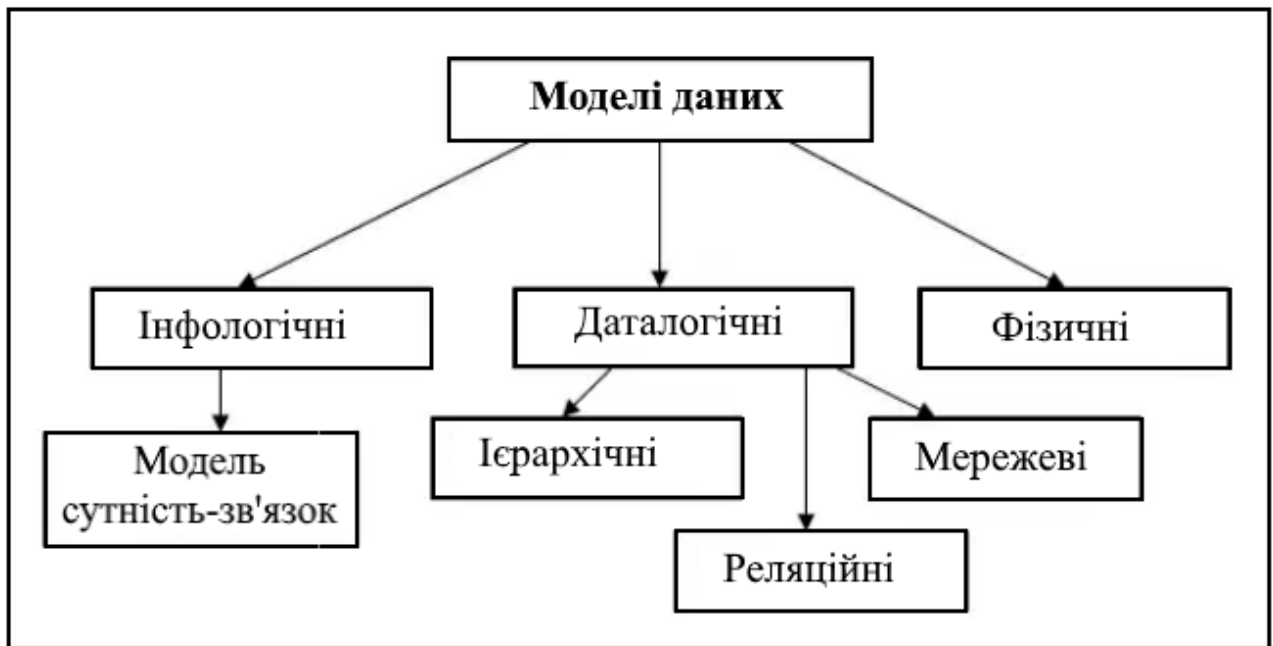


Рисунок 2.77 – Моделі даних

Після того як закінчено дослідження предметної області і детально поставлена задача проектування можна переходити до другого етапу, на якому проектується *база даних*. На цьому етапі використовуються формальні моделі даних, в які треба перетворити інфологічну модель. Такі моделі, які безпосередньо використовуються в базах даних, називаються *даталогічними*.

Найбільше поширення серед даталогічних моделей отримали реляційні моделі баз даних.

Базу даних незалежно від її даталогічної моделі можна по-різному розмістити на різних зовнішніх носіях. Наприклад, можна використати жорсткий диск або твердотільну зовнішню пам'ять. Для опису фізичного розміщення бази даних служить *фізична модель*.

2.6.1 Концептуальна модель предметної області

При проектуванні ІС дані треба представити у вигляді простої моделі, що відображає сенс даних, їх взаємозв'язок і не прив'язується при цьому до конкретного типу бази даних. Такі моделі дістали назву *інфологічних (концептуальних)*.

Концептуальна модель предметної області – це знання про предметну область у вигляді понять (концептів). Знання можуть бути як у вигляді неформальних знань, так і виражені формально за допомогою деяких засобів. Такими засобами можуть виступати текстові описи предметної області, набори посадових інструкцій тощо.

Досвід показує, що текстовий спосіб представлення моделі предметної області є неефективним. Набагато інформативнішим і кориснішим при розробці баз даних є опис предметної області, виконаний за допомогою спеціалізованих графічних нотацій. Є велика кількість методик опису предметної області. Концептуальна (інфологічна) модель БД – відображає інформаційний зміст даних як основних понять і відношення між ними. Концептуальна модель не зачіпає фізичного стану даних, у тому числі архітектури даних, методів доступу, форматів фізичних даних.

Концептуальна модель предметної області описує процеси, що відбуваються у предметній області й дані, які використовуються цими процесами. Від того, наскільки правильно змодельована предметна область, залежить успіх подальшої розробки програм.

Побудова моделі предметної області розпочинається з виявлення абстракцій, що існують у реальному світі та належать модельованій предметній області. Концептуальна модель відображає семантику предметної області у вигляді сукупності понять (сутностей), їх характеристик (атрибутів) і зв'язків (асоціативних відношень між сутностями). Варто відзначити, що концептуальна модель створюється в першу чергу для полегшення сприйняття інформації звичайним користувачем.

У 1976 році Пітер Чен запропонував термін *сутність*, а модель, що складається з пов'язаних між собою сутностей, дістала назву *модель «сутність-зв'язок» (entity-relationship, ER- модель)*.

Нотація – відображення структури системи, елементів даних за допомогою спеціальних графічних символів (нотації П. Чена, Р. Баркера, Гордона Еверста, Дж. Мартіна).

ER-модель використовується при високорівневому (концептуальному) проектуванні баз даних. За її допомогою можна виділити ключові сутності та позначити зв'язки, які можуть встановлюватися між ними. Як інструмент концептуального проектування баз даних використовуються різні варіанти (нотації) діаграм сутність-зв'язок.

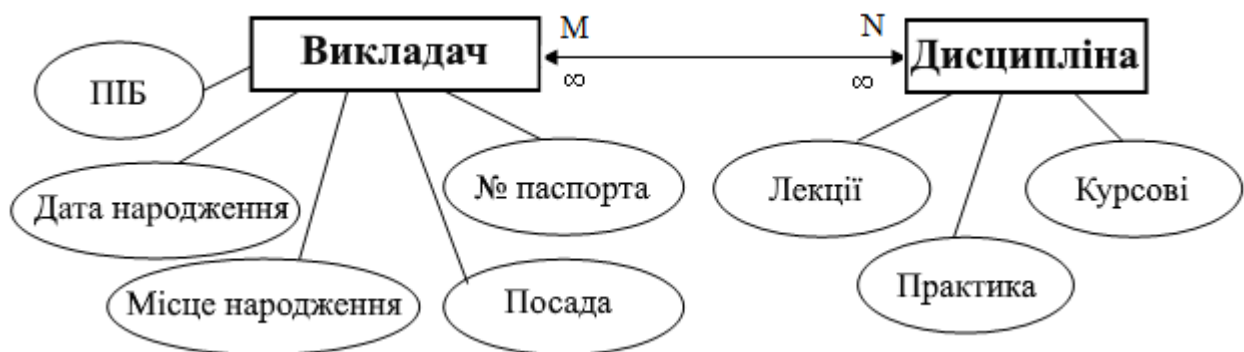
При розробці ER-моделей необхідно обстежити предметну область (організацію, підприємство) і виявити:

1. Сутності, про які зберігаються дані в організації (підприємстві), наприклад, люди, місця, ідеї, події тощо (будуть представлені у вигляді блоків).
2. Зв'язки між цими сутностями (будуть представлені у вигляді ліній, що з'єднують ці блоки).
3. Властивості цих сутностей (будуть представлені у вигляді імен атрибутів в цих блоках).

На рис. 2.78 наведений простий (вироджений) приклад інфологічної моделі в нотації Чена.

Рисунок 2.78 – Інфологічна модель *Викладач*

Оскільки викладач може вести декілька дисциплін, то потрібно детальніше описати дисципліни, щоб мати можливість, знаючи дисципліну, знайти викладача, що веде її, то модель можна ускладнити, виділивши дисципліну окремо і зв'язавши її з викладачем (рис. 2.79).

Рисунок 2.79 – Інфологічна модель *Викладач-Дисципліна*

Один викладач може вести багато дисциплін і одну дисципліну можуть вести багато викладачів. Такий зв'язок називається **багато до багатьох** і позначається як $M:N$, а на рисунках часто як $\infty \longleftrightarrow \infty$.

На рис. 2.80 зображена спрощена інфологічна модель для побудови розкладу занять. У реальній задачі необхідно врахувати багато додаткових чинників. Є різні види занять: лекції, практичні, лабораторні тощо. Лекції проводяться з потоком, а лабораторні – з підгрупами. Розклад залежить від номера тижня. На рис. 2.77 відразу 6 елементів пов'язані один з одним. Такий зв'язок називається **зв'язком степені K** ($K=6$). Факультет з кафедрою, кафедра з групою і група із студентом мають зв'язки **один до багатьох**.

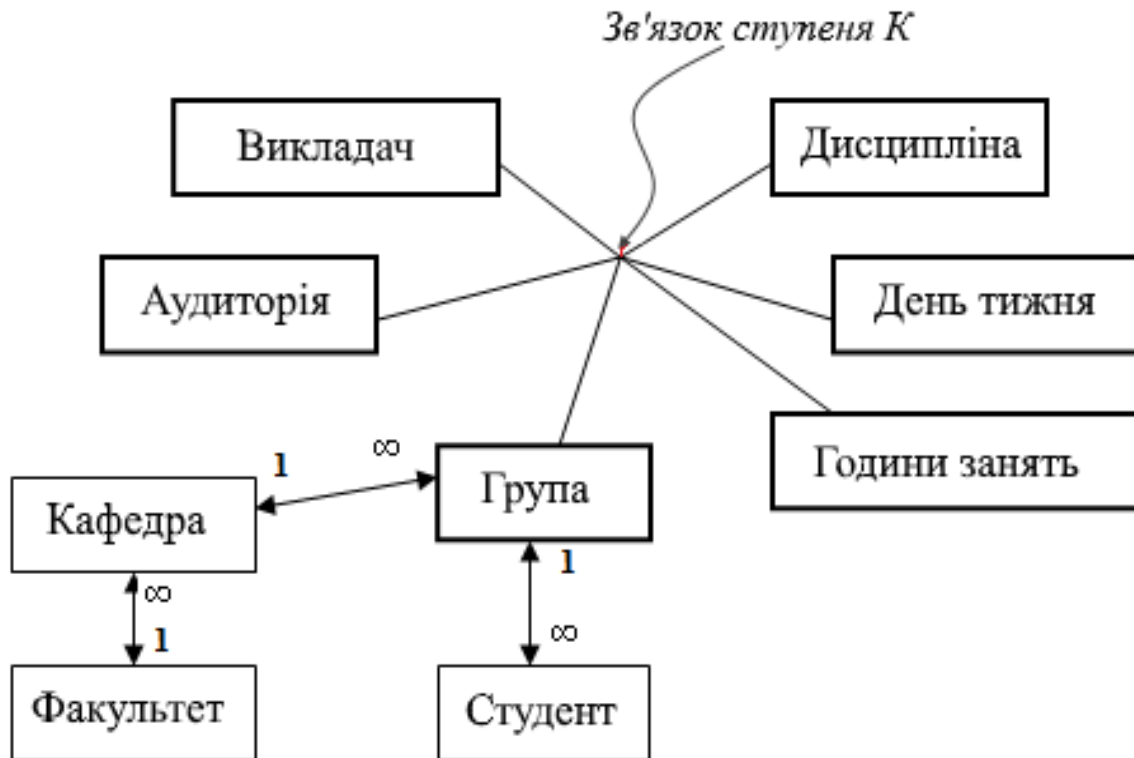


Рисунок 2.80 – Інфологічна модель побудови розкладу занять

Розглянемо детальніше, якими бувають самі сутності, їх властивості та зв'язки. У літературі часто замість терміну *властивість сутності* використовують *атрибут сутності*. Але термін *атрибут* відношення (колонка, стовпчик таблиці) застосовують у реляційній моделі. При одночасному застосуванні обох моделей може виникнути плутанина.

Види сутностей бувають *сильними* і *слабкими*, діляться на *тип* і *підтип*. Слабка сутність у контексті конкретної задачі не може існувати без сильної. Наприклад, у ЗВО студент обов'язково входить до групи, тобто сутність *Студент* не може існувати без сутності *Група*, а сутність *Співробітник* – без сутності *Відділ*.

Якщо частина екземплярів сутності має додаткові властивості відносно до інших екземплярів, то цю частину виділяють у *підтип*. На рис. 2.81 сутність *Студент* є підтипом типу *Житель міста*, а сутність *Програміст* – підтипом типу *Співробітник*. Між типом і підтипом зв'язок 1:1, але він неповний, оскільки не кожному екземплярі типу відповідає екземпляр підтипу.

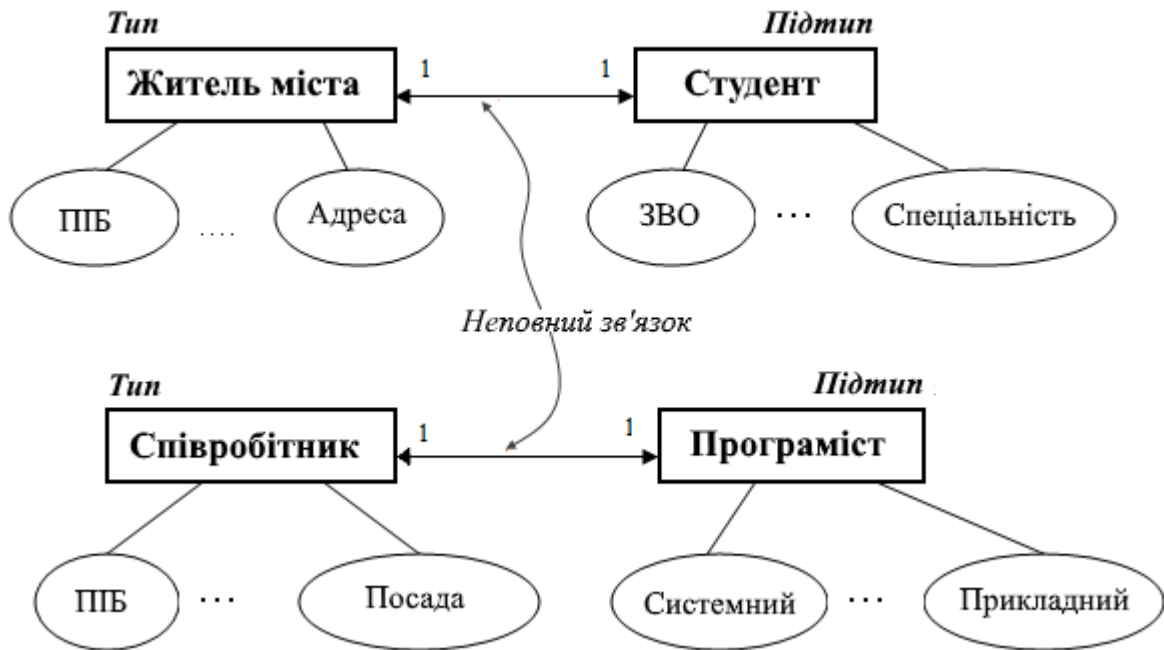


Рисунок 2.81 – Тип і підтип сутностей

Зв'язки між сутностями можуть бути *степені K*, *повні і неповні*, а також типу $M:N$ (бінарні зв'язки, $K=2$):

- 1:1;
- 1:N (на схемах часто використовують позначення $1 \longleftrightarrow \infty$);
- M:N, $M>1$, $N>1$ ($\infty \longleftrightarrow \infty$).

Степінь K – кількість пов'язаних між собою сутностей. Приклад зв'язку міри $K=6$ показаний на рис. 2.80. Для побудови розкладу необхідно, щоб були пов'язані рівно по одному екземпляру кожної сутності.

Повним зв'язком між двома сутностями називається такий зв'язок, при якому кожному екземпляру однієї сутності відповідає хоч би один екземпляр іншої сутності. Наприклад, кожному студентові відповідає одна група і кожній групі відповідає багато студентів.

При **неповному (частковою) зв'язку** деякі екземпляри однієї сутності не пов'язані ні з одним екземпляром іншої сутності. Наприклад, не всі студенти живуть у гуртожитку.

Зв'язок типу один-до-одного (1:1, 1–1 або \longleftrightarrow) означає, що один екземпляр першої сутності пов'язаний з одним екземпляром другої сутності. На практиці

такий вид зв'язку зустрічається нечасто, оскільки майже завжди два зв'язані об'єкти можна об'єднати в один. Як приклад використання такого зв'язку може послужити зв'язок між персональною і службовою характеристикою працівника деякої фірми. Оскільки доступ до службової інформації зазвичай обмежений, то захист від несанкціонованого доступу набагато легше реалізувати при роздільному зберіганні таких даних. Зв'язок 1:1 ще створюють для кращого розуміння моделі або для економії дискового простору. Наприклад, директора і завод краще розглядати як різні сутності, номер телефону краще зберігати в окремій сутності, якщо він є, наприклад, тільки у 20-40% працівників фірми.

Зв'язок типу один-до-багатьох ($1:N$, $1 \longleftrightarrow \infty$ або \longleftrightarrow) означає, що один екземпляр першої сутності пов'язаний з декількома екземплярами другої сутності. Це найчастіше використовуваний тип зв'язку. Сутність з боку «один» називається *батьківською*, а сутність з боку «багато» – *дочірньою*. Зв'язок типу $1:N$ – найпростіший і зручніший для побудови реляційної бази даних. На рис. 2.82 наведений приклад моделі сутність-зв'язок із чотирьох сутностей зі зв'язками $1:N$.



Рисунок 2.82 – Зв'язки типу $1:N$

Зв'язок типу багато-до-багатьох ($M:N$, $\infty \longleftrightarrow \infty$ або \longleftrightarrow) означає, що кожен екземпляр першої сутності може бути пов'язаний з декількома екземплярами другої сутності, і кожен екземпляр другої сутності може бути пов'язаний з декількома екземплярами першої сутності. Зв'язок типу $M:N$ вимагає при переході до реляційної моделі будувати додаткове відношення (таблицю зв'язків). Приклад зв'язку типу $M:N$ приведений на рисунку 2.80.

При перетворенні моделі сутність-зв'язок у реляційну модель кожній сутності з моделі сутність-зв'язок у реляційній моделі ставиться у відповідність відношення (таблиця), кожній властивості сутності – атрибут відношення. У відношення, як правило, додається атрибут-лічильник, який служить

формальним *первинним ключем* (ідентифікатором кортежу). У базі даних кортеж – це рядок або запис.

Щоб задати зв'язок *один до багатьох*, у відношенні з боку *багато* створюється додатковий атрибут *зовнішній ключ* (рис. 2.83). Зовнішній ключ набуває значення тільки з множини значень первинного ключа відношення з боку *один*.

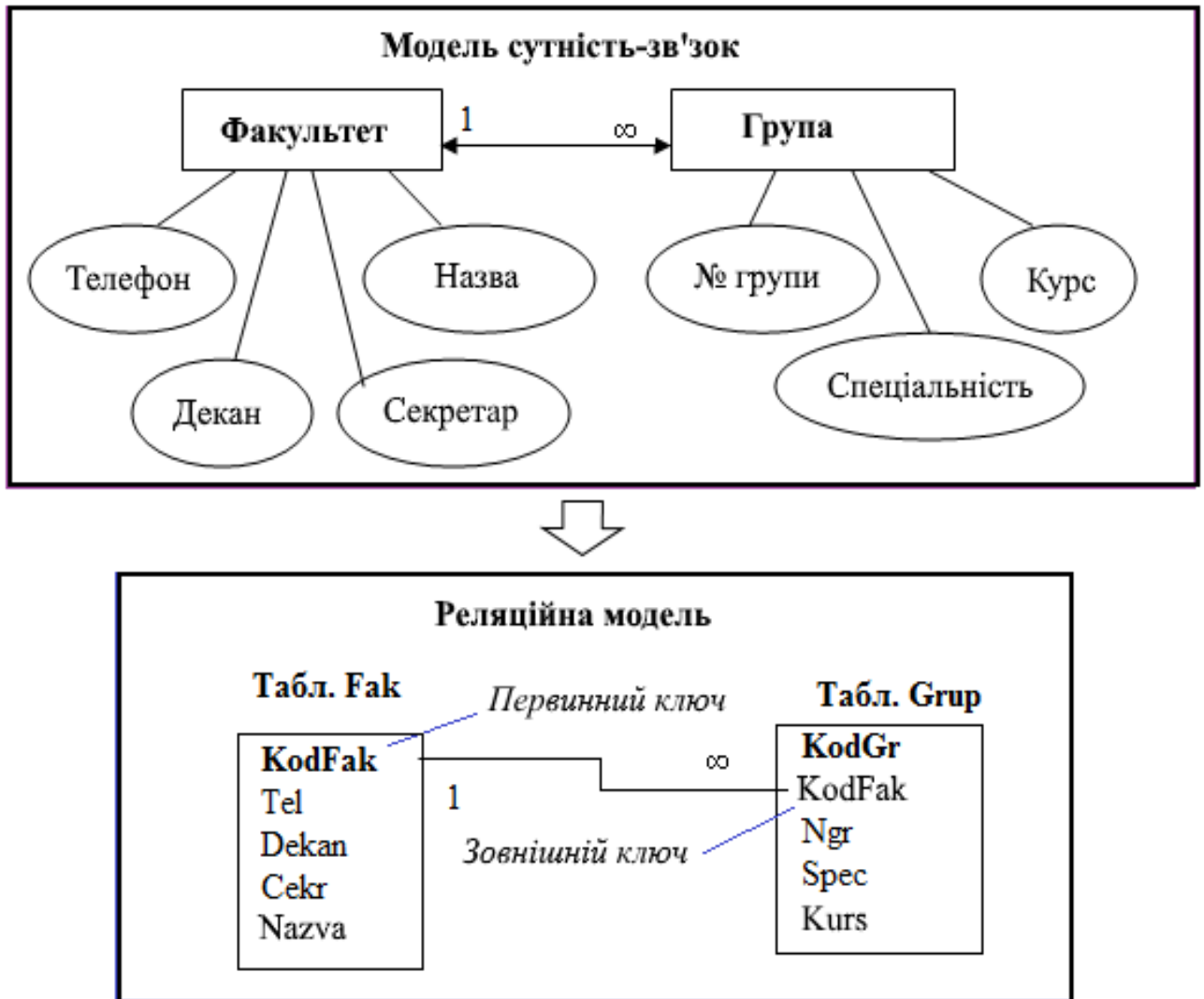


Рисунок 2.83 – Перетворення моделі сутність-зв'язок (1:N) у реляційну модель

Щоб задати зв'язок *багато до багатьох*, необхідно створити додаткову таблицю зв'язків, атрибутами якої служать зовнішні ключі, що відповідають первинним ключам зв'язаних таблиць (рис. 2.84).

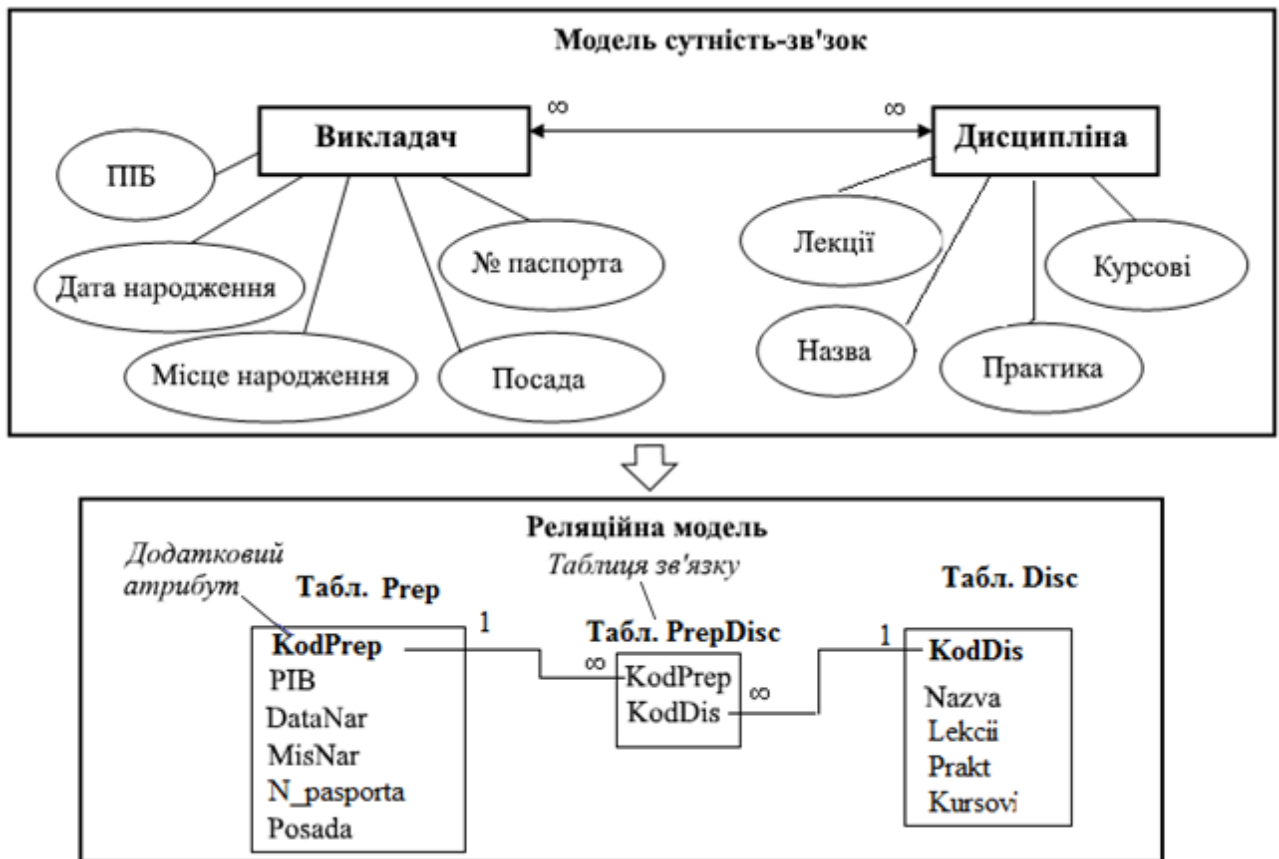


Рисунок 2.84 – Перетворення моделі сутність-зв'язок (M:N) у реляційну модель

На рис. 2.85 показані заповнені таблиці *Viklad*, *Disc* і *VikladDisc*. Для того, щоб вказати, що викладач *Андреев* веде дисципліну *Асемблер*, у таблиці *VikladDisc* створений рядок зі значеннями $KodV=1$ і $kodD=3$.

KodV	PIB
1	Андреев Б. С
2	Борисова М. П.
.....
18	Яценко М.Т.

KodV	KodD
1	20
1	3
2	1
2	3
2	20

KodD	Nazva
1	Мова С++
2	Бази даних
3	Асемблер
.....
20	Ява

Рисунок 2.85 – Приклад заповнення таблиці зв'язків у реляційній моделі

Для зв'язку *степені K* у реляційній моделі будується таблиця зв'язку з *K* зовнішніми ключами. На рис. 2.86 побудовані модель сутність-зв'язок і реляційна модель для зв'язку степені $K=5$.

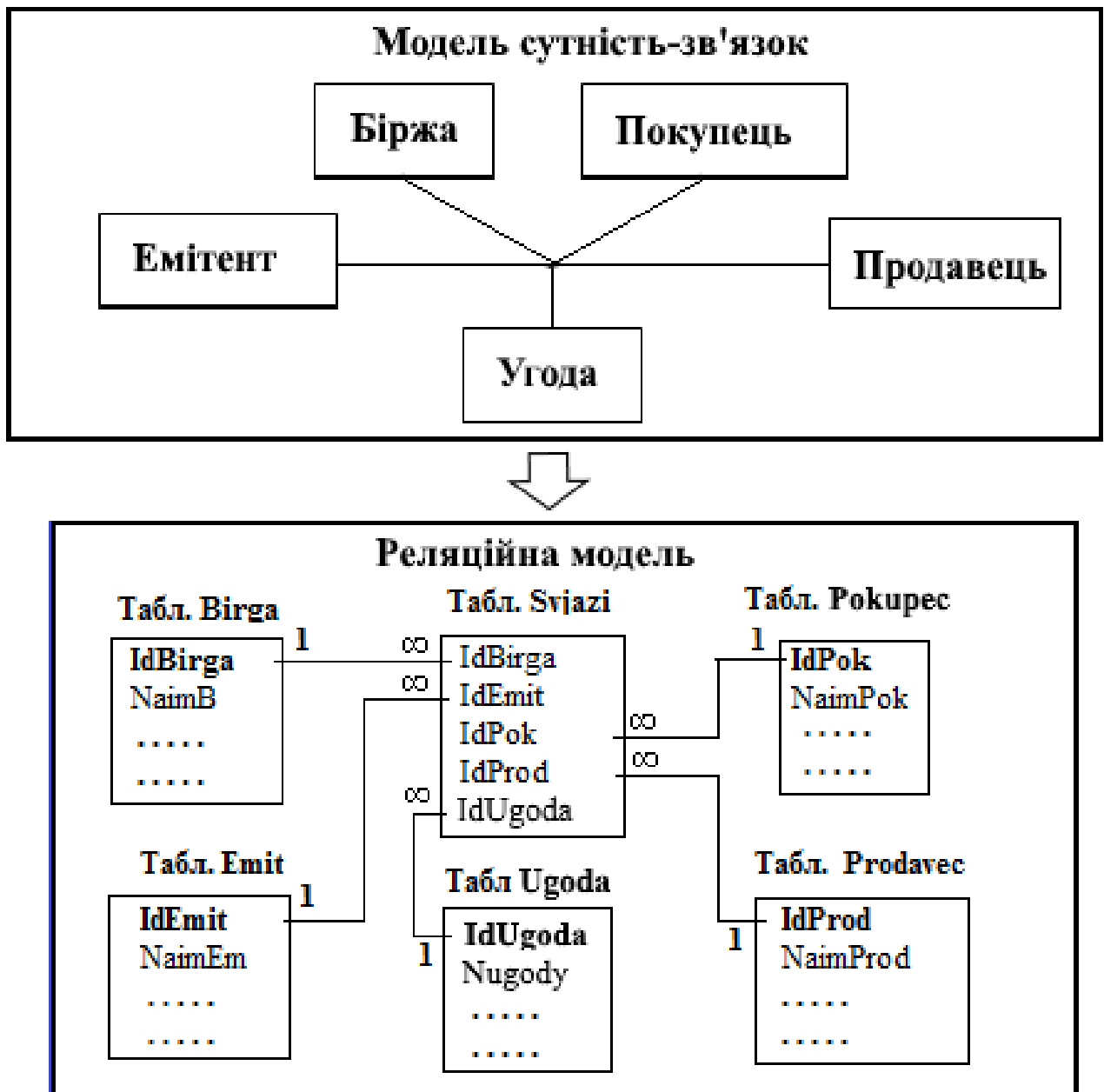


Рисунок 2.86 – Перетворення моделі сутність-зв'язок зі зв'язком *степені K* у реляційну модель

Концептуальне моделювання предметної області також зручно виконувати за допомогою **Діаграми класів** рівня нотацій аналізу мови UML. Розглянемо простий приклад побудови ER-моделі на UML з використанням діаграм класів (рис. 2.87). Операції (методи) у класах на етапі побудови концептуальної моделі відсутні.

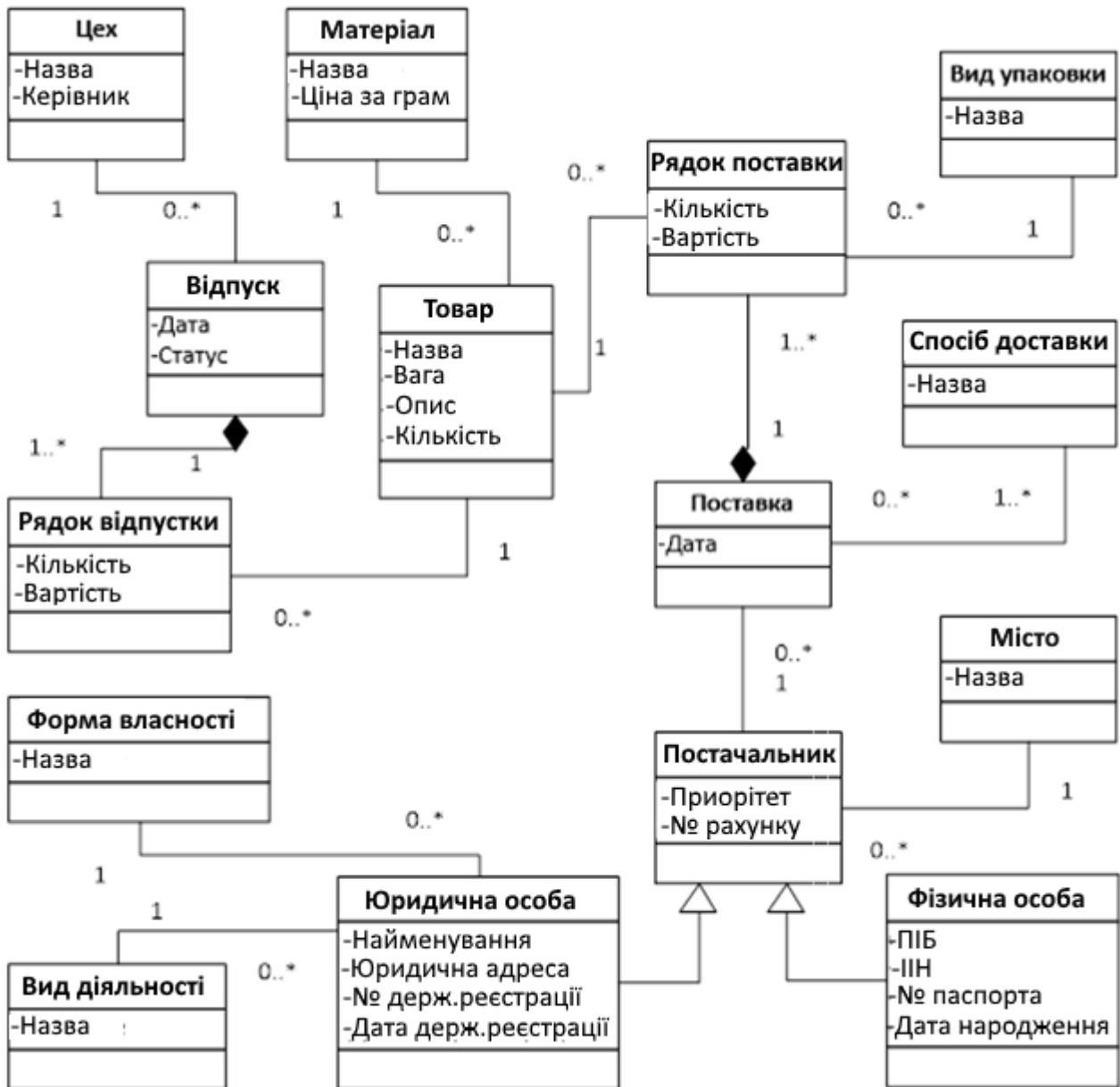


Рисунок 2.87 – Приклад концептуальної моделі (діаграма класів)

2.6.2 Логічна модель бази даних

На наступному, нижчому рівні, знаходиться логічна модель даних предметної області. Логічна модель описує поняття предметної області, їх взаємозв'язок, а також обмеження на дані, що накладаються предметною областю. Приклади понять – «співробітник», «відділ», «проект». Приклади взаємозв'язків між поняттями – «співробітник числиться рівно в одному відділі», «співробітник може виконувати декілька проектів», «над одним проектом може

працювати декілька співробітників». Приклади обмежень – «вік співробітника не менше 18 і не більше 60 років».

Можна виділити три основні види логічних моделей:

- ієрархічна модель;
- мережева модель;
- реляційна модель.

Логічна модель даних є початковим прототипом майбутньої бази даних. Логічна модель будується в термінах інформаційних одиниць, реалізація якої можлива в обраній СКБД, а найчастіше – без прив'язки до конкретної СКБД. Попереднім засобом розробки логічної моделі даних є різні варіанти інфологічних (інформаційно-логічних) моделей – ER-діаграма. Одну і ту ж ER-модель можна перетворити як у реляційну модель даних, так і в модель даних для ієрархічних і мережевих СКБД.

Для логічної моделі даних характерне те, що виконуючи всі основні вимоги, які пред'являються до СКБД, не підтримується орієнтація на конкретну СКБД, що реалізується у фізичній моделі даних.

Логічна модель – це графічне представлення структури БД з урахуванням моделі даних (ієрархічною, мережевою, реляційною тощо) і незалежне від кінцевої реалізації бази даних і апаратної платформи. Іншими словами, вона показує, **що** зберігається в БД (об'єкти предметної області, їх атрибути і зв'язки між ними), але не відповідає на питання **«як?»**.

Методологія IDEF1X – один з підходів до моделювання даних, заснована на концепції «сутність-зв'язок», яка орієнтована на проектування реляційних моделей баз даних.

В основі реляційної моделі лежить поняття нормалізованого відношення (таблиці). При цьому сутності предметної області відображаються в таблиці бази даних, кожній властивості сутності – атрибут відношення (рис. 2.88), що мають такі властивості:

- немає однакових кортежів (рядків), вони розрізняються за унікальним ідентифікатором – первинним ключем;

- кортежі (рядки / записи) не впорядковані зверху вниз;
- атрибути (стовпчики) не впорядковані зліва направо; в операціях з таблицею її рядки і стовпчики можуть бути видимими в будь-якій послідовності безвідносно їх змісту і сенсу;
- усі значення атрибутів – скаляри і мають однакову природу (побудовані на одному домені).



Рисунок 2.88 – Таблиця реляційної бази даних

Ключ – стовпчик або група стовпчиків, значення яких однозначно ідентифікують кожен рядок (кортеж, запис).

В одній таблиці може бути декілька ключів: один **первинний**, за допомогою якого здійснюється зв'язування відношень, а інші – **альтернативні**.

Властивості ключа:

- унікальність (не може бути рядків з однаковим ключем);
- ненадмірність (влучення будь-якого атрибуту з ключа позбавляє його властивості унікальності).

Реляційна база даних – це безліч пов'язаних між собою відношень. Зв'язки задаються за допомогою вторинних ключів (Foreign key – FK), тобто атрибутів, які в інших відношеннях є первинними ключами (Primary key – PK).

Основні обмеження цілісності реляційної моделі:

1. Атрибути з первинного ключа не можуть набувати невизначеного значення (цілісність об'єктів).

2. Вторинні ключі не можуть набувати значень, яких немає серед значень первинних ключів пов'язаної таблиці. Якщо відношення R2 має серед своїх атрибутів якийсь зовнішній ключ (FK), який відповідає первинному ключу (PK) відношення R1, то кожне значення FK має дорівнювати одному зі значень PK.

У табл. 2.1 наведені основні елементи нотації IDEF1X для проектування реляційних моделей баз даних.

Таблиця 2.1. Основні елементи нотації IDEF1X

Сутність	Графічне зображення		
Незалежна сутність	<p>Найменування</p> <table border="1"> <tr> <td>Унікальний ідентифікатор (PK)</td> </tr> <tr> <td>Атрибути</td> </tr> </table>	Унікальний ідентифікатор (PK)	Атрибути
Унікальний ідентифікатор (PK)			
Атрибути			
Залежна сутність	<p>Найменування</p> <table border="1"> <tr> <td>Посилання на ідентифікатор (FK)</td> </tr> <tr> <td>Атрибути</td> </tr> </table>	Посилання на ідентифікатор (FK)	Атрибути
Посилання на ідентифікатор (FK)			
Атрибути			
Зв'язок	Графічне зображення		
Неідентифікуючий зв'язок			
Ідентифікуючий зв'язок			
Зв'язок «Багато до багатьох»			
Спадкування (узагальнення)			
Повне			
Неповне			

Незалежна сутність – це сутність, унікальний ідентифікатор якої не наслідують з інших сутностей. Зображається у вигляді прямокутника з прямими краями.

Залежна сутність – це сутність, унікальний ідентифікатор якої включає щонайменше один зв'язок з іншою сутністю. Наприклад, рядок документу не може існувати без самого документу (залежить від нього). Залежна сутність зображається у вигляді прямокутника із заокругленими краями.

Приведемо правила відображення (перетворення) концептуальної моделі БД в реляційно нормалізовану модель бази даних, яка зазвичай відповідає третій нормальній формі.

1. Перетворення сутностей в реляційні відношення (таблиці) повинне задовольняти першій нормальній формі.

2. Перетворення асоціацій пов'язане з підтримкою посилення цілісності між таблицями. Асоціативні відношення 1:М, М:1 реалізуються за допомогою переміщення вторинного ключа в сутність з мірою зв'язку «багато». Відношення М:М вимагає введення перехресної таблиці, в яку в якості вторинних ключів включаються первинні ключі обох сутностей.

3. Перетворення агрегації. Створюється незалежна таблиця (частина) з атрибутом (зовнішнім ключем), що посиляється на первинний ключ таблиці власника (ціле).

4. Перетворення композиції. Створюється залежна таблиця (частина) з первинним ключем, що посиляється на первинний ключ таблиці власника (ціле).

5. Перетворення спадкоємства (узагальнення) може здійснюватися різними способами. Найприйнятнішим є відображення кожного підтипу в залежну таблицю з посиленням на первинний ключ таблиці-супертипу в якості первинного ключа.

На рис. 2.89 наведена логічна модель БД, яка була отримана перетворенням концептуальної моделі (див. рис. 2.87) в модель реляційної бази даних.

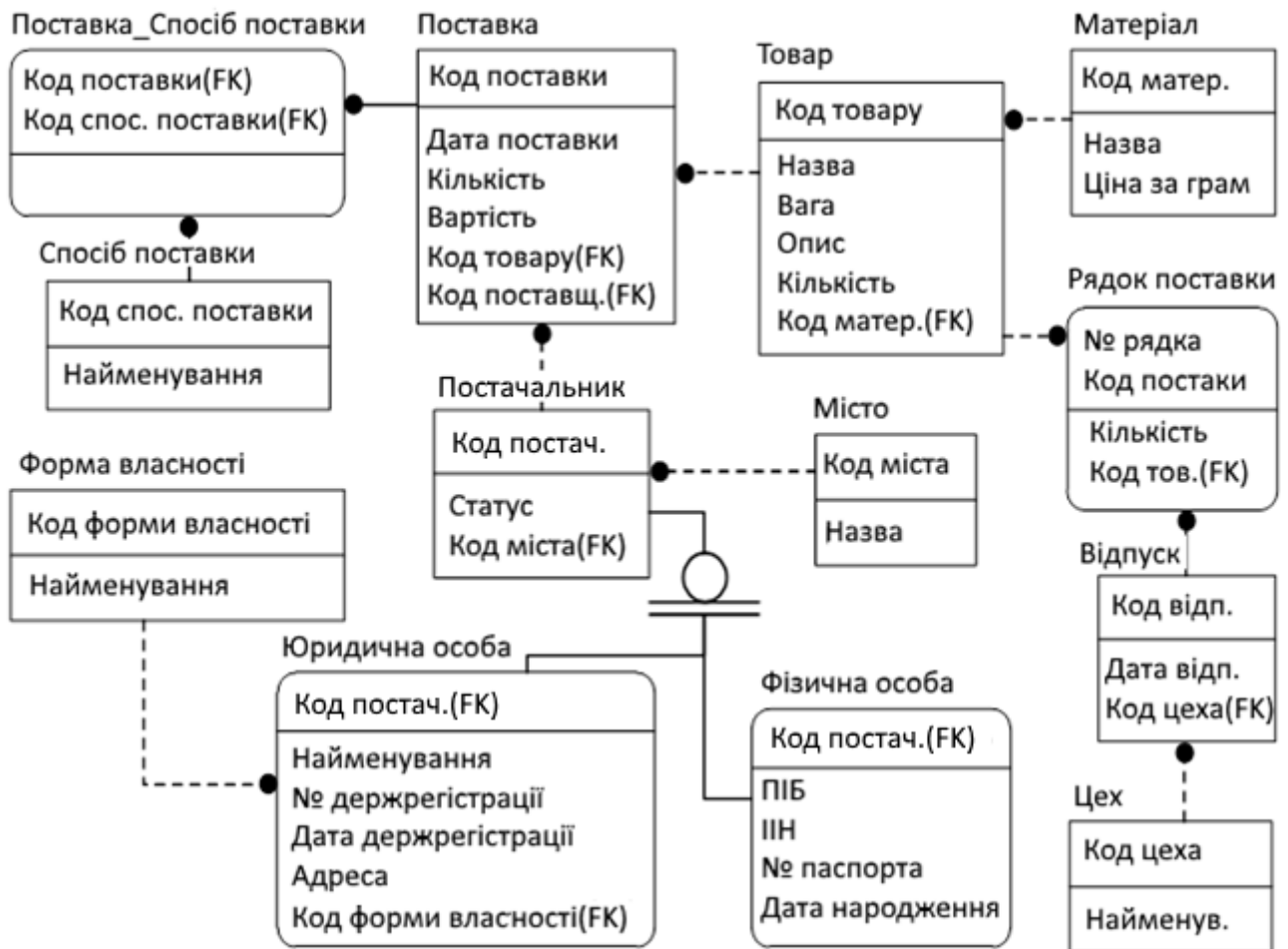


Рисунок 2.89 – Логічна модель бази даних

2.6.3 Фізична модель бази даних

На ще нижчому рівні знаходиться фізична модель даних.

Фізична модель – логічна модель бази даних, виражена в термінах мови опису даних конкретної СКБД. Обмеження, наявні в логічній моделі даних, реалізуються різними засобами СКБД, наприклад, за допомогою індексів, декларативних обмежень цілісності, тригерів, процедур, що зберігаються. При цьому рішення, прийняті на рівні логічного моделювання, визначають деякі обмеження, у межах яких можна розвивати фізичну модель даних. Наприклад, відношення, що містяться в логічній моделі даних, мають бути перетворені в таблиці, але для кожної таблиці можна додатково оголосити різні індекси, що підвищують швидкість звернення до даних.

Якщо фізична модель даних реалізована засобами реляційної СКБД, то відношення, розроблені на стадії формування логічної моделі даних, перетворюються в таблиці, атрибути стають стовпцями таблиць, для ключових атрибутів створюються унікальні індекси, домени перетворюються в типи даних, прийняті в конкретній СКБД.

Фізична модель бази даних містить усі деталі, необхідні конкретній СКБД для її створення: найменування таблиць і стовпців, типи даних, визначення первинних і зовнішніх ключів тощо (рис. 2.90).

Фізична модель, як структура бази даних для ЕОМ, будується на основі логічної моделі (структура для проектувальника) з урахуванням обмежень, що накладаються можливостями вибраної СКБД.

Імена. Імена сутностей і полів (таблиць і стовпців) мають бути записані латинськими буквами, без пропусків, починаючи з букви.

Типи даних. Для кожного атрибуту необхідно визначити тип даних і їх значень, які підтримуються вибраною СКБД.

Зв'язки. Реляційна модель даних вимагає перетворення зв'язків «багато до багатьох» та ієрархії успадкування та допускає їх тільки на рівні логічної моделі бази даних.

Наведені три варіанти фізичної моделі розрізняються способами реалізації наслідування:

- міграція первинного ключа і неключових атрибутів в ієрархії від нащадків до предка (рис. 2.90);
- міграція первинного ключа і неключових атрибутів в ієрархії від предка до нащадків (рис. 2.91);
- заміна ієрархії наслідування ідентифікуючими зв'язками (рис. 2.92).

Перетворення зв'язку «багато до багатьох» (таблиці «Постачання» і «Спосіб доставки») включає створення нової таблиці-розв'язки і двох нових зв'язків «один до багатьох» від старих таблиць до нової.

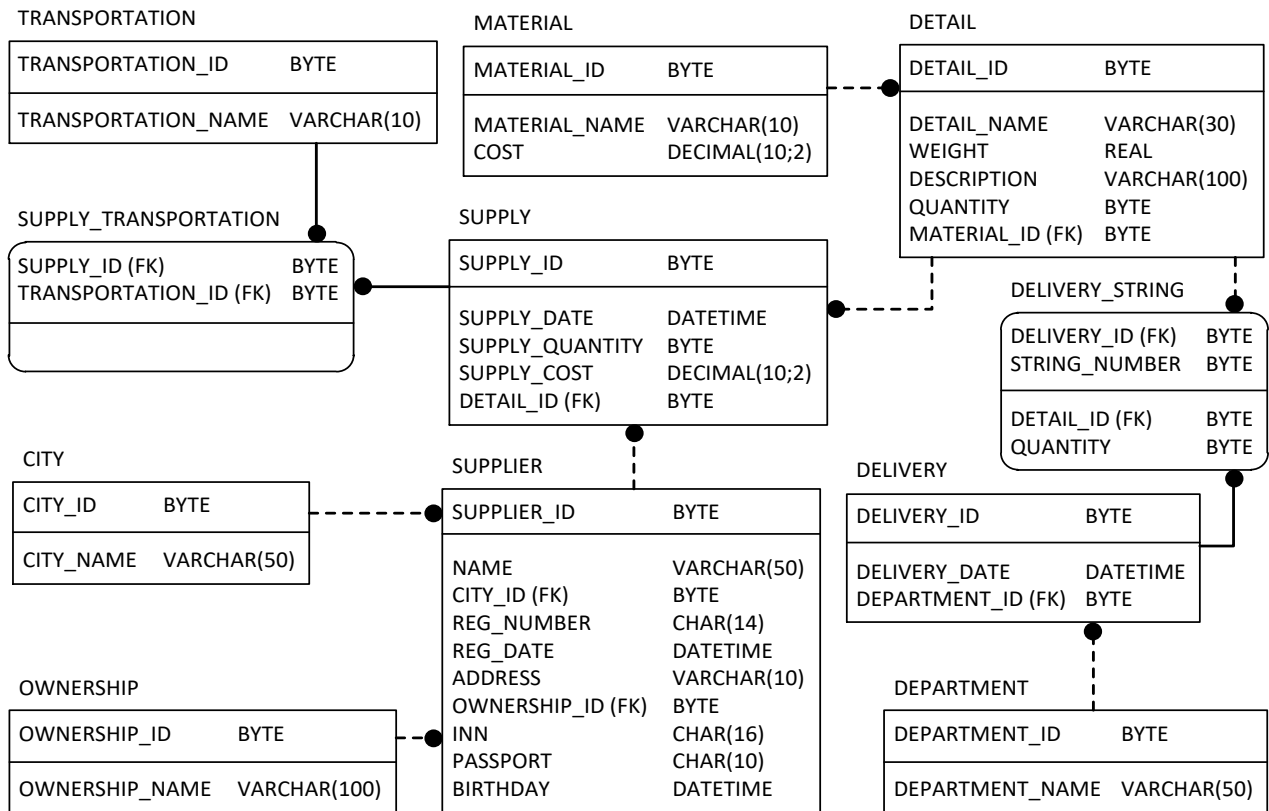


Рисунок 2.90 – Фізична модель бази даних: реалізація наслідування через міграцію нащадків у предка

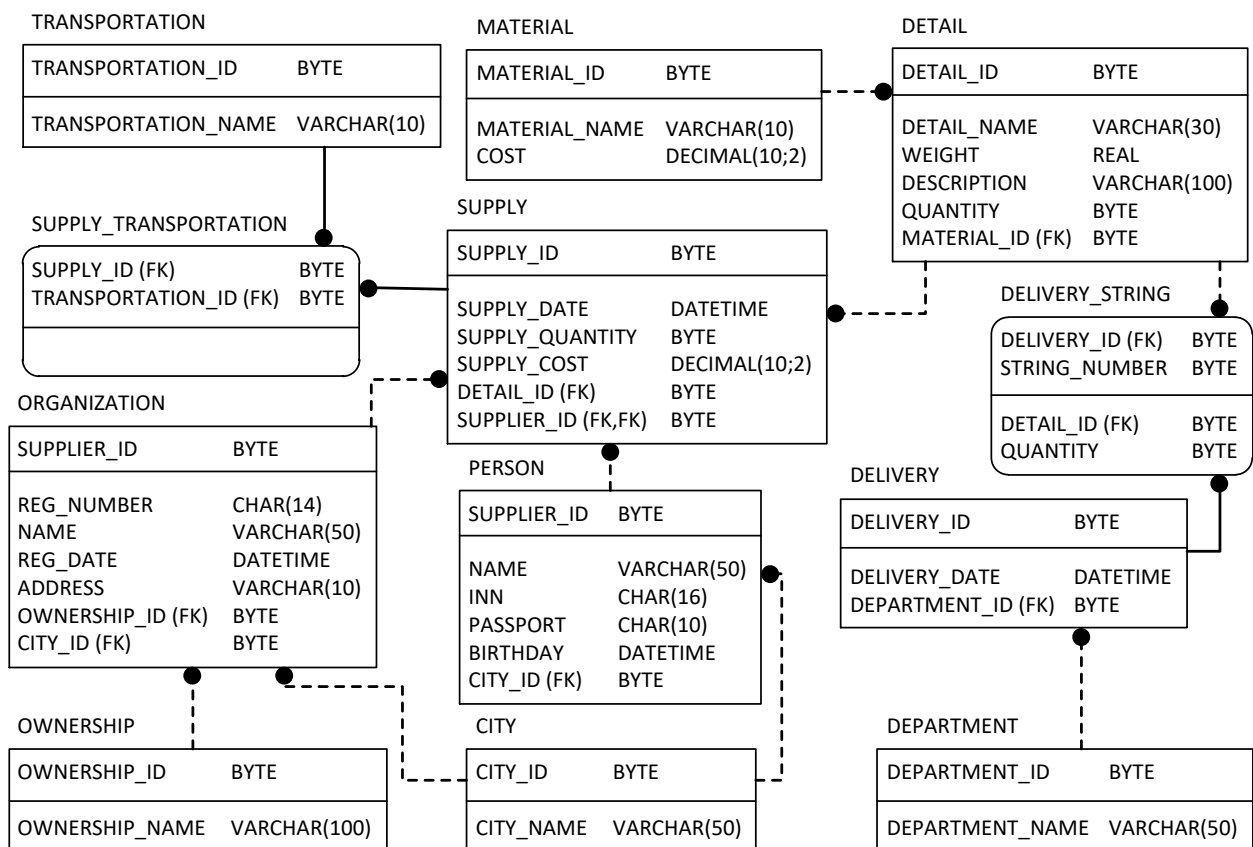


Рисунок 2.91 – Фізична модель бази даних: реалізація наслідування через міграцію предка в нащадків

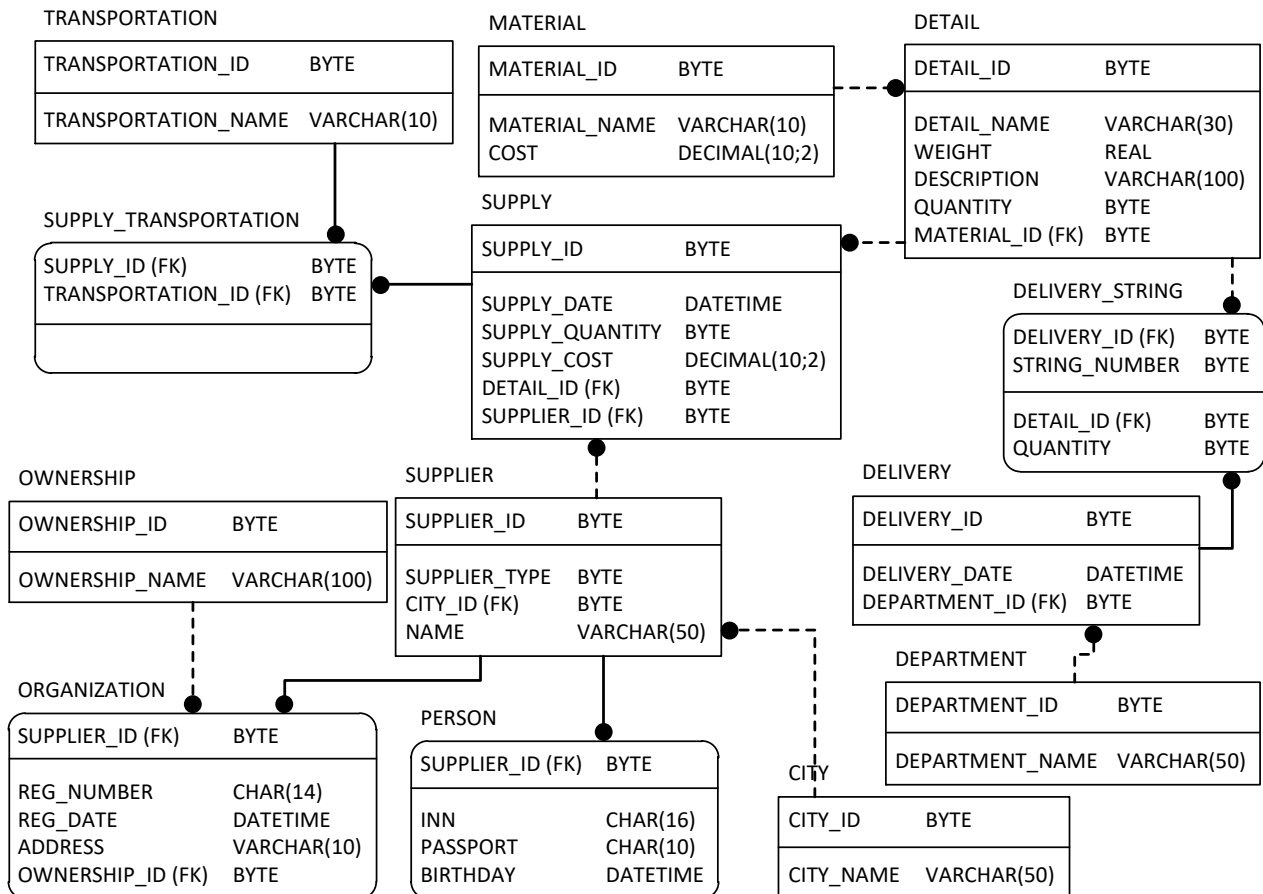


Рисунок 2.92 – Фізична модель бази даних: реалізація наслідування через заміну ієрархії ідентифікуючими зв'язками

Щоб вирішити проблему швидкого пошуку, СКБД використовують об'єкти, які називаються *індексами*. Індекс рекомендується створювати для тих колонок (або декількох колонок), за якими часто проводиться пошук.

Після опису логічної моделі проектувальник може вибрати необхідну СКБД і CASE-засіб (наприклад, ERwin) для автоматичного створення відповідної фізичної моделі. Тим самим досягається масштабованість – створивши одну логічну модель даних, можна згенерувати фізичні моделі під будь-яку підтримувану ERwin СКБД. ERwin також дозволяє вирішити задачу по перенесенню структури даних з одного сервера на інший. Наприклад, можна перенести структуру даних з Oracle на Informix або перенести структуру dbf-файлів в будь-яку реляційну СКБД, тим самим полегшивши задачу щодо переходу від файл-серверної до клієнт-серверної ІС.

3 РОЗРОБКА І ТЕСТУВАННЯ СИСТЕМИ

Розділ розробки (кодування) і тестування системи дипломної роботи повинен включати таке:

- загальні відомості про алгоритми, які використовуються в роботі;
- вибір алгоритмів роботи системи;
- функціональний аналіз алгоритму;
- опис логічної структури;
- опис і обґрунтування методу організації вхідних і вихідних даних;
- опис і обґрунтування вибору апаратних і програмних засобів (у тому числі необхідні розрахунки);
- вибір мови програмування та СКБД;
- застосовані методи оптимізації програмного продукту;
- надійність програми.

Розгляд наведених положень при виконанні дипломної роботи підпорядковується задачам, які визначаються як темою роботи, так і необхідністю продемонструвати вміння застосовувати на практиці методологію розробки програмних додатків.

3.1 Розробка системи

Етап розробки (20-25 сторінок) передбачає моделювання та реалізацію програмного забезпечення ІС, а саме вибір програмного середовища, розробка алгоритмів та кодування. Обов'язково необхідно відобразити процес розробки бази даних та проектування інтерфейсу користувача.

Процес розробки ПЗ необхідно будувати з використанням технологій моделювання та верифікації ПЗ, зокрема може бути застосована технологія Model Checking.

Обґрунтування вибору засобів реалізації. Вибір програмних та апаратних засобів для розв'язання поставлених задач проводиться на підставі постановки задачі, розроблених структури та алгоритму функціонування розроблюваного об'єкту. Необхідно навести аргументи на користь вибору середовища розробки, мови (мов) програмування, СКБД та інших необхідних засобів, а також конфігурації апаратних засобів. Вибір засобів розробки повинен бути спрямованим на оптимізацію процесу розробки (можуть розглядатися часові, вартісні, функціональні та інші параметри).

Опис структурної (функціональної) схеми. Загальний структурний опис системи (програми) має відображати основні структурні компоненти та зв'язки між ними. Переважно відображається структура за функціональними ознаками. При необхідності наводяться структурні схеми всієї системи та її складових. Опис подається в текстовому та графічному вигляді.

Результат уточнення структури може бути представлений у вигляді структурної та функціональної схем, які дають досить повне уявлення про проєктоване ПЗ.

Структурна схема програмної системи визначає основні функціональні частини системи, їх взаємозв'язки та призначення. Під функціональною частиною розуміють складову частину схеми (елемент): підсистему, функціональну групу або інші структурні компоненти.

Структурна схема призначена для відображення загальної структури системи, тобто її основних блоків, вузлів, частин та головних зв'язків між ними. Із структурної схеми повинно бути зрозуміло, навіщо потрібний даний елемент і як він працює в основних режимах роботи, як взаємодіють його частини. Позначення елементів структурної схеми можуть обиратись довільно, хоча загальноприйнятих правил виконання схем слід дотримуватись.

Більш повне уявлення про проєктоване програмне забезпечення з точки зору взаємодії його компонентів між собою і з зовнішнім середовищем дає функціональна схема.

Функціональна схема – це схема взаємодії компонентів ПЗ з описом інформаційних потоків, складу даних в потоках і зазначенням використовуваних файлів і пристроїв. Для зображення функціональних схем використовують спеціальні позначення (графічні символи), встановлені стандартом (ГОСТ 19.701-90 – Единая Система Программной Документации, рис. 3.1).

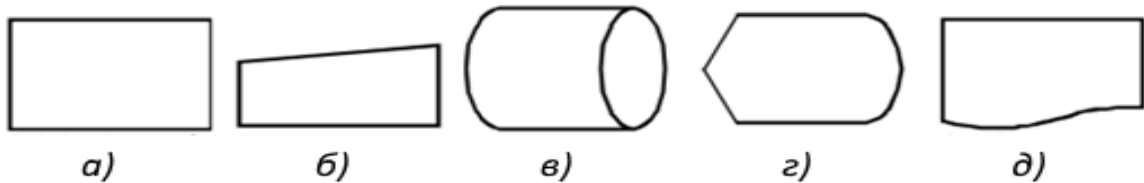


Рисунок 3.1 – Основні графічні символи для опису функціональної схеми ПЗ:
 а) – процес, підсистема; б) – ручне введення; в) – пристрій пам'яті з прямим доступом; г) – дисплей; д) – документ

Функціональні схеми інформативніші ніж структурні. На рис. 3.2 приведені структурна (а) та функціональна (б) схеми програмної системи.

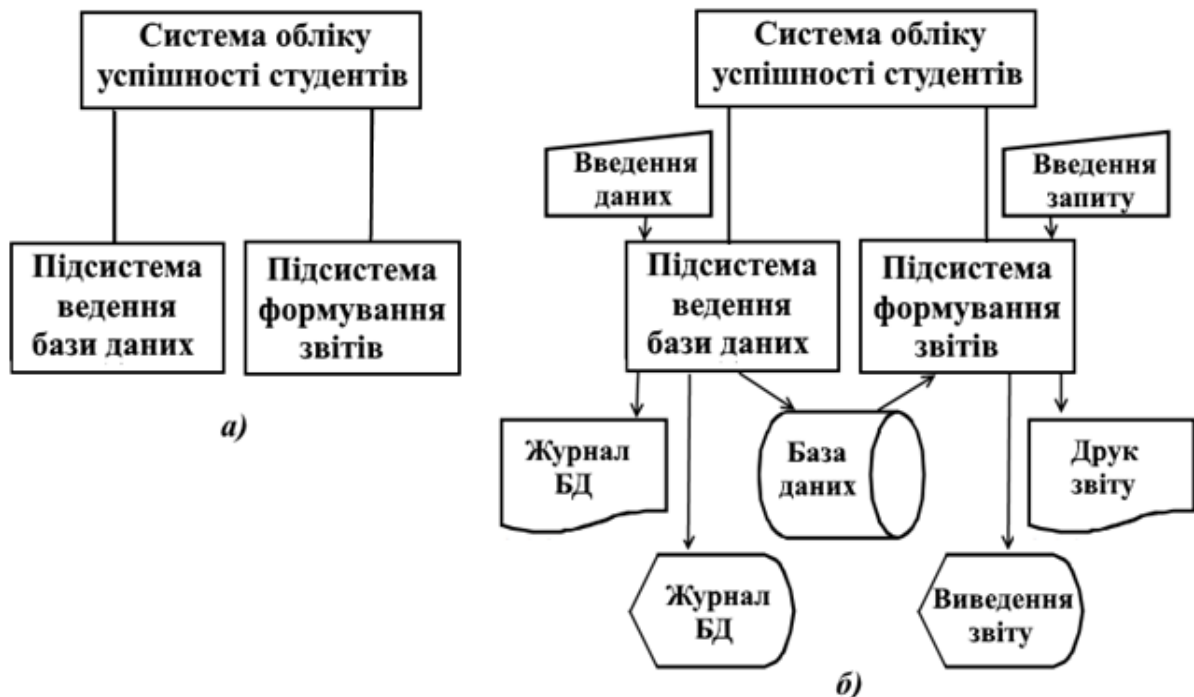


Рисунок 3.2 – Приклад схем програмної системи

Усі компоненти структурних і функціональних схем мають бути описані на різних рівнях деталізації, при цьому кількість рівнів залежить від розмірів і складності завдання обробки даних. Рівень деталізації повинен бути таким, щоб різні частини і взаємозв'язок між ними були зрозумілі в цілому.

Опис логічної схеми системи У цьому підрозділі повинна бути описана послідовність дій, що відбувається у проєктованому об'єкті, яка призводить до того чи іншого результату. Звичайною формою опису логічної схеми є опис алгоритму функціонування системи або програми, який виконується в текстовому вигляді, а потім подається його графічна інтерпретація у вигляді блок-схеми алгоритму (або у вигляді діаграм діяльності UML [7]).

Якщо об'єкт має складну структуру, рекомендується подавати узагальнену блок-схему алгоритму всієї системи та блок-схеми алгоритмів її компонентів. Ступінь деталізації опису визначається за погодженням з керівником роботи, але представлена логічна структура повинна однозначно описувати основні інформаційні процеси, які відбуваються у проєктованому об'єкті, та їхній характер: операції зберігання, обробки, відображення та передачі даних, циклічні структури, переходи за умовою тощо.

Реалізація фізичних аспектів бази даних. Невід'ємною частиною чисельної категорії інформаційних систем є бази даних. БД є єдиним і великим сховищем даних, яке визначається один раз, а потім багаторазово (і спільно) використовується різними користувачами або функціональними частинами системи.

Проектування бази даних – це ітераційний процес, який має свій початок, але не має кінця і складається з нескінченної низки уточнень. Його слід розглядати перш за все, як процес пізнання. Як тільки проєктувальник приходить до розуміння роботи системи і сенсу оброблюваних даних, а також висловлює це розуміння засобами обраної моделі даних, набуті знання можуть показати, що потрібне уточнення і в інших частинах проєкту. Особливо важливу роль в загальному процесі успішного створення системи грає концептуальне і логічне проектування бази даних. Якщо на цих етапах не вдасться отримати повне уявлення про роботу системи, то задача визначення всіх необхідних для користувача уявлень або забезпечення захисту бази даних стає надмірно складною або навіть нездійсненною. До того ж буде важче визначення способи фізичної реалізації або досягнення прийнятної продуктивності системи.

У ході третьої фази процесу проектування бази даних (перша фаза – концептуальне проектування, друга фаза – логічне проектування) розробник повинен був прийняти рішення про те, як перетворити логічний проект бази даних (тобто сукупність сутностей, атрибутів, зв'язків і встановлених обмежень) в проект фізичної бази даних, яка реалізується в середовищі обраної цільової СКБД. Оскільки багато аспектів фізичного проектування баз даних істотно залежать від типу обраної цільової СКБД, то можливе існування більш ніж одного способу реалізації будь-якого заданого елемента бази даних. Цілком очевидно, що розробник повинен добре знати функціональні можливості обраної цільової СКБД, а також чітко розуміти всі її переваги і недоліки, що дозволить йому приймати обґрунтовані рішення при виборі того чи іншого методу реалізації бази даних. Крім того, у кожному конкретному випадку розробник повинен вміти вибрати оптимальну стратегію розміщення і зберігання даних.

Фізичне проектування бази даних на четвертій фазі являє собою процес підготовки опису *реалізації* цієї бази у вторинній пам'яті. Створюється опис таблиць бази даних і обраних для них структур зберігання, а також методів доступу, які будуть використовуватися для ефективного доступу до даних. Високоякісний проект реалізації таблиць бази даних може бути створений тільки розробниками, які добре знають всі функціональні можливості обраної цільової СУБД.

Таким чином, якщо при логічному проектуванні і першій фазі фізичного проектування розробник зосереджується на тому, *що треба зробити*, тоді як при фізичній реалізації БД він шукає спосіб, *як це зробити*. У кожному разі потрібно наявність різних навичок. Так, фахівець з фізичного проектування баз даних повинен чітко уявляти, як та чи інша СКБД функціонує в комп'ютерній системі, а також добре знати усі функціональні можливості цільової СКБД.

Оскільки функціональні можливості різних СКБД досить сильно відрізняються один від одного, фізичне проектування завжди тісно пов'язане з особливостями конкретної обраної системи. Однак етап фізичної реалізації бази даних не є абсолютно ізольованим від інших – як правило, між логічним і

фізичним проектуванням є постійний зворотний зв'язок, який часто охоплює і розробку додатків користувача. Наприклад, рішення, прийняті на етапі фізичної розробки (реалізації) БД з метою підвищення продуктивності системи, можуть впливати на структуру її логічної і фізичної схеми БД.

Фізичне проектування (реалізація) БД складається з таких етапів:

1. Проектування фізичного представлення бази даних.
2. Аналіз транзакцій.
3. Вибір файлової структури.
4. Визначення вторинних індексів.
5. Аналіз необхідності введення контролю надлишкових даних.
6. Визначення вимог до дискової пам'яті.
7. Розробка механізмів захисту.
8. Визначення прав доступу.
9. Організація моніторингу і налаштування функціонування системи.

Подальше фізичне проектування бази даних являє собою процес підготовки опису реалізації цієї бази у вторинній пам'яті. Створюється опис таблиць бази даних і обраних для них структур зберігання, а також методів доступу, які будуть використовуватися для ефективного доступу до даних. Високоякісний проект реалізації таблиць бази даних може бути створений тільки розробниками, які добре знають всі функціональні можливості обраної цільової СКБД.

Наступний етап передбачає вибір структури файлів і методів доступу, які будуть застосовані при реалізації кожної з таблиць бази даних. Як правило, кожна СКБД надає кілька альтернативних варіантів схеми зберігання даних. Виконання цього етапу передбачає проведення аналізу транзакцій, які будуть виконуватися в базі даних. Результати аналізу використовуються для вибору оптимальної файлової організації таблиць.

З метою оптимізації продуктивності системи можуть бути прийняті рішення про створення вторинних індексів або внесення в контрольованих надлишкових даних. Завершується даний етап проведенням оцінки обсягу

дискового простору, необхідного для розміщення створюваної бази даних.

Існує кілька показників, які можуть бути використані для оцінки досягнутої ефективності:

1. **Пропускна здатність транзакцій.** Цей показник є кількістю транзакцій, які можуть бути оброблені за заданий інтервал часу. У деяких системах, наприклад, в службах резервування авіаквитків, банківських службах – забезпечення високої пропускної спроможності (здатності) транзакцій є вирішальним чинником успіху усієї системи.
2. **Час відповіді.** Характеризує часовий проміжок, необхідний для виконання однієї транзакції.
3. **Дискова пам'ять.** Цей показник є об'ємом дискового простору, необхідного для розміщення файлів бази даних. Розробник повинен прагнути мінімізувати об'єм використовуваної дискової пам'яті.

Проте жоден з цих чинників не є самодостатнім. Як правило, розробник вимушений жертвувати одним з показників заради іншого, щоб досягти оптимального балансу.

Так, наприклад, *послідовні файли* є оптимальним варіантом формату файлу для таблиць, в які планується внести велику кількість записів. Їх використання малоефективне, якщо потрібний вибірковий доступ тільки до окремих записів.

Хешовані файли ефективні в тих випадках, якщо дані вибираються по точній відповідності, заданій значенню ключа. Цей формат файлів не підходить для вибірки даних за шаблоном з символами підстановки, по діапазону значень ключа, по неповному значенню ключа або за значенням атрибуту, відмінного від ключа перемішування.

Індексно-послідовні файли (ISAM) є гнучкішою структурою, ніж хешовані файли. Вони ефективні при вибірці даних по заданому значенню ключа, за шаблоном з символами підстановки, по діапазону ключів або по частині ключа. Проте індекс ISAM-файлів статичний і створюється безпосередньо при створенні самого файлу. У результаті продуктивність вибірки даних з ISAM-файлу зменшується в міру внесення змін до його даних.

Вторинні індекси є механізмом визначення додаткових ключів для таблиць бази даних, які можуть використовуватися для підвищення ефективності вибірки даних. Проте наявність вторинного індексу створює додаткове навантаження при внесенні змін до даних, що повинне враховуватися при ухваленні рішення про створення цих індексів.

У деяких випадках можна відмовитися від переваг, що досягаються при використанні повністю нормалізованих відносин, на користь забезпечення більш високого рівня продуктивності системи. Подібні рішення слід приймати тільки тоді, коли задовольнити вимоги до продуктивності системи будь-якими іншими способами неможливо. Практика показує, що денормалізація частини відносин може виявитися придатним методом, якщо рівень продуктивності системи незадовільний, дані в таблицях бази даних оновлюються відносно рідко, а кількість виконуваних запитів дуже велика.

База даних є цінним корпоративним ресурсом, тому організація її захисту є одним з найважливіших завдань розробників. Призначення наступного етапу фізичного проектування (етап 7) полягає в реалізації тих вимог до захисту даних, які були встановлені ще на етапі логічного проектування бази даних. Використовувані рішення можуть включати створення представлень користувача (видів) і організацію механізму контролю над доступом користувачів до даних, що реалізовується за допомогою засобів мови SQL.

Кожному користувачеві бази даних адміністратор цієї бази привласнює ідентифікатор користувача. Як правило, цей ідентифікатор захищається особистим паролем – з цілком очевидних причин. Будь-який SQL-оператор, який виконується в середовищі СКБД, виконується від імені певного користувача. Ідентифікатор користувача використовується для визначення того, на які об'єкти бази даних цей користувач може посилатися, а також які саме операції він може виконувати з цими об'єктами. Будь-який об'єкт, що створюється засобами мови SQL, має свого власника. Власник вказується його ідентифікатором користувача. Власник об'єкту є єдиною персоною, яка знає про існування цього об'єкту і має право виконувати над ним будь-які операції.

Привілеями (чи правами доступу) називаються дії, які користувачеві дозволено виконувати відносно конкретної таблиці або представлення. Наприклад, привілей SELECT дозволяє користувачеві вибирати інформацію з відповідної таблиці. Коли користувач створює таблицю за допомогою оператора CREATE TABLE, він автоматично призначається власником створеного об'єкта і отримує повний набір прав доступу до нього. Всі інші користувачі не мають ніяких прав доступу до створеного їм об'єкту. Для надання іншим користувачам можливості доступу до нової таблиці, її власник повинен явно надати їм необхідні привілеї, наприклад, за допомогою оператора GRANT.

Останній етап (етап 9) фізичного проектування бази даних полягає у проведенні безперервного контролю за функціонуванням створеної системи і виконанням налаштування, сприяючого досягненню її максимальної продуктивності. Більшість комерційних СКБД надає в розпорядження адміністратора бази даних набір утиліт, призначених для спостереження за функціонуванням системи і її налаштування.

Розробка інтерфейсу користувача. Розробка інтерфейсу користувача є важливою стадією конструювання. Інтерфейс забезпечує взаємодію користувача та програмно-технічного забезпечення ІС.

Ефективність інтерфейсу визначається здатністю проектувальника передбачити вимоги користувачів, які виникнуть на першому та наступному етапах використання ІС. Необхідно передбачити зростання (зміну) вимог до програмно-технічного комплексу в майбутньому та забезпечити здатність ІС до їх виконання без доопрацювання системи.

Форма або звіт повинні мати привабливий зовнішній вигляд і бути гармонійним поєднанням полів або груп полів, рівномірно розподілених на поверхні форми/звіту. При цьому у формі/звіті не повинно бути областей з дуже малою або занадто великою концентрацією полів. Крім того, поля треба розміщувати через регулярні інтервали і вирівнювати їх по вертикалі і горизонталі. Якщо екранна форма має якесь еквівалентне представлення на папері, то їх зовнішній вигляд має бути погоджений.

Всюди повинні використовуватися тільки знайомі і зрозумілі терміни або ж скорочення, вибрані із заздалегідь погодженого списку. Для поліпшення зовнішнього вигляду форми або звіту можна використати колірне оформлення. Крім того, виділення кольором може застосовуватися для найважливіших полів або повідомлень. Для досягнення оптимального результату колір слід використати узгоджено і продумано. Наприклад, у формах поля з білим фоном можуть означати поля введення, а поля з синім фоном – поля з даними, призначеними тільки для відображення на екрані.

Користувач має бути візуально проінформований про загальний простір, доступний для введення даних в кожному з полів. Це дозволить йому ще до введення даних вибрати для них найбільш відповідну форму представлення.

Користувач повинен легко визначати, які операції доступні йому для переміщення курсора у формі або звіті. Зазвичай для подібних цілей використовуються клавіші табуляції, клавіші із стрілками або покажчик миші. Користувач повинен легко визначати, які саме операції доступні йому для виправлення помилки, допущеної при введенні даних. Для цієї мети зазвичай використовуються прості механізми, подібні до натиснення клавіші <Backspace> чи повторному введенню поверх помилкових символів.

При введенні в поле неправильних даних програма повинна виводити повідомлення про помилку. Це повідомлення повинне інформувати користувача про помилки, що припустилася, і вказати діапазон допустимих значень.

Необов'язкові для введення поля мають бути явно відмічені за допомогою відповідного напису або виділення особливим кольором. Подібні поля слід розташовувати після обов'язкових для введення полів.

Коли користувач поміщає курсор миші в чергове поле, то в деякому стандартному місці (наприклад, в рядку стану цього вікна) слід вивести інформацію про це поле.

Користувач повинен ясно уявляти собі, коли процес заповнення форми буде закінчений. Проте завершення цього процесу не має бути автоматичним – доцільно виводити застережливе повідомлення, щоб при необхідності

користувач зміг ще раз проглянути введені їм дані.

Процес проектування інтерфейсу користувача в роботі повинен бути відображений шляхом висвітлення результатів виконання вимог, запропонованих відомим проектувальником інтерфейсів Якобом Нільсеном [6], та описаних в роботі [7, розділ 13].

Опис розробки програмних компонентів. У даному підрозділі необхідно представити опис розроблених модулів або інших структурних компонентів, а також навести відомості про призначення модуля, зв'язки з іншими компонентами розробки, особливості реалізації алгоритму. Опис супроводжується фрагментами тексту модуля з необхідними коментарями. Опис розробки програмних компонентів повинен містити достатні відомості щодо використаних методів, технологій, стандартних та запозичених компонентів.

3.2 Тестування системи

Результатом цього етапу (5-10 сторінок) є підтвердження коректності виконання своїх завдань ІС у цілому та її програмним забезпеченням зокрема. Тестування використовує критерій коректності виконавчого додатку для ідентифікації дефектів або для того, щоб показати мінімальний рівень прийнятності. Даний процес складається, як мінімум, з модульного, інтеграційного або системного тестування [7, розділ 14].

Тестування розробленої системи, програми або програмного комплексу в загальному випадку проводиться за схемою, етапи якої передбачають:

- визначення об'єкту та мети тестування;
- вибір технології тестування та розробка програми тестування;
- проведення тестування за розробленою програмою.

Повинні також бути розроблені та застосовані:

- тести для класів, які базуються на реалізації поведінки;
- тести для класів і малих кластерів класів, які часто утворюють шаблони,

відповідальні за реалізацію конкретних зовнішніх або внутрішніх функцій;

- тести для великих кластерів класів або підсистем, що охоплюють архітектуру об'єктно-орієнтованого додатку;
- тести для об'єктно-орієнтованої системи загалом, що базується на її перевірці в робочому режимі.

Тестування розробленої системи зазвичай проводять на таких основних рівнях тестування.

Модульне тестування – це процес перевірки окремих програмних процедур (модулів) і підпрограм, що входять до складу програм. Модульне тестування проводиться безпосереднім розробником і дозволяє перевіряти всі внутрішні структури і потоки даних у кожному модулі. Цей вид тестування є частиною етапу розробки.

Інтеграційне тестування проводиться для перевірки спільної роботи окремих модулів і передує тестуванню всієї системи як єдиного цілого. У ході інтеграційного тестування перевіряються зв'язки між модулями, їх сумісність і функціональність. Воно здійснюється незалежним тестувальником і входить до складу етапу тестування.

Системне тестування призначене для перевірки програмної системи в цілому, її організації та функціонування на відповідність специфікаціям вимог замовника. Його проводить незалежний тестувальник після успішного завершення інтеграційного тестування.

Приймальне тестування проводиться організацією, що відповідає за інсталяцію, супровід програмної системи та навчання кінцевого користувача.

Особливості цього етапу дипломної роботи залежать від специфіки теми та завдання на її виконання.

Розділ тексту має закінчуватися підрозділом – **Висновки до розділу** не більш 1-2 сторінки.

ВИСНОВКИ

Висновки є завершальною та особливо важливою частиною дипломної роботи, що має продемонструвати результати розробки, ступінь реалізації поставленої мети та завдань. У цьому розділі необхідно надати підсумок усієї виконаної роботи, отриманих результатів розробки та їхнє співвідношення із загальною метою та завданнями ДР. Висновки повинні містити відповіді на всі пункти завдання, які були визначені на розробку.

Висновки і пропозиції мають бути сформульовані в тезовій формі чітко, зрозуміло; відображати зміст роботи, її теоретичне і практичне значення, виклад основних результатів (як позитивних, так і негативних) дипломної роботи.

На підставі отриманих результатів у роботі можуть надаватися рекомендації щодо практичного та наукового використання здобутих результатів. У рекомендаціях визначають необхідні, на думку автора, подальші дослідження проблеми; подають пропозиції щодо ефективного використання результатів розробки і перспективи її подальшого розвитку.

Висновки, пропозиції і рекомендації можна розділити на підпункти у вигляді послідовно пронумерованих абзаців. При цьому кожен абзац має містити окремий завершений висновок чи рекомендацію.

Будь-які бібліографічні посилання у висновках недоречні, тут подається виключно авторський текст.

Рекомендований обсяг висновків – **2-4** сторінки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Список літературних джерел містить відомості про весь обсяг інформації, який використовується в роботі. Він певною мірою відбиває наукову етику і культуру наукової праці. Саме з нього можна зробити висновок про ступінь ознайомлення студента з наявною літературою з досліджуваної проблеми.

Список повинен бути оформлений відповідно до чинних стандартів з бібліотечної та видавничої справи, на які мають бути посилання у відповідних місцях тексту (додаток К).

Прізвище автора (авторів) подаються в називному відмінку. Якщо є два, три або чотири автори, то їх прізвища з ініціалами подають через кому в тій послідовності, у якій вони надруковані в літературних джерелах. Бібліографічні описи в переліку посилань подають за алфавітним порядком або в порядку звертання відповідно до вимог стандарту [4, підрозділ 6.12].

У тексті посилання на використане джерело виконується в межах речення у квадратних дужках із зазначенням номера джерела у списку та сторінки (сторінок) в тексті джерела, на яких знаходиться потрібна інформація (роздільник – кома). Діапазон сторінок вказується номерами першої та останньої, розділених дефісом. Якщо одночасно робиться посилання на кілька джерел, їх розділяють крапкою з комою. Наприклад:

[16, 44; 18, 7-13] – сторінка 44 у джерелі №16, сторінки 7-13 у джерелі №18.

Список використаних джерел слід розміщувати без повторів одним із таких способів: у порядку появи посилань у тексті ДР або в алфавітному порядку прізвищ перших авторів або назв книг.

Перелік використаних джерел у ДР повинен містити не менше **20** сучасних джерел (підручники та навчальні посібники, періодичні видання, журнали, монографії, матеріали конференцій, Інтернет-джерела, стандарти).

ДОДАТКИ

Додатки (якщо вони є в роботі) розміщуються після списку використаних джерел. Додатки оформляються як продовження роботи на наступних її сторінках, розміщуючи їх у порядку посилань у тексті роботи. У додатки слід відносити допоміжний матеріал, який при включенні його в основну частину роботи захиращує текст:

- допоміжні рисунки і таблиці великих розмірів;
- заповнені бланки різних документів;
- інструкції та методики;
- описи алгоритмів і програм, виконаних на ЕОМ;
- зразки документів.

Кожний додаток починається з нової сторінки, має свій заголовок, який розташовується по центру. Малими літерами з першої великої друкується слово «Додаток __» і велика літера, що позначає додаток.

Додатки слід позначати послідовно великими літерами української абетки, за винятком літер Г, Є, І, Ї, Й, О, Ч, Ь, наприклад, «Додаток А», «Додаток Б».

До матеріалів допоміжного характеру відносять громіздкі таблиці та ілюстрації; матеріали, які через великий обсяг, специфіку викладення або форму подання не можуть бути внесені до основної частини (оригінали фотографій, інструкції, методики, блок-схеми, дослідницькі анкети, форми звітності тощо). У додатках також подається лістинг програми, текст керівництва користувача та інші документи, якщо вони не виконуються у вигляді самостійних програмних документів відповідно до стандартів.

На всі додатки обов'язково повинні бути зроблені посилання в основному тексті ДР.

ВІДГУК КЕРІВНИКА ТА РЕЦЕНЗІЯ НА ДИПЛОМНУ РОБОТУ

Відгук керівника (додаток Д) – це офіційний документ, в якому керівник дає оцінку дипломної роботи та особистої фахової підготовки студента.

Рецензія на дипломну роботу (додаток Ж) є офіційним документом, в якому рецензент (співробітник ВНЗ чи сторонньої організації, яка має повноваження здійснювати рецензування дипломних робіт) дає загальну оцінку дипломної роботи студента та вказує на можливі напрямки її прикладного застосування відповідно до об'єкта дослідження.

Відгук керівника та рецензія додаються до дипломної роботи окремо в конверті або у прозорому файлі.

Відгук керівника має висвітлювати такі питання:

1. Тема дипломної роботи.
2. З огляду на програму проектування – обсяг і повнота виконання її розділів у роботі.
3. Систематичність роботи студента над ДР.
4. Ступінь самостійності виконання розділів студентом.
5. Обсяг і повнота використання студентом вітчизняних й іноземних літературних джерел за темою. Додаткові дослідження та роботи проведені студентом.
6. Можливість впровадження розроблених (отриманих) матеріалів студентом у роботі.
7. Точка зору керівника про можливості допуску ДР до захисту і конкретну оцінку роботи в балах.

У рецензії на дипломну роботу варто звернути увагу на такі питання:

1. Актуальність та новизна теми ДР.
2. Відповідність виконаної роботи поставленим задачам.
3. Теоретичний рівень досліджень, рівень інженерних рішень окремих розділів ДР. Використання досвіду вітчизняної і закордонної науки та техніки.

4. Наукова, практична і методична цінність роботи; можливі рекомендації з подальшого розвитку роботи.
5. Якість графічних робіт й оформлення ДР (відповідно до вимог стандартів).
6. У висновках рецензії повинні міститися ґрунтовний аналіз недоробок роботи, а також (бажано) не менше двох зауважень (побажань) по роботі.
7. Завершується рецензія диференційованою (загальною) оцінкою якості (у тому числі актуальності) виконаної ДР.

6 ПОРЯДОК ВИКОНАННЯ ТА ОРГАНІЗАЦІЯ ЗАХИСТУ ДР

6.1 Робота студента

Студент разом з науковим керівником уточнює коло питань, що підлягають вивченню, складає план дослідження і календарний план роботи на весь період із зазначенням черговості виконання окремих етапів.

Студент систематично працює над літературою, займається збором і аналізом первинного матеріалу, постійно тримає зв'язок з науковим керівником, доповідає про хід роботи і отримує необхідну інформацію.

У міру написання окремих розділів ДР студент представляє їх науковому керівнику, виправляє і доповнює ДР відповідно до отриманих зауважень. Після остаточного узгодження з керівником чернетки ДР можна оформляти чистий варіант.

У встановлені терміни студент звітує перед керівником про готовність роботи, у необхідних випадках – перед кафедрою.

За достовірність інформації та обґрунтованість прийнятих у ДР рішень відповідальність несе дипломник.

6.2 Організація і виконання дипломної роботи

У процесі виконання дипломних робіт кафедра створює сприятливі умови для самостійної роботи студентів. Студентам-дипломникам необхідні:

- керівництво та консультації;
- систематичний контроль за роботою;
- відповідна методична документація;
- сприяння при отриманні фактичних матеріалів та їх обробка;
- обладнані робочі місця на підприємстві та в університеті.

Безпосереднє і систематичне керівництво роботою дипломника покладається на наукового керівника, який зобов'язаний:

- видати завдання на виконання ДР;
- надавати студенту допомогу в розробці календарного графіка на весь період виконання ДР;
- рекомендувати студенту необхідну літературу за темою роботи;
- регулярно проводити консультації відповідно до затвердженого графіка;
- систематично контролювати хід роботи та інформувати кафедру про її стан;
- надавати об'єктивний детальний відгук на закінчену ДР.

У ході виконання ДР науковий керівник проводить передбачені розкладом бесіди і консультації. Тут керівник виступає як опонент, указуючи студентові на недоліки аргументування в композиції, стилі викладу матеріалу, даючи поради по встановленню наявних недоліків. При ознайомленні з рукописом керівник може перевірити точність виконаних розрахунків, правильність обраних і застосованих методик, об'єктивність висновків і пропозицій. Проте за об'єктивність початкових даних і правильність всіх розрахунків відповідальність покладається на автора ДР.

У процесі підготовки ДР керівник одночасно є і нормоконтролером. Він перевіряє відповідність оформлення роботи вимогам стандартів і методичних рекомендацій кафедри. У випадку виявлення порушень та недоліків ДР (або окремі її розділи) повертається студентові для їх усунення.

Рекомендації й зауваження наукового керівника дипломник повинен сприймати творчо. Він може враховувати їх або відхиляти на свій розсуд, оскільки відповідальність за розробку й висвітлення теми, якість змісту й оформлення ДР повністю покладається на нього.

Кафедрі надається право запрошувати консультантів з окремих розділів ДР в рахунок часу, що виділяється на наукове керівництво роботою. Консультанти запрошуються з числа професорсько-викладацького складу університету та спеціалістів підприємств відповідної кваліфікації.

6.3 Нормоконтроль дипломної роботи

Для забезпечення належної якості виконання ДР підлягає обов'язковому нормоконтролю за дотриманням встановлених вимог оформлення. Нормоконтроль ДР здійснює призначений завідувачем кафедри співробітник кафедри. Нормоконтролер встановлює графік проходження нормоконтролю. У передбачені графіком строки студент зобов'язаний особисто подати нормоконтролеру роздрукований (але не переплетений) екземпляр повністю завершеної ДР.

За результатами своєї роботи нормоконтролер підписує роботу або не підписує її, якщо оформлення роботи в цілому не відповідає вимогам, встановленим даними методичними рекомендаціями, зауваження мають системний характер та стосуються майже всіх елементів ДР.

Студент зобов'язаний протягом 1-2 днів після проходження нормоконтролю усунути зауваження щодо оформлення ДР. Керівник дипломної роботи враховує якість її оформлення та ступінь доопрацювання за зауваженнями нормоконтролера при формуванні її оцінки, яка відображається у відгуку керівника.

Дипломні роботи, що пройшли нормоконтроль, після усунення зауважень щодо їх оформлення можуть бути здані в типографію для переплітання.

6.4 Відгук керівника дипломної роботи

Не пізніше, ніж за **десять днів до захисту**, повністю завершена, оформлена відповідно до вимог і підписана студентом-здобувачем дипломна робота подається студентом керівнику на відгук і підпис.

Науковий керівник є офіційним експертом кафедри і складає письмовий відгук на остаточний варіант ДР. Оформлений письмовий відгук наукового керівника (додаток Д) має бути представлений на кафедру до захисту ДР.

Використання бланків-шаблонів неприпустимо. Оформлений письмовий відгук наукового керівника має бути представлений на кафедру до захисту ДР. У відгуку керівника дається характеристика професійних та особистих якостей студента та його роботи у процесі виконання ДР, тобто керівник повинен дати відповіді на наступні питання:

- актуальність роботи (для розв’язання яких науково-практичних питань вона призначена);
- відповідність роботи виданому завданню і вимогам до її виконання;
- наявність елементів наукової новизни;
- оцінка змісту (оригінальність, глибина розробки, коректність, обґрунтованість оцінок і висновків, самостійність і персональний внесок студента в розв’язанні поставлених завдань);
- обсяг і повнота використання студентом літературних джерел (вітчизняних й іноземних) за темою роботи;
- позитивні сторони (ступінь вирішення завдань ДР; наскільки вдало використаний теоретичний і методичний апарат тощо);
- недоліки в методиці викладу, інтерпретації, аргументації висновків, в оформленні роботи, організації підготовки;
- доцільність впровадження результатів роботи на підприємстві чи в освітній процес;
- попередня оцінка ДР за чотирибальною системою («відмінно», «добре», «задовільно», «незадовільно») і висновок про присвоєння випускнику кваліфікації відповідного освітнього ступеня (формулювання згідно з навчальним планом спеціальності).

У своїй оцінці керівник ДР відзначає ритмічність виконання роботи відповідно до затвердженого графіка, визначає ступінь самостійності, активності й творчого підходу, виявлені студентом у період підготовки ДР, якість її виконання з точки зору дотримання передбачених вимог щодо змісту її елементів, оформлення, теоретичної та прикладної цінності.

Критерії оцінювання керівником дипломної роботи подані в табл. 6.1.

Таблиця 6.1 – Критерії оцінки дипломної роботи керівником

№	Критерії оцінювання дипломної роботи	Кількість балів
1	Відповідність ДР вимогам нормоконтролю (додержання всіх правил оформлення на час проведення нормоконтролю; відсутність невиправлених зауважень щодо оформлення роботи після його проходження)	5, 7, 10
2	Дотримання графіку виконання ДР (повне виконання графіку; виконання із незначними порушеннями із своєчасним поданням готової роботи; порушення терміну подання завершеної роботи)	0, 5, 10
3	Відповідність виконаної роботи затвердженому завданню (повне або неповне виконання поставлених завдань)	5, 7, 10
4	Наявність логічної послідовності і наукового стилю викладу матеріалу (володіння студентом літературною мовою і професійною термінологією, вміння логічно, аргументовано викладати результати досліджень і розробок, вдало використовувати графічний матеріал)	0, 5, 10
5	Ступінь самостійності виконання і оригінальність матеріалів (присутність наукової дискусії при посиланнях на інформаційні джерела, наявність прототипів роботи, використання оригінальних методів обґрунтування результатів)	0, 5, 10
6	Глибина практичного аналізу проблеми (рівень розв'язання завдань дослідження, досягнення поставленої мети, загальний ступінь розкриття теми в дипломній роботі)	5, 7, 10
7	Теоретична цінність результатів (загальний рівень опрацювання літературних джерел, рівень узагальнення існуючих теоретичних підходів, методів та методик)	0, 5, 10
8	Практична цінність роботи (використання наведених пропозицій у практичній діяльності, їх ефективність, обґрунтованість висновків щодо виявлених проблем, постановкою завдань та розробкою заходів щодо їх розв'язання)	5, 10, 20, 30
	Разом максимальна сума балів	100

Робота, що набирає за відгуком керівника менше 50 балів (негативний відгук наукового керівника) **не є перешкодою для захисту дипломної роботи** на засіданні ЕК, якщо студент визнає його недостатньо об'єктивним.

Студент повинен ознайомитися з відгуком і підготувати відповіді на зауваження керівника.

Після підпису і складання позитивного письмового відгуку ДР здається на кафедру.

6.5 Рішення про допуск дипломної роботи до захисту

Попередній захист ДР студент проходить на випускній кафедрі. Під час захисту особлива увага приділяється відпрацюванню форми і змісту доповіді. При цьому визначається готовність студента до захисту в ЕК.

За п'ять днів до дня захисту студент представляє на кафедру:

1. Пояснювальну записку ДР, підписану автором, керівником (куратором) та консультантами. Назва теми ДР повинна точно відповідати формулюванню, зазначеному в наказі по університету.
2. Відгук керівника.
3. Бажано довідку про зв'язок ДР з реальним виробництвом, завірену печаткою установи чи організації, на базі якої виконано дипломну роботу.

Завідувач кафедри вивчає зміст поданого студентом комплексу документів. Якщо подана робота відповідає встановленим вимогам, завідувач кафедри підписує роботу і допускає студента до її захисту.

Якщо завідувач кафедри вважає неможливим допустити студента до захисту, то це питання розглядається на засіданні кафедри у присутності студента і керівника дипломної роботи. У результаті може бути прийняте одне із таких рішень:

- допустити дипломну роботу до захисту;
- перенести строк захисту роботи;
- видати академічну довідку про складені іспити і заліки за період навчання без присвоєння кваліфікації за спеціальністю.

Основними причинами, з яких дипломна робота не допускається до захисту є:

- робота написана на тему, що вчасно не була затверджена;
- вимоги завдання на дипломну роботу не виконані;
- виконана не самостійно або запозичено (має місце плагіат);
- структура та зміст роботи не відповідає загальним вимогам, що ставляться до дипломних робіт;
- робота неправильно оформлена;
- робота вчасно не подана до захисту.

Витяг з протоколу рішення кафедри про перенесення строку захисту дипломної роботи або про видачу довідки подається на затвердження декану факультету. Ухвалена завідувачем кафедри (або на засіданні кафедри) дипломна робота переплітається студентом, візується в завідувача кафедри з поміткою «До захисту допускається» та передається рецензенту.

Студент повинен бути ознайомлений із рецензією на дипломну роботу за два дні до офіційного захисту. За наявності зауважень у рецензії він готує короткі відповіді або заперечення, що може висловити на захисті. Після рецензування жодні виправлення в роботі не дозволяються.

6.6 Рецензія на дипломну роботу

На підставі ознайомлення з роботою, відгуком керівника та бесіди з автором роботи для одержання незалежної об'єктивної оцінки роботи дипломника здійснюється її рецензування. Для дипломних робіт освітнього ступеня «бакалавр» рецензентами можуть бути доктори або кандидати наук, які є фахівцями у відповідній професійній галузі.

Рецензія не повинна дублювати відгук керівника – у відгуку керівника дається характеристика професійних та особистих якостей студента та його роботи у процесі виконання ДР, а рецензія характеризує якості безпосередньо ДР. Випадки їх повного збігу свідчать про формальний підхід до рецензування і повинні своєчасно виявлятися завідувачем випускової кафедри, який має вжити заходів щодо недопущення цього. Одним із них є вилучення відгуку керівника з пакету документів, які передаються на рецензування.

У рецензії (додаток Ж) відзначається значення дослідження даної теми, її актуальність, відповідність виконаної роботи дипломному завданню, наскільки успішно дипломник упорався з розглядом теоретичних і практичних питань, практична і методична цінність роботи та можливі рекомендації з подальшого розвитку роботи. Потім дається розгорнута характеристика кожного розділу дипломної роботи з виділенням позитивних сторін і недоліків.

Якщо рецензія надається представниками організації, за матеріалами або за замовленням якої виконувалася дипломна робота, у ній необхідно відзначити практичну цінність отриманих результатів і стан або перспективи їх впровадження.

У висновках рецензії повинні міститися ґрунтовний аналіз недоробок роботи, а також (бажано) не менше двох зауважень (побажань) по роботі. Далі рецензент викладає свою точку зору про загальний рівень дипломної роботи і визначає її загальну оцінку за стобальною системою, але оцінка роботи та її відповідність вимогам до ДР в рецензії проставляється за чотирибальною національною шкалою («відмінно», «добре», «задовільно», «незадовільно»).

Висновок про можливість присвоєння студенту відповідної кваліфікації формулюється згідно з навчальним планом спеціальності.

Критерії оцінювання ДР рецензентом подано в табл. 6.2. Кожен критерій оцінюється дискретно, згідно із наведеними в таблиці варіантами можливої кількості балів, враховуючи наявність або відсутність зауважень.

Таблиця 6.2 – Критерії оцінки дипломної роботи рецензентом

Критерій	Характеристика критерію	Кількість балів
Ступінь самостійності виконання і оригінальність матеріалів	Враховують відгук наукового керівника і рецензію, присутність наукової дискусії при посиланнях на інформаційні джерела, наявність прототипів роботи, використання оригінальних методів обґрунтування результатів	10, 15, 20
Глибина практичного аналізу проблеми	Відбиває рівень розв'язання завдань дослідження, досягнення поставленої мети, загальний ступінь розкриття теми в дипломній роботі	10, 15, 20
Теоретична цінність результатів	Визначається актуальністю обраної теми дослідження, загальним рівнем опрацювання літературних джерел, рівень узагальнення існуючих теоретичних підходів, методів та методик	10, 15, 20
Практична цінність роботи	Визначається можливістю використання наведених пропозицій у практичній діяльності, їх ефективністю, обґрунтованістю висновків щодо виявлених проблем, постановкою завдань та розробкою заходів щодо їх розв'язання	10, 15, 20
Якість викладення змісту дипломної роботи	Відбиває ступінь володіння студентом літературною мовою і професійною термінологією, вміння логічно, аргументовано викладати результати досліджень і розробок, вдало використовувати графічний матеріал	10, 15, 20
Максимальна загальна кількість балів		100

Негативна оцінка ДР, яка може бути висловлена в рецензії, не є підставою до недопущення ДР до захисту в Екзаменаційній комісії.

Якщо рецензент ДР є співробітником іншого закладу вищої освіти, підприємства, установи, організації, то на рецензії на ДР ставиться **печатка цієї установи, яка засвідчує підпис рецензента.**

Не пізніше ніж за **три дні** до захисту рецензент подає до Екзаменаційної комісії рецензію.

6.7 Підготовка до захисту дипломної роботи

Сутність даного етапу полягає в перевірці ступеня готовності студента до захисту ДР.

Перед захистом ДР студент повинен підготувати та погодити з керівником текст виступу (доповідь) та демонстраційні матеріали. Форму захисту ДР студент вибирає, виходячи з технічних можливостей та побажань. Захист може проходити за однією з перерахованих нижче форм.

1. За відсутності графічних матеріалів (та без слайдів) студент складає лише доповідь. Цей варіант є найгіршим, бо ускладнює розуміння матеріалу, що викладається, і може бути використаний лише для підсумкових частин комплексних дипломних робіт.

2. Розробляється кілька пакетів графічного матеріалу (відповідно до кількості членів ЕК), які роздаються до початку засідання. Доповідь студента ілюструється показом слайдів (не більше **25** слайдів), діапозитивів тощо. Прийнятний за наявності відповідних технічних можливостей.

3. Якщо потрібно, то розробляється комплект плакатів формату не менше ніж А1 (841x594 мм). Останній варіант є доволі трудомістким, проте дозволяє провести показовий попередній захист та значно поліпшує сприйняття матеріалу Екзаменаційною комісією.

У структурному відношенні доповідь можна розділити на три частини. Перша частина доповіді у своїх основних моментах повторює вступ до ДР. Тут обґрунтовується актуальність обраної теми, дається характеристика наукової проблеми, формулюються мета, завдання ДР, об'єкт і предмет дослідження.

Друга, найбільш ємна частина доповіді, характеризує в логічній послідовності результати аналітично-дослідницького розділу, проектування системи, отримані наукові результати, програму та аргументацію прикладних результатів.

Закінчується доповідь заключною частиною, що будується на основі тексту висновків дипломної роботи.

Доповідь має бути узгоджена з графічним матеріалом, який надається всім членам ЕК та демонструється під час доповіді присутнім за допомогою технічних засобів.

Для успішної презентації дипломної роботи студенту рекомендується заздалегідь скласти короткі тези (орієнтовно 3-4 сторінок), а також пройти попередній захист.

Попередній захист дипломних робіт може організовуватися за рішенням профільної кафедри. Його мета – здобуття навичок ефективної презентації роботи, раціональний розподіл виділеного на доповідь часу, правильне розміщення акцентів на ключових результатах виконаної роботи.

6.8 Процедура захисту дипломної роботи

Захист дипломної роботи може проводитися як в університеті, так і на підприємствах, в установах та організаціях різних форм власності, для яких тематика роботи має науково-теоретичну чи практичну цінність.

Захист дипломних робіт проводиться на відкритому засіданні ЕК за участі не менш ніж **половини його складу**.

Порядок захисту дипломних робіт визначається «Положенням про організацію освітнього процесу у вищих навчальних закладах».

До Екзаменаційної комісії перед захистом ДР подаються такі документи:

- довідка деканату про виконання студентом навчального плану і оцінок по теоретичних дисциплінах, курсових роботах, практиках;
- дипломна робота у двох екземплярах: перший – на папері, другий – одним суцільним файлом на електронному носії (електронна версія дипломної роботи має бути ідентичною паперовій);
- відгук наукового керівника;
- рецензія на дипломну роботу.

За наявності графічний матеріал розвішується на спеціальних стендах, презентаційний файл готується для показу на проекторі.

Запрошуючи чергового студента до захисту, секретар Екзаменаційної комісії коротко повідомляє присутніх про особу здобувача, що буде захищатися, тему його дипломної роботи, прізвища керівника і рецензента, середній бал студента за весь період навчання в університеті тощо.

Тривалість захисту однієї атестаційної роботи на здобуття ступеня освіти «бакалавр» становить не більше **20 хвилин**, з яких до **40 відсотків може відводитися на доповідь** здобувача (7-8 хвилин). Значна частина часу захисту відводиться дискусії та можливості для присутніх поставити питання здобувачу.

У випадку комплексної дипломної роботи захист, як правило, планується і проводиться на одному засіданні ЕК, причому студенту, який захищається першим, доручається доповісти як про загальну частину роботи, так і про індивідуальну частину зі збільшенням (за необхідності) часу на доповідь. Усі студенти, які виконували комплексну ДР, повинні бути повною мірою обізнані із загальною частиною роботи і готові до запитань членів комісії не тільки з індивідуальної, а й із загальної частини роботи.

Здобувачу надається право в довільній формі висвітлити основні ідеї, отримані результати та ступінь виконання завдання і насамперед те, що зробив безпосередньо сам здобувач, з використанням демонстраційного матеріалу – таблиць, графіків, діаграм, схем чи відповідних слайдів. Головне призначення демонстраційного матеріалу – детально і наочно проілюструвати основні положення роботи, тому необхідно вчасно посилатися на відповідний матеріал, акцентувати на ньому увагу членів ЕК.

Краще доповідь не читати з аркушу, а вивчити її і розповісти про дипломну роботу, що стане додатковим плюсом в очах комісії. Але, на всяк випадок, здобувач повинен мати письмовий варіант промови для захисту ДР.

Важливо говорити впевненим тоном, дивитись в очі слухачам. Не слід говорити комісії про своє хвилювання або показувати його аудиторії. Під час доповіді належить звертатися до всього підготовленого демонстраційного матеріалу, коротко пояснюючи його зміст.

Типова структура доповіді й розподіл часу мають бути такими:

1. Почати промову необхідно зі звернення до голови Екзаменаційної комісії та її членів. Наприклад: *«Шановний голово та шановні члени Екзаменаційної комісії, Вашій увазі пропонується дипломна робота на тему ...»*. У промові до ДР обов'язково повинні бути вказані актуальність роботи, її об'єкт, предмет, мета і завдання. Озвучення у процесі захисту ДР її новизни буде позитивно оцінено комісією (0,5-1 хв.).

2. Дати стислий аналіз існуючих методів розв'язання поставлених в роботі задач та аналогів існуючих систем із зазначенням переваг і недоліків, а також з урахуванням вітчизняного й закордонного досвіду. Обґрунтування вибраного шляху рішень цієї проблеми (1-3 хв.).

3. Спеціальна частина ДР повинна бути висвітлена так, щоб підкреслити самостійну творчість дипломника, суть виконаної роботи, її новизну (4-5 хв.).

4. У висновках необхідно дати стислий виклад результатів (як позитивних, так і негативних), які отримані в дипломній роботі. Висновки повинні містити відповіді на всі пункти завдання, які були визначені на розробку, а також зазначено ступінь впровадження розробки та перспективи її подальшого розвитку (0,5-1 хв.).

5. Наприкінці доповіді необхідно подякувати аудиторії за увагу. Наприклад: *«Дякую за увагу»* або *«Доповідь закінчено. Дякую за увагу»*.

Після доповіді члени ЕК, що є присутніми на захисті, можуть задавати здобувачу питання, на які він має відповісти (дозволяються питання й з боку присутніх на захисті). Питання можуть бути пов'язані з темою дипломної роботи, а можуть бути значно ширшими, ніж тема роботи, оскільки захист має на меті виявити не тільки знання по темі дослідження, але й рівень загальної підготовки дипломника-випускника за обраною спеціальністю.

Питання допомагають краще розкрити переваги роботи, тому необхідно уважно вислухати кожне запитання, уточнити при необхідності його значення. Це дасть необхідний запас часу для обміркування своєї відповіді. Відповідаючи, необхідно дотримуватися тільки суті питання, не відхилятися в деталі.

Після відповідей на запитання слово може надаватися керівникові роботи, який характеризує студента та його роботу. Після виступу керівника зачитується відгук керівника і рецензента, його зауваження і рекомендована оцінка, після чого надається заключне слово дипломникові, в якому він може відповісти на зауваження керівника та рецензента.

Голова ЕК пропонує бажаним висловитися про враження від дипломної роботи, що захищається. На цьому процедура захисту дипломної роботи закінчується. Хід захисту фіксується у протоколі ЕК.

Ухвалення рішення відбувається в самому кінці, тобто після завершення виступів всіх студентів, намічених цього дня, на **закритому засіданні ЕК**. Студенти та присутні на цей час звільняють приміщення. Голова ЕК оголошує результати захисту дипломних робіт.

В разі успішного захисту ЕК виносить рішення про присвоєння відповідної кваліфікації за спеціальністю, з якої відбувся захист. Рішення ЕК є **остаточним і оскарженню не підлягає!**

Студенти, які захистили дипломну роботу з оцінкою «відмінно» і мають середній бал не нижче від 4,75 (за відсутності задовільних оцінок), за рішенням ЕК можуть одержати диплом з відзнакою.

Дипломна робота, автор якої припустився **плагіату, знімається з розгляду** незалежно від стадії підготовки без права її повторного захисту.

Науковий керівник цієї роботи також як і дипломник несе персональну відповідальність за перевірку роботи на плагіат.

Студент, який не виконав ДР в установлений термін або отримав незадовільну оцінку на захисті, відраховується з університету. За рішенням ЕК він може представити дипломну роботу до повторного захисту з доробками, або взяти для розробки нову тему, що окремо затверджується кафедрою. Йому надається право повторно захищати ДР впродовж трьох років після закінчення теоретичного курсу навчання. При повторній невдалій спробі захисту ДР дипломнику видається академічна довідка встановленого зразка без присвоєння кваліфікації.

Електронні версії захищених ДР зберігаються на випускових кафедрах і розміщуються на сайті факультету. Кафедра має право використовувати матеріали ДР у освітньому процесі з метою його поліпшення. При використанні матеріалів цих робіт посилання на них є обов'язковим.

6.9 Оцінювання дипломної роботи

Результати захисту дипломних робіт обговорюються ЕК на закритому засіданні.

У загальному випадку дипломна робота, в якій розкрито тему, здійснено аналіз певної інформаційної та статистичної БД, сформульовано висновки без необхідного їх обґрунтування не може бути оцінена вище «задовільно».

Робота, в якій зроблено власну оцінку використаних джерел, самостійно проаналізовано підібраний матеріал, зроблено висновки та сформульовано пропозиції, але вони не є достатньо аргументованими, може бути оцінена на «добре».

Робота, в якій зроблено власну оцінку різноманітних джерел, зроблені власні дослідження і на їх основі – аргументовані висновки та обґрунтовано пропозиції, може бути оцінена на «відмінно».

При оцінюванні дипломної роботи враховуються вміння студента-дипломника:

- визначати мету і завдання роботи (дослідження), скласти план роботи;
- вести бібліографічний пошук із застосуванням сучасних інформаційних технологій;
- опрацьовувати отримані дані, аналізувати і синтезувати їх на базі відомих літературних джерел;
- оформляти результати досліджень відповідно до сучасних вимог.

Детальна оцінка виконаних робіт здійснюється на підставі об'єктивних критеріїв за 100-бальною шкалою.

1. Головні методичні аспекти роботи:

- 1) актуальність обраної теми (1-10 балів);
- 2) чіткість формулювання мети і головних завдань дослідження (1-10 балів);
- 3) відповідність логічної побудови плану роботи поставленим цілям і завданням (1-10 балів);
- 4) якість і глибина теоретико-методологічного аналізу проблеми (1-10 балів);
- 5) якість критичного огляду літературних джерел (1-10 балів);
- 6) достовірний аналіз фактичних матеріалів, наявність і переконливість узагальнень і висновків аналізу (1-20 балів);
- 7) глибина практичного аналізу проблеми і застосування сучасних методів дослідження, використання інформаційних технологій (1-20 балів);
- 8) ступінь самостійності виконання дипломної роботи (1-10 балів);
- 9) доступність розумінню і логічна послідовність у викладенні текстової частини, якісне оформлення роботи (1-5 балів);
- 10) зауваження і пропозиції, що містяться в рецензії і відгуку наукового керівника (1-5 балів).

2. Якість захисту роботи (1-20 балів):

- 11) уміння стисло, послідовно і чітко викласти в доповіді сутність і результати дослідження;
- 12) наявність і якість демонстраційного матеріалу для захисту ДР;
- 13) здатність аргументовано і послідовно відстоювати свою точку зору;
- 14) якість відповідей на питання членів Екзаменаційної комісії.

Критерії оцінки дипломної роботи подані в табл. 6.3.

Таблиця 6.3 – Оцінки захисту дипломної роботи

За шкалою ECTS	За національною шкалою	Загальна кількість балів	Коротка характеристика
A	відмінно (5)	90-100	Робота написана на основі особистих досліджень і бездоганна у виконанні, містить елементи новизни, має практичне і наукове значення; доповідь є логічною і короткою, відгук і рецензія позитивні, відповіді на запитання членів ЕК аргументовані, переконливі і логічні.
B	дуже добре (4)	82-89	Тема роботи розкрита, але мають місце окремі недоліки непринципового характеру: робота носить описовий характер, елементи новизни чітко не висвітлені, недостатньо використані інформаційні матеріали, неповнота посилань на використані джерела, мають місце окремі зауваження у відгуку та рецензії. Доповідь логічна, викладена вільно, відповіді на запитання членів ЕК в основному правильні і аргументовані.
C	добре (4)	75-81	
D	задовільно (3)	68-74	Тема роботи в основному розкрита, але мають місце недоліки змістовного характеру: нечітко сформульована мета роботи, теоретичний розділ має виражений компліятивний характер. Не на всі запитання членів ЕК отримана відповідь. Є істотні зауваження з оформлення роботи.
E	достатньо (3)	60-67	
F		1-59	Нечітко сформульована мета роботи. Розділи погано пов'язані між собою. Аналітична частина роботи виконана поверхнево, неповно, переважає компліятивна описовість. Оформлення роботи виконане зі значними помилками. Доповідь прочитана за готовим текстом. Відповіді на запитання членів ЕК неточні чи неповні. Рецензія на роботу негативна або ж із суттєвими зауваженнями.

Оцінка якості оформлення роботи відбиває ступінь володіння студентом літературною мовою і професійною термінологією, вміння логічно, аргументовано викладати результати досліджень і розробок, вдало використовувати графічний матеріал. Враховується також дотримання стандартів та методичних рекомендацій щодо оформлення дипломної роботи.

Презентація роботи характеризує вміння студента стисло, змістовно, переконливо і наглядно демонструвати основні результати роботи, відповідати на поставлені в ході захисту запитання, а також на зауваження наукового керівника і рецензента.

У випадках, коли захист ДР визнається незадовільним, ЕК встановлює чи може студент подати на повторний захист ту саму роботу з доопрацюванням, чи він повинен опрацювати нову тему, визначену випусковою кафедрою.

Повторний захист ДР з метою підвищення оцінки не дозволяється.

Назва теми захищеної роботи та отримана оцінка заносяться в залікову книжку і у виписку з екзаменаційних відомостей (вкладиш), прикладену до диплома про закінчення університету.

Після захисту ДР на здобуття освітнього ступеня «бакалавр» зберігається в університеті (на кафедрі або в архіві університету) як документ суворої звітності й студенту не повертається.

7 ОФОРМЛЕННЯ ДИПЛОМНОЇ РОБОТИ

7.1 Загальні вимоги

Загальні вимоги до структури, викладення і оформлення дипломної роботи розроблені на підставі Державних стандартів України (ДСТУ): ДСТУ 3659 3008-2015, ДСТУ 8302:2015, ГОСТ 2.105-95, ГОСТ 7.32-2001 [1-4].

З огляду на вимоги нормативних документів необхідно неухильно дотримуватися порядку подання окремих видів текстового матеріалу, таблиць, формул та ілюстрацій, відомостей про джерела, включені до списку тощо.

У даних методичних рекомендаціях використані такі стандарти:

- ДСТУ ГОСТ 7.1 – 2006 Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання;
- ДСТУ 7.80 – 2007 Система стандартів з інформації, бібліотечної та видавничої справи;
- ДСТУ 8302:2015 БІБЛІОГРАФІЧНЕ ПОСИЛАННЯ. Загальні положення та правила складання;
- ГОСТ 7.82 2001 Бібліографічний запис. Бібліографічний опис електронних ресурсів;
- ДСТУ 1.15:2015 Правила розроблення, викладання та оформлення національних нормативних документів.

Текст дипломної роботи подають у вигляді спеціально підготовленого рукопису у твердій палітурці.

Дипломну роботу друкують машинним способом (за допомогою комп'ютера) на одній стороні аркушу білого паперу формату А4 (210x297 мм) до тридцяти рядків на сторінці та вирівнюють «по ширині».

Загальний обсяг ДР на здобуття освітнього ступеня «бакалавр» повинен становити **60-80** аркушів (у тому числі таблиці, схеми, графіки, діаграми тощо), без урахування додатків.

Для написання ДР використовують шрифт тексту – Times New Roman; розмір шрифту (кегель, вимірюється в пунктах) – 14 пт; міжрядковий інтервал – 1,5 (інтерліньяж, 21 пунктів). Наприклад, 10 пунктів – це 3,53 мм, 12 пунктів – 4,23 мм, 14 пунктів – 4,94 мм.

Текст роботи необхідно друкувати, залишаючи береги таких розмірів: ліве – 25 мм, праве – 15 мм, верхнє – 20 мм, нижнє – 20 мм.

Шрифт друку повинен бути чітким, щільність тексту – однаковою. Текст ДР не прийнято перевантажувати маркерами, які використовуються для виділення елементів списку, курсивом, іншими шрифтами.

У дипломній роботі прізвища та ініціали, одиниці виміру, при вказівці дат необхідно відокремлювати нерозривними пробілами. Нерозривні пробіли ставляться за допомогою одночасного натискання на клавіатурі клавіш Ctrl + Shift + Space bar (пробіл). Нерозривні пробіли в режимі недрукованих знаків зображуються у вигляді маленького кружечка «°».

Окремі помилки, описки, графічні та інші неточності припускається виправляти підчищенням або зафарбуванням білою фарбою за допомогою коректора та нанесенням на те ж саме місце виправленого зображення від руки, але не більше трьох виправлень на сторінці. Вписувати в текст окремі слова, формули, умовні позначки можна тільки чорними чорнилами, чорною пастою або тушшю.

Прізвища, назви закладів, організацій, фірм та інші власні назви в тексті дипломної роботи наводять мовою оригіналу. Допускається транслітерувати власні назви і наводити назви організацій в перекладі на мову звіту, додаючи (при першій згадці) оригінальну назву.

Всі лінії, літери, цифри і знаки повинні бути однаково чорними впродовж усього документу.

Відступ у абзацах повинен бути однаковим впродовж усього тексту і дорівнювати п'яти знакам (**1,25 см**).

Окрім чіткого дотримання описаних вище норм оформлення роботи, слід також дотримуватися таких загальних правил оформлення текстів:

- не допускається наявність у тексті подвійних пробілів, пробілів перед початком та після абзацу, пробілів перед знаками пунктуації (крапками, комами, двокрапками тощо);
- в усьому тексті роботи використовуються однакові типи маркерів для маркованих списків, причому допускається вживання лише таких маркерів: «←», «а)» «1.», «1)» (без лапок);
- в усьому тексті роботи використовуються однакові типи лапок: «*текст*»;
- стиль «напівжирний» у тексті застосовується лише при оформленні заголовків розділів, підрозділів та пунктів роботи, а також для позначення заголовків елементів вступу («Актуальність роботи», «Мета і задачі розробки» тощо). Для інших цілей він не використовується;
- не припускається використання кольорового шрифту;
- стиль «курсив» у тексті може використовуватись для позначення окремих фрагментів тексту, на які автор хоче звернути увагу. Курсивом можуть оформлюватися елементи формули в тексті, для однозначного трактування. Наприклад, формула $C=A+B$. У тексті йде роз'яснення, що таке A , B і C , тоді їх виділяють курсивом для наочності;
- стиль «підкреслений» у тексті не використовується.

7.2 Вимоги до мови і стилю написання роботи

Пояснювальна записка та демонстраційний матеріал повинні виконуватися державною мовою.

Перед написанням ДР необхідно ретельно обміркувати її зміст та рубрикацію, відібрати ілюстративний матеріал, створити цифрові таблиці.

Найважливіша вимога до дипломних робіт – прагнення до лаконічності та точності формулювань. Не рекомендується вживати вирази-штампи, вести виклад від першої особи («я вважаю», «я думаю», «я пропоную», «мені здається», «на мою думку» тощо).

Щоб уникнути суб'єктивних суджень і зосередити основну увагу на предметі висловлювання, у тексті застосовують переважно виклад від третьої особи («*автор рекомендує*») або від множини першої особи («*на нашу думку*», «*зазначаємо, що*»), тобто на Вашу і керівника. Часто використовують безіменну форму подачі інформації («*як було сказано*»).

При згадуванні в тексті прізвищ учених-дослідників ініціали, як правило, ставляться перед прізвищем (Х. М. Дейтел, а не Дейтел Х. М., як це прийнято у списках використаних джерел).

Не можна набирати в різних рядках прізвища та ініціали до них, а також відділяти один ініціал від іншого при переході на другий рядок.

Числа, що характеризують кількість і не мають при собі одиниць вимірювання, від 1 до 9 у тексті прийнято записувати словами. Тобто правильно писати – «*результати трьох досліджень*», а не – «*результати 3 досліджень*». Але багатозначні кількісні числа пишуть цифрами, наприклад, «*результати 12 досліджень*» тощо.

Порядкові однозначні та багатозначні числівники краще писати словами, наприклад: «*... на другу добу*», «*... повторювали три рази*» тощо. Порядкові номери з'їздів, конгресів, конференцій краще наводити римськими цифрами. Однозначні порядкові числівники під час переліку можна наводити цифрами: «*4, 5, 6 варіант досліджень*», або «*цю залежність відображають рівняння 2 та 3*».

Порядкові числівники, що позначені арабськими цифрами, при записі після риски пишуть так: одну останню літеру, якщо вони закінчуються на голосну (крім «о» та «у») або на приголосну; дві літери, якщо вони закінчуються на приголосну та голосну «о» та «у». Наприклад, 3-я декада (а не 3-тя декада), 15-й день (а не 15-тий день), 10-го класу (а не 10-о класу), у 7-му рядку (а не у 7-у рядку).

Будь-які кількісні числівники пишуть цифрами, якщо після них стоять одиниці виміру: «5 г», «25 мл», «3 см» тощо. Числівники у складі прикметників пишуть завжди цифрами: «8 компонентна». При відображенні відсотків можливо написання як «15 – відсотковий розчин» так і «15% розчин».

Знаки градуса ($^{\circ}$), хвилини ($'$), секунди ($''$), відсотка ($\%$) від попередніх чисел не мають бути відокремлені пропуском, а від подальших чисел мають бути відокремлені пропуском ($10^{\circ} 15'$).

Індекси і показники між собою і від попередніх і подальших елементів набору не мають бути розділені пропуском ($H_2O, m^3/c$)

Текст дипломної роботи слід старанно вчитати та перевірити на наявність помилок.

7.3 Вимоги до нумерації

7.3.1 Нумерація сторінок дипломної роботи

Сторінки ДР нумерують наскрізно арабськими цифрами, охоплюючи додатки. Номер сторінки проставляють праворуч у верхньому куті сторінки без крапки в кінці.

Титульний аркуш входить до загальної нумерації сторінок звіту. Номер сторінки на титульному аркуші не проставляють. Далі йдуть сторінки завдання й анотації (в загальну нумерацію роботи вони не входять), перелік скорочень та умовних позначень (якщо в них є необхідність), сторінки змісту. Наступна порядкова нумерація сторінок починається зі змісту.

Сторінки, на яких розміщено рисунки й таблиці, охоплюють загальною нумерацією сторінок роботи.

7.3.2 Нумерація розділів, підрозділів, пунктів, підпунктів

Текст основної частини ДР поділяють на розділи, підрозділи, пункти та підпункти, які слід нумерувати арабськими цифрами без крапки в кінці.

Такі структурні частини роботи, як «АНОТАЦІЯ», «ЗМІСТ», «ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ», «ВСТУП», «ВИСНОВКИ», «СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ», «ДОДАТКИ» не мають порядкового номеру, а їх назви (напівжирним шрифтом) є заголовками структурних елементів, які друкують тим же кеглем, але великими (прописними) літерами і вирівнюють по центру сторінки. Всі аркуші, на яких розміщені згадані структурні частини, нумерують звичайним чином.

Розділи ДР повинні мати порядкову нумерацію в межах всієї дипломної роботи та позначатися арабськими цифрами без крапки в кінці.

Заголовки розділів треба друкувати великими літерами напівжирним шрифтом, не підкреслюючи, без крапки в кінці, і вирівнювати по центру сторінки. Перенесення слів у заголовку розділів, підрозділів не дозволяється.

Кожний розділ роботи слід починати з нової сторінки. У разі підрозділів, пункту чи підпункту цього робити не слід.

Для розділів, підрозділів і пунктів наявність заголовка обов'язкова. Підпункти можуть мати заголовки.

Заголовки підрозділів, пунктів і підпунктів (якщо вони є) слід починати з абзацного відступу і друкувати маленькими літерами, крім першої великої, без крапки в кінці, напівжирним шрифтом. Номер підрозділу складається з номера розділу і порядкового номера підрозділу, між якими ставлять крапку.

Після номеру розділу, підрозділу (пункту, підпункту) крапка не ставиться. Наприклад: «2.3» (третій підрозділ другого розділу). Потім у тому ж рядку йде заголовок підрозділу. Підпункти нумерують у межах кожного пункту за такими ж правилами. Після номера підпункту крапку не ставлять.

Якщо заголовок розділу, підрозділу, пункту або підпункту складається з двох або більше речень, їх розділяють крапкою.

Відстань між заголовком, приміткою, прикладом і подальшим або попереднім текстом має бути не менше ніж два міжрядкових інтервали (**28 пунктів**). Відстань між рядками заголовка, а також між двома заголовками приймають такою, як у тексті ДП.

Не дозволено розміщувати назву розділу, підрозділу, а також пункту й підпункту на останньому рядку сторінки, а також підкреслювати заголовки, скорочувати та переносити в них слова.

Якщо розділ або підрозділ складається з одного пункту, або пункт складається з одного підпункту, його не нумерують.

Посилання в тексті на розділи (підрозділи) виконується за формою: «...наведено в розділі 3 (підрозділі 3.2)».

7.4 Рисунки

Усі графічні матеріали ДР (ескізи, діаграми, графіки, схеми, фотографії, рисунки тощо) повинні мати однаковий підпис «Рисунок». Ілюстрації слід розміщувати по центру безпосередньо після тексту, де вони згадуються вперше, або якнайближче до нього на наступній сторінці, а за потреби – в додатках до ДР. На всі рисунки мають бути посилання в дипломній роботі. Нумерація рисунків має бути згідно з розділами.

Якщо рисунки створені не автором дипломної роботи, необхідно при їх поданні дотримуватись вимог чинного законодавства про авторські права – вказати використовуване джерело.

Номер і назву розташовують під графічним матеріалом, набирають тим же шрифтом, що і решту тексту, вирівнюють «по центру» без використання відступу першого рядка з одинарним міжрядковим інтервалом.

ДСТУ 3659 3008-2015 пропонують такий запис ілюстрації [1]:

Рисунок номер – Назва ілюстрації

наприклад, «Рисунок 3.1 – Схема розміщення»

Ілюстрація позначається словом «Рисунок», яке разом з номером і назвою ілюстрації розміщують по центру після пояснювальних даних.

Рисунки необхідно нумерувати арабськими цифрами в межах кожного розділу, при цьому номер рисунка складається з номера розділу та порядкового номера рисунка в цьому розділі, які відокремлюють крапкою, наприклад, «Рисунок 3.2» – другий рисунок третього розділу. Крапка в кінці назви не ставиться крім випадків, коли наявність крапки обумовлена скороченням або одиницями виміру. Номер ілюстрації відділяється від її назви рискою (–, тире).

Приклад назви ілюстрації наведено нижче:

Рисунок 6.1 – Обсяг контрактів з цінними паперами за період 2016-2017 р.р.

Не слід захоплюватися надлишком написів на самому рисунку – їх кількість мусить бути мінімальною. Найкраще замість написів на рисунку наводити умовні позначки (букви або цифри), які потім пояснювати в підпису під рисунком (підрисунковий підпис) або в тексті, що йде слідом. Підрисунковий підпис (експлікація, лат. *explicatio* – пояснення) будується так: деталі сюжету (умовні позначки) позначають цифрами (літерами або графічними символами), які виносять у підпис, супроводжуючи їх текстом **перед назвою рисунка**. Наприклад:

1 – комп'ютер; 2 – дані; 3 – документ; 4 – дисплей; 5 – процес
Рисунок 3 – Графічні позначки основних блоків алгоритмів

Рисунок відокремлюється від попереднього та наступного тексту вільним рядком, та виконують на одній сторінці аркуша. Якщо він не вміщується на одній сторінці, його можна переносити на наступні сторінки. У такому разі назву рисунка зазначають лише на першій сторінці, пояснювальні дані – на тих сторінках, яких вони стосуються, і під ними друкують:

«Рисунок ____, аркуш ____».

На всі рисунки мають бути посилання за формою: «... як це видно з рис. 3.1», або «... як це показано на рисунку 3.1», або «... на рис. 3.1 – 3.5», або «... відповідно до рис. 3», або в дужках по тексту (рис. 3.6). Посилання на раніше наведений рисунок дають за скороченим словом «дивись» (див. рис. 1.2).

Як правило, деяка частина ілюстративного матеріалу в роботі наводиться у вигляді графіків. Вони повинні мати осі координат з позначенням величин, які на них відображуються.

Якщо графік має експериментальні точки, то при великій насиченості графіка експериментальними точками, що належать різним кривим, найкраще використовувати різну форму для відображення точок, які належать різним експериментальним кривим (круглі, трикутні тощо), та різний колір або різну фактуру отриманих кривих (прямих).

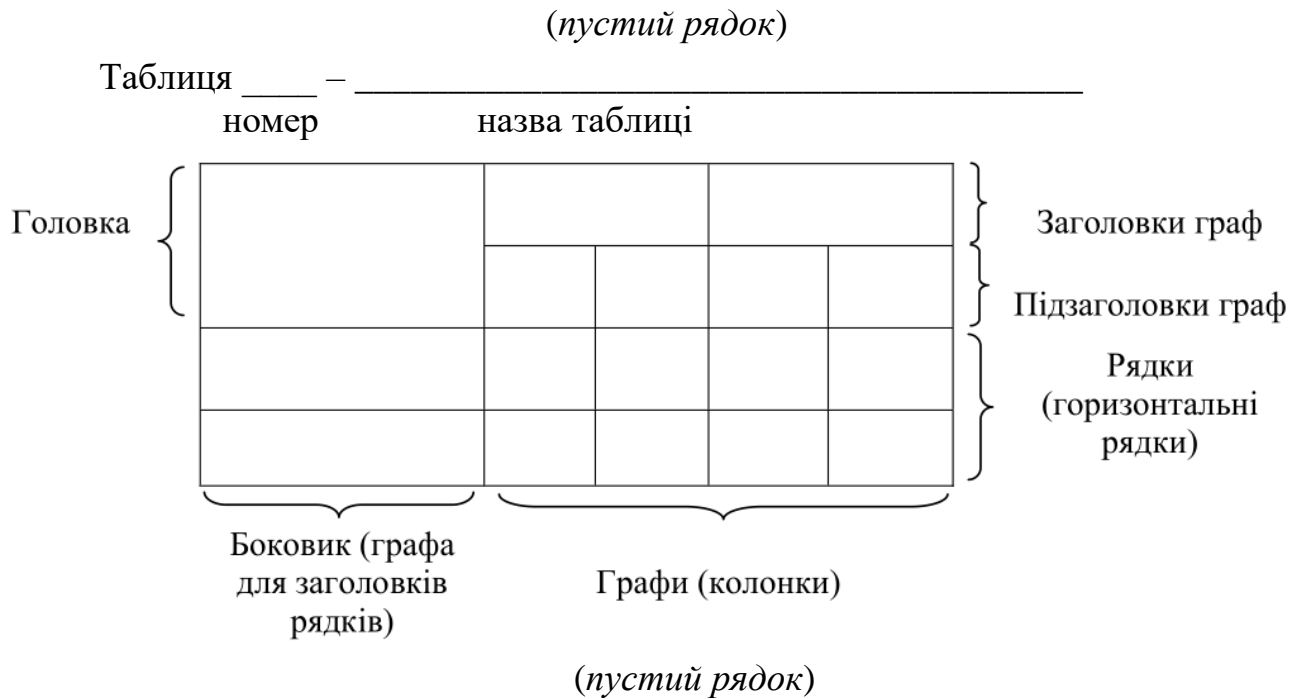
7.5 Таблиці

Для зручності викладання тексту дипломної роботи цифрові дані слід оформлювати у вигляді таблиць. Друкувати таблиці можна 12 або 10 кеглем, припускається використання одинарного міжрядкового інтервалу. Таблиці, які містять менший кегль, виносять у додатки.

Горизонтальні й вертикальні лінії, що розмежовують рядки таблиці, можна не наводити, якщо це не ускладнює користування таблицею.

Таблиці не повинні виступати за поля основного тексту. Назву таблиці друкують з великої літери і розміщують над таблицею з **абзацного відступу**. Назва містить: слово «Таблиця», номер арабськими цифрами, риску (–, тире), назву (з великої літери), одиниці виміру (якщо вини єдині для всіх показників таблиці). Крапка в кінці назви не ставиться крім випадків, коли наявність крапки обумовлена скороченням або одиницями виміру. Назва має бути стислою і відбивати зміст таблиці, її не підкреслюють.

Оформлюють таблиці у вигляді відповідно до рисунку [1]:



Таблицю слід розташовувати безпосередньо після тексту **по центру**, в якому вона згадується вперше, або на наступній сторінці. На всі таблиці мають бути посилання в тексті ДР.

Таблиці нумерують арабськими цифрами в межах розділу, крім таблиць у додатках. У цьому разі номер таблиці складається з номера розділу та порядкового номера таблиці, відокремлених крапкою, наприклад, «Таблиця 2.1» – перша таблиця другого розділу.

Таблиці кожного додатка нумерують окремо. Номер таблиці додатка складається з позначки додатка та порядкового номера таблиці в додатку, відокремлених крапкою. Наприклад, «Таблиця В.1 – _____», тобто перша таблиця додатка В.

Заголовки колонок таблиці починають з великої літери, а підзаголовки – з малої літери, якщо вони становлять одне речення із заголовком.

Підзаголовки, які мають самостійне значення, подають з великої літери. У кінці заголовків і підзаголовків таблиць крапки не ставлять. Переважна форма іменників у заголовках – однина.

Текст у шапці таблиці повинен бути розміщений по центру колонки. Текст у основній частині таблиці повинен рівнятися по лівому боку. Числові значення в колонках таблиці повинні рівнятися по правому боку колонок. Не припускається наявність порожніх комірок, тобто якщо значення відсутнє, необхідно проставити риску (–).

Якщо рядки або колонки таблиці виходять за межі формату сторінки, таблицю поділяють на частини, розміщуючи одну частину під іншою або поруч, чи переносять частину таблиці на наступну сторінку. У кожній частині таблиці повторюють її «головку» та «боковик».

У разі поділу таблиці на частини дозволено її «головку» чи «боковик» замінити відповідно номерами колонок або рядків, нумеруючи їх арабськими цифрами в першій частині таблиці.

Слово «Таблиця» подають лише один раз над першою частиною таблиці. Над іншими частинами таблиці з **абзацного відступу** друкують «Продовження таблиці __ » або «Кінець таблиці __ » без повторення її назви.

Приклад оформлення таблиці (табл. 7.1, з переносом таблиці на наступний аркуш).

Таблиця 7.1 – Основні техніко-економічні показники діяльності підприємства за 2016-2017 роки

ПОКАЗНИКИ	Одиниці виміру	201А	201Б	201В	Темп росту 201В до 201Б, %
1	2	3	4	5	6
1 Комп'ютери IBM PC	штук	17961	23759	26000	109,43
2 Реалізація продукції	тис. грн.	45500	75000	99000	132,00
3 Основні фонди (залишкова вартість)	тис. грн.	18713	18400	19000	103,30
4 Оборотні фонди	тис. грн.	14973	16000	16500	103,10
5 Об'єм продукції в діючих цінах	тис. грн.	50930	79367	99700	125,60
6 Об'єм продукції в порівняльних цінах	тис. грн.	58281	75522	82440	109,20

Продовження таблиці 7.1

1	2	3	4	5	6
7 Прибуток	тис. грн.	7720	10352	13000	125,58
8 Собівартість продукції	тис. грн.	37334	69015	86690	125,61
9 Рентабельність продукції	%	17,1	15	15	100,00
10 Витрати на гривню товарної продукції	коп.	0,838	0,87	0,87	100,00
11 Чисельність ППП	чоловік	1261	1333	1335	100,10

При наведенні номеру показника в таблиці після номеру крапка не ставиться, а залишається інтервал і з великої літери наводиться назва показника. Якщо одиниці виміру єдині для окремих колонок таблиці, то вони вказуються в цих колонках, а для окремих рядків таблиці вони вказуються через кому разом з назвою показника, або для одиниць виміру вводиться додаткова колонка.

Таблиця відокремлюється від попереднього та наступного тексту вільним рядком. Після назви таблиці вільний рядок не залишається.

На всі таблиці мають бути посилання за формою: «наведено в табл. 3.1»; «... в табл. 3.1 – 3.5» або в дужках по тексту (табл. 3.6). Посилання на раніше наведену таблицю дають зі скороченим словом «дивись (див. табл. 1.3)».

Таблиці треба заповнювати за правилами, які відповідають ДСТУ 1.5:2015.

7.6 Формули та рівняння

Формули та рівняння розташовують безпосередньо після тексту, в якому вони згадуються, посередині сторінки. Вище і нижче кожної формули або рівняння повинно бути залишено один **вільний рядок**.

Формули та рівняння в тексті (за винятком формул і рівнянь, наведених у додатках) слід нумерувати порядковою нумерацією арабськими цифрами в межах розділу. Номер формули або рівняння складається з номера розділу та порядкового номера формули, відокремлених крапкою, наприклад, формула (1.3) – третя формула першого розділу. Номер формули зазначають на рівні

формули в дужках у крайньому правому положенні на рядку.

Пояснення значень символів і числових коефіцієнтів, що входять до формули, слід наводити безпосередньо під формулою в тій послідовності, в якій вони наведені у формулі чи рівнянні.

Пояснення значення кожного символу та числового коефіцієнта слід давати з нового рядка. Перший рядок пояснення починають **без абзацного відступу** з нового рядка словом «де» **без двокрапки**, далі один під одним наводяться умовні позначення з поясненням, далі через кому наводяться одиниці виміру та ставиться крапка з комою, далі з нового рядку (з **абзацу**). Після пояснення останнього умовного позначення ставиться крапка.

При наведені формул не бажано використовувати замість умовних позначень повну назву показників. Приклад запису формули, на яку йде посилання в тексті роботи:

... відомо, що

$$Z=(M_1-M_2)/(D_1^2+D_2^2)^{1/2}, \quad (7.1)$$

де M_1, M_2 – математичне очікування;

D_1, D_2 – середнє квадратичне відхилення міцності та навантаження.

Якщо формула не вміщується в один рядок, то її можна перенести на наступний рядок тільки на знаках операцій, що виконуються – рівності (=), плюс (+), мінус (–), множення (x) і ділення (/) – при цьому знаки на початку наступного рядка повинні повторюватися. Формули, що йдуть одна за одною й не розділені текстом, відокремлюють комою до її номера, наприклад:

$$f1(x, y) = S1 \text{ і } S1 \leq S1 \text{ max}, \quad (7.2)$$

$$f2(x, y) = S2 \text{ і } S2 \leq S2 \text{ max}. \quad (7.3)$$

Вказувати розмірність параметра поряд з формулою не дозволяється.

Посилання на формули вказують порядковим номером формули в дужках, наприклад: «... у формулі (3.1)». Формула входить до речення як його рівноправний елемент. Тому в кінці формул в тексті й перед ними розділові знаки ставлять відповідно до правил пунктуації.

Числові значення величин з допусками наводять так:

$(65 \pm 3) \%$;

$80 \text{ мм} \pm 2 \text{ мм}$ або $(80 \pm 2) \text{ мм}$.

Діапазон чисел фізичних величин наводять, використовуючи прикметники «від» і «до». Наприклад,

від 1 мм до 5 мм (а не від 1 до 5 мм).

Якщо треба зазначити два чи три виміри, їх подають так: 80 мм х 25 мм х 50 мм (а не 80 х 25 х 50 мм).

Детальнішу інформацію стосовно запису числових значень див. ДСТУ 1.5: 2015 [8].

7.7 Переліки

У тексті ДР можуть бути наведені переліки, що складаються як з закінчених, так і незакінчених фраз.

Якщо перелік складається з окремих слів (або невеликих фраз без розділових знаків), вони пишуться в підбір з іншим текстом і відокремлюються один від одного комою, наприклад:

«Фірма Microsoft поставляє MS Office в такому складі: Word, Excel, Access, PowerPoint».

Якщо перелік складається з розгорнутих фраз із власними розділовими знаками, то перед кожною позицією переліку слід ставити малу літеру української абетки з дужкою, або, не нумеруючи – тире. Перед кожною позицією перерахування слід ставити цифру з дужкою. Такі переліки подають з малої літери з **абзацного відступу**. Після кожного з пунктів переліку ставиться крапка з комою, а після останнього – крапка, наприклад:

До складу пакету Microsoft Office входить програмне забезпечення для роботи з різними типами документів:

- текстами;
- електронними таблицями;
- базами даних тощо.

Коли елементи переліку є закінченими фразами (або декілька фраз), їх пишуть з абзацними відступами як звичайний текст, починаються з номеру та назви переліку з великої літери, і відокремлюють один від одного крапкою, наприклад: Взнявши за основу класифікацію систем В. П. Белогурова, можна дати таку характеристику системі розрахунків:

1. За здатністю до змінення вона є динамічною – під впливом зовнішніх і внутрішніх факторів елементи системи розрахунків здатні переміщуватись.

2. За умовами функціонування вона є ймовірнісною – тобто це система, кінцевий стан якої можна передбачити з певним ступенем ймовірності.

Коли елементи переліку є закінченими фразами (або декілька фраз) і містять перелік певних однорідних об'єктів, або кількісний перелік, їх можна оформляти як звичайний перелік, починаються з номеру та назви переліку з великої літери, і відокремлюють один від одного крапкою, наприклад:

. . . базою фінансового аналізу є дані:

1. Баланс підприємства (форма № 1).
2. Звіт про фінансові результати (форма № 2).
3. Звіт про рух грошових коштів (форма № 3).

Для подальшої деталізації переліку слід використовувати арабські цифри з дужкою (другий рівень деталізації з **абзацним відступом** відносно місця розташування переліків першого рівня), далі – через знаки «тире». Наприклад:

а) форма і розмір клітин;

б) живий склад клітин:

1) частини клітин;

2) неживі включення протопластів.

в) утворення тканини.

У разі розвиненої та складної ієрархії переліків дозволено користуватися можливостями текстових редакторів автоматичного створення нумерації переліків (наприклад, цифра–літера–тире) [1].

Переліки першого рівня деталізації друкують малими літерами з абзацного відступу, другого рівня – з відступом відносно місця розташування переліків першого рівня, як правило, під першою літерою переліків першого рівня деталізації.

7.8 Примітки

Примітки – це стисла, лаконічна довідка до того чи іншого слова в тексті, яке може бути незрозуміле читачеві, або потребує пояснення. Іноді примітки роблять до авторських скорочень.

Примітки розташовують відразу після тексту, який потребує пояснення. Примітки розташовують безпосередньо після тексту, ілюстрації або таблиці, до яких вони відносяться. Примітки до таблиці розміщують під лінією, яка визначає кінець таблиці. Примітку друкують через один інтервал.

Слово «Примітка» друкують кеглем 12 через один міжрядковий інтервал з абзацного відступу з великої літери з крапкою в кінці. У тому самому рядку через проміжок з великої літери друкують текст примітки тим самим шрифтом.

Примітки нумерують послідовно в межах однієї сторінки. У випадку розташування однієї примітки на сторінці їх не нумерують, наприклад: «Примітка. Ціни наведені згідно з прейскурантом постачальника».

Якщо приміток дві та більше, їх подають після тексту, якого вони стосуються і нумерують арабськими цифрами, наприклад:

Примітка 1. _____

Примітка 2. _____

7.9 Підрядкове бібліографічне посилання (виноски)

Бібліографічні посилання оформляються згідно стандарту ДСТУ 8302:2015 (ДСТУ 8302:2015 БІБЛІОГРАФІЧНЕ ПОСИЛАННЯ. Загальні положення та правила складання. Київ, ДП «УкрНДНЦ», 2016. 20 с.) [4], який установлює основні види бібліографічних посилань, загальні положення щодо їхнього складу й структури, а також правила складання та розміщення в документах (виданнях, депонованих документах тощо).

Підрядкове бібліографічне посилання на джерела інформації використовують за умов, якщо всередині тексту документа його розмістити неможливо або небажано, щоб не переобтяжувати текст та не ускладнювати його читання. Підрядкове бібліографічне посилання розміщують **як примітку** в нижній частині сторінки (полоси набору) в повній або короткій формі, відмежовуючи від основного тексту горизонтальною рисою.

Якщо в тексті або таблиці необхідно пояснити окремі дані, то їх можна оформлювати за допомогою підрядкових бібліографічних посилань. Підрядкове бібліографічне посилання пов'язують із текстом документа за допомогою знаків виноски, які подають на верхній лінії шрифту після відповідного фрагмента в тексті (наприклад: *Текст*²) та перед підрядковим посиланням (наприклад: ² *Посилання*).

Підрядкові бібліографічні посилання в тексті розташовують у кінці сторінки, на якій вони зазначені, з абзацу. Підрядкове посилання набирають на формат основного тексту зменшеним кеглем шрифту.

Виноски позначають надрядковими знаками у вигляді арабських цифр (порядкових номерів). Наприклад, цитата в тексті: «Бібліографічні посилання оформляються згідно стандарту ДСТУ 8302:2010¹»

Відповідно подання виноски.

¹ ДСТУ 8302:2015 БІБЛІОГРАФІЧНЕ ПОСИЛАННЯ. Загальні положення та правила складання. – Київ, ДП «УкрНДНЦ», 2016. 20 с.

У підрядковому бібліографічному посиланні на електронний ресурс віддаленого доступу за наявності в тексті бібліографічних відомостей, що його ідентифікують, дозволено зазначати тільки електронну адресу, використовуючи замість слів «Режим доступу» аббревіатуру «URI» або «URL».

Наприклад, у тексті: Маніфест ІФЛА про Internet: прийнято Сесією Ради ІФЛА 23 серпня 2002 р./пер. з англ. В. С. Пашкова.*

У підрядковому посиланні:

* URL: <http://archive.ifla.org/III/misc/im-ua.pdf>. (дата звернення: 15.09.2017).

Нумерація виносок здійснюється окремо для кожної сторінки.

7.10 Позатекстове бібліографічне посилання

Позатекстове бібліографічне посилання використовують переважно в наукових виданнях у разі багаторазових посилань на одні й ті самі документи задля уникнення повторного подання однакових бібліографічних записів або через їхню велику кількість, або за браком місця для підрядкових посилань [4].

Враховуючи сучасний рівень поширення інформації та її перецитовування, важливо аналізувати надійність джерел (як друкованих, так і електронних інформаційних ресурсів). Потрібно звертати увагу на репутацію джерела інформації, а також на компетентність її автора, первинність джерела, на яке посилається автор.

Використання праць інших авторів можливе лише шляхом цитування, перефразовування або узагальнення.

При написанні ДР студент повинен робити посилання на джерела, матеріали або окремі результати, на ідеях і висновках яких досліджуються проблеми та розв'язуються поставлені завдання. Такі посилання дають змогу відшукати документи і перевірити достовірність відомостей про цитування документа, допомагають з'ясувати його зміст, мову тексту, обсяг.

Якщо в роботі використовуються відомості, матеріали з монографій, оглядових статей, інших джерел з великою кількістю сторінок, то в посиланні необхідно точно вказати номери сторінок з джерела, на яке зроблено посилання. У науковій літературі прийняті два типи цитування – вільне (не дослівне) та безпосереднє (дослівне).

Вільне цитування в науковій літературі використовується у вигляді двох типів посилань: цифрового та за прізвищем. У разі цифрових посилань, посилання в тексті на літературне джерело зазвичай оформлюють у квадратних дужках, де першою цифрою позначають номер літературного джерела у списку використаних джерел, а другою через кому – сторінку (сторінки, якщо треба), з якої запозичено цитату.

Наприклад. **У тексті:**

Правила банківського кредитування підприємств державної форми власності викладено в навчальному посібнику «Кредитування та ризики» (автори Денисенко М. П., Догмачов В. М., Кабанов В. Г.) [35, с. 123].

У позатекстовому посиланні:

35. Денисенко М. П., Догмачов В. М., Кабанов В. Г. Кредитування та ризики: навч. посіб. Київ, 2008. 213 с.

Якщо посилання наводять на багатотомний документ в цілому, зазначають номер тома (випуску, частини): [13, т. 2, с. 150-154].

Коли в посиланні певне джерело тільки згадується, номер сторінки не наводиться. У випадках, коли в посиланні згадуються кілька джерел, їхні номери допускається писати через знак «;» або (коли їх підряд три і більше) – через тире. Наприклад: Розв'язанню цієї проблеми присвячені роботи [12-15; 18; 21, с. 42-135].

При посиланнях на розділи, підрозділи, пункти, ілюстрації, таблиці, формули, додатки зазначають їх номери, наприклад: «... у розділі 4 ...», або «...дивись 2.1...»; «відповідно до 3.1.1...»; «... на рис. 2.3 ...»; «... у таблиці 3.2 ...»; «...за формулою (3.1)...»; «...у додатку Б...».

Для підтвердження власних аргументів посиланням на авторитетне джерело або для критичного аналізу того чи іншого друкованого твору слід наводити прямі цитати. При прямому цитуванні в тексті наводиться фраза або частина фрази з якого-небудь іншого джерела, або з джерела зі списку. Не слід зловживати цитатами при написанні дипломних робіт. Науковий етикет потребує точно відтворювати цитований текст, бо найменше скорочення наведеного витягу може спотворити зміст, закладений автором. Загальні вимоги до цитування такі:

1. Текст цитати починається і закінчується лапками і наводиться в тій граматичній формі, в якій він поданий у джерелі, із збереженням особливостей авторського написання. Перед цитатою або після неї вказується автор, або джерело цитованого положення, наприклад: «В умовах власної держави Довженко виріс би на світового генія», – сказав у своєму виступі Євген Маланюк [12, с. 39]. Якщо автор відомий у всьому світі, то можна наводити тільки цитату. Наприклад: Юлій Цезар казав: «Краще померти відразу, ніж все життя провести в очікуванні смерті». Або ж інший варіант: «Краще померти відразу, ніж все життя провести в очікуванні смерті», – так говорив Юлій Цезар.
2. Цитування повинно бути повним, без довільного скорочення авторського тексту та без перекручень думок автора. Пропуск слів, речень, абзаців при цитуванні допускається без перекручення авторського тексту і позначається трьома крапками. Вони ставляться в будь-якому місці цитати (на початку, всередині, наприкінці). Якщо перед цитованим текстом або за ним стояв розділовий знак, то він не зберігається.
3. Кожна цитата обов'язково супроводжується посиланням на джерело, звідки вона була запозичена.

7.11 Перелік умовних позначок (скорочень)

Якщо в роботі використовується специфічна термінологія та мало поширені скорочення, нові символи та позначки, то треба навести їх перелік у вигляді окремого списку.

Всі слова в дипломній роботі слід писати повністю. Скорочення слів проводиться відповідно до стандартів ДСТУ 3582-97 «Інформація та документація. Скорочення слів в українській мові в бібліографічному описі Загальні вимоги та правила».

У дипломній роботі допускаються лише загальноприйняті скорочення:

- замість скорочень «*і т. д.*» (і так далі), «*і т. ін.*» (і таке інше), «*і т. п.*» (і тому подібне), «*подібні*» рекомендовано вживати «*тощо*»;
- при посиланнях (наприклад: *дивись* – див., *рисунок* – рис., *таблиця* – табл.);
- при позначенні цифрами століть і років, одиниць виміру (наприклад: *ст.* – століття (однина), *ст. ст.* – століття (множина), *рік* – р., *роки* – рр.; тисяч (мільйон, мільярд) гривень – *тис.* (*млн, млрд*) *грн.*).

Слова «*та інші*», «*і таке інше*» всередині речення не скорочують. Не допускається скорочення слів «*так званий*» (т. з.), «*наприклад*» (напр.), «*формула*» (ф-ла), «*рівняння*» (р-ня), «*діаметр*» (діам.). Інші скорочення та літерні абрєвіатури, що наводяться в дипломній роботі при першому згадуванні автором, вказуються у круглих дужках після повного найменування.

При наборі розділових знаків – кома (,), крапка з комою (;), двокрапка (:), точка (.), знак оклику (!), знак питання (?) і багатокрапка (...) від попереднього слова пропуском не відділяються, але відділяються поодиноким пропуском від подальшого слова.

При наборі традиційних скорочень, а також ініціалів імені та по батькові, слід використати пропуск: *і т. п.* (і тому подібне), *т. з.* (*так званий*), *і т. ін.* (і таке інше), *і т. п.* (*і тому подібне*), *до н. е.* (до нашої ери).

Неправильно: *і т.п., Т.Г.Шевченко*; **правильно:** *і т. п., Т. Г.Шевченко.*

7.12 Оформлення списку використаних джерел

Список використаних джерел – елемент бібліографічного апарату, який містить бібліографічні описи використаних джерел і розміщується після висновків. Відомості про джерела, включені до списку, необхідно надавати відповідно до вимог державного стандарту [4].

Загалом це оформлення більш просте: прибираються зайві пробіли, замість сполучення «крапка й тире» використовуються просто крапки, прибираються деякі інші знаки та повторне зазначення імен авторів (якщо їх від 1 до 4, можна зазначати імена всіх на початку опису), дозволяється не вказувати видавництво, для електронних ресурсів введено єдине формулювання «URL» незалежно від мови статті (також може зазначатися номер DOI).

Джерела необхідно розміщувати в порядку появи посилань у тексті, або в алфавітному порядку, або у хронологічному порядку. Бібліографічний список нумерується від першої до останньої назви.

Приклади оформлення бібліографічного опису списку джерел згідно стандарту (ДСТУ 8302:2015 «Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання». Київ, ДП «УкрНДНЦ», 2016. 20 с.) [4] наведені в додатку К.

Слова і словосполучення скорочуються відповідно до стандарту:

«Інформація та документація. Бібліографічний опис. Скорочення слів і словосполучень українською мовою. Загальні вимоги та правила»: (ISO 4:1984, NEQ; ISO 832:1994, NEQ). ДСТУ 3582:2013. – [Чин. від 2014-01-01]. – Київ: Мінекономрозвитку України, 2014. – 14 с.

До списку варто включати переважно нові джерела інформації, видані не раніше 2000 року.

7.13 Оформлення додатків

Додатки слід оформлювати як продовження пояснювальної записки до дипломної роботи у вигляді окремої частини, розташовуючи їх у порядку з'явлення посилань на них в тексті записки. При цьому додатки повинні мати спільну із всією роботою наскрізну нумерацію сторінок.

Усі додатки повинні бути перераховані у змісті.

Додатки, як структурний елемент дипломної роботи, починають з окремого аркушу, на якому зверху, по центру рядка друкують великими літерами слово ДОДАТКИ.

Додатки після слова **Додаток** позначають великими літерами української абетки, за винятком літер Г, Е, Є, З, І, Ї, Й, О, Ч, Щ, Ъ. Якщо додаток один, його також позначають як «**Додаток А**». Додаток повинен мати заголовок, надрукований під словом «**Додаток _**» (симетрично відносно до тексту сторінки) маленькими літерами (з першою великою літерою, напівжирним шрифтом).

Якщо додаток розташовується на декількох аркушах, то на кожному наступному аркуші посередині першого рядка вказується «Продовження додатку _». Якщо додатками є копії документів, проспекти тощо, то перед копією слід розмістити чистий аркуш, на якому посередині пишуть слово «ДОДАТОК» та його найменування. Сторінки копій нумерують, продовжуючи наскрізну нумерацію.

Ілюстрації, таблиці, формули та рівняння в тексті додатка слід нумерувати в межах кожного додатку. Наприклад: рисунок Д.3 – третій рисунок додатка Д; таблиця А.2 – друга таблиця додатка А; формула (В.1) – перша формула додатка В.

При поданні лістингу програм, роздруківок із програм тощо допускається відхилення від правил, наприклад, шрифт Courier New або Arial з кеглем 10 пт тощо. При поданні UML-діаграм слід дотримуватись вимог до UML-нотації. При поданні блок-схем алгоритмів програм – вимог ЄСПД, а креслень – ЄСКД.

7.14 Демонстраційний матеріал

Демонстраційний матеріал до дипломної роботи призначений для допомоги студентові при захисті ДР і до його оформлення не встановлюється жодних обов'язкових вимог.

Рекомендується виносити необхідну інформацію на 12-20 слайдів (плакатів), додатково можна виконувати слайди у вигляді комплекту відповідних матеріалів на аркушах формату А4 для кожного члена Екзаменаційної комісії.

На відміну від тексту дипломної роботи, при підготовці презентації дозволено використовувати жирний, кольоровий шрифт, курсив, підкреслення. До об'єктів слайдів також можливо застосовувати анімаційні ефекти.

На слайди слід виносити: титульний аркуш (1 слайд); актуальність, мета, завдання, предмет та об'єкт дослідження (1-2 слайди); графічні матеріали (схеми, графіки, таблиці, діаграми тощо, 4-6 слайдів), математичні моделі та стислі текстові фрагменти, які дають максимальне уявлення про зміст роботи (3-4 слайдів); основні положення висновків (1-2 слайди). Всі матеріали розташовуються в тій послідовності, в якій студент має викладати доповідь по своїй випускній дипломній роботі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДСТУ 3659 3008-2015 «Документи. Звіти у сфері науки і техніки. Структура і правила оформлення». Київ: наказ ДП «УкрНДНЦ» від 22 червня 2015 р. № 61 з 2017-07-01. 26 с.
2. ГОСТ 2.105-95 «Общие требования к текстовым документам». Введен в действие с 01.07.1996. 30 с.
3. ГОСТ 7.32-2001 «Отчет о научно-исследовательской работе». Структура и правила оформления. Введен в действие с 01.07.2002. 21 с.
4. ДСТУ 8302:2015 «Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання». Київ: ДП «УкрНДНЦ», 2016. 20 с.
5. Леоненков А. В. Самоучитель UML 2. СПб.: БХВ-Петербург, 2007. 576 с.
6. Нильсен Я. Веб-дизайн. Книга Якоби Нильсена. Киев: Символ-Плюс, 2006. 512 с.
7. Авраменко В. С. Авраменко А. С. Проектування інформаційних систем. Черкаси: Чабаненко Ю. А, 2017. 434 с.
8. Купер А. Психбольница в руках пациентов. СПб.: Символ-Плюс, 2004. 336 с.
9. Рамбо Дж., Блах М. UML 2.0. Объектно-ориентированное моделирование и разработка. – 2-е изд. СПб.: Питер, 2007. 544 с.
10. Розенберг Д., Скотт К. Применение объектного моделирования с использованием UML и анализ прецедентов / пер. с англ. Москва: ДМК Пресс, 2002. 160 с.

ДОДАТОК А – Зразок титульного листа дипломної роботи**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРКАСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ БОГДАНА ХМЕЛЬНИЦЬКОГО**

Факультет обчислювальної техніки, інтелектуальних та управляючих систем
Кафедра <назва випускової кафедри>
Спеціальність 122 Комп'ютерні науки
(кегель 14 пт)

До захисту допускаю
Завідувач кафедри

(ініціали, прізвище)

(дата, підпис)

ДИПЛОМНА РОБОТА
освітнього ступеня «Бакалавр»
(кегель 14 пт)

**ІНТЕЛЕКТУАЛЬНА СИСТЕМА АДАПТИВНОГО
ТЕСТУВАННЯ ЗНАНЬ**
(кегель 16 пт)

Студент групи _____
(шифр групи) (прізвище, ім'я, по батькові) (підпис)

Науковий керівник _____
(науковий ступінь та вчене звання, прізвище, ініціали) (підпис)

Консультанти: (за рішенням кафедри)

ДОДАТОК Б – Завдання та календарний план роботи студента (зразок)

Черкаський національний університет імені Богдана Хмельницького
 Факультет _____
 Кафедра _____
 Спеціальність _____
 Освітній ступінь _____
 Форма навчання _____ курс _____ група _____

ЗАТВЕРДЖУЮ

Завідувач кафедри _____

« ____ » _____ 20__ року

**ЗАВДАННЯ
 НА ДИПЛОМНУ РОБОТУ СТУДЕНТА**

_____ (прізвище , ім'я, по батькові)

1. Тема роботи _____

_____ затверджена наказом ЗВО від « ____ » _____ 20__ року № _____

2. Термін подання студентом закінченої роботи _____

3. Вихідні дані до роботи _____

4. Зміст розрахунково-пояснювальної записки (перелік питань, які треба розробити) _____

5. Перелік графічного, наочного матеріалу _____

6. Консультант(и) дипломної роботи _____

Продовження додатку Б

7. Календарний план підготовки дипломної роботи

<i>№ з/п</i>	<i>Етапи роботи</i>	<i>Термін виконання</i>
1	Вибір, погодження й затвердження теми, призначення наукового керівника, рецензента, консультанта	
2	Складання календарного плану й розширеного плану-конспекту роботи. Опрацювання джерел	
3	Організація і проведення теоретичного, емпіричного (експериментального) дослідження	
4	Підготовка складових частин (розділів) роботи	
4.1	Вступ	
4.2	Розділ 1	
4.3	Розділ 2	
4.4	Розділ 3	
4.5	Висновки	
4.6	Список використаних джерел	
4.7	Додатки	
5	Усунення зауважень, урахування рекомендацій наукового керівника, доповнення або скорочення обсягу роботи. Узгодження виправленого варіанту всієї роботи до пропозицій наукового керівника	
6	Оформлення тексту роботи	
7	Попередній захист. Обговорення роботи на кафедрі й рекомендації її до захисту. Оформлення супровідних документів	
8	Подання роботи науковому керівникові для написання відгуку	
9	Подання роботи на рецензування	
10	Підготовка доповіді на захист	
11	Захист роботи в Екзаменаційній комісії	

Студент-дипломник _____
(підпис)

Науковий керівник _____
(підпис) _____ (прізвище, ім'я, по батькові)

« _____ » _____ 20 __ року

ДОДАТОК В – Зразок анотації дипломної роботи

АНОТАЦІЯ

Сучасний ринок інформаційних систем документообігу розвивається дуже інтенсивно і потреба в їх розробці стає усе більш актуальною. У той же час, перед розробниками стоїть задача оптимального вибору кращого варіанту інформаційної системи, що враховує специфічні особливості документообігу і взаємодії підрозділів конкретного підприємства.

У цій дипломній роботі розроблена інформаційна система документообігу підприємства «АВРОРА». Проект розроблявся на основі даних, зібраних в період переддипломній практики за інформаційному, технічному, математичному і програмному забезпеченням діючої системи документообігу підприємства.

Тема актуальна, оскільки вирішує питання підвищення ефективності діяльності усього підприємства і підвищення його конкурентоспроможності, створення нового класу інформаційних систем для малого і середнього бізнесу при максимальному використанні типових програмних продуктів, що реалізуються на сучасній науковій основі.

Проведено дослідження існуючих варіантів інформаційних систем документообігу, подано їх критичний аналіз. Спроектований і розроблений програмний комплекс, що становить основу інформаційної системи, забезпечує оптимізацію потоків документації і її оформлення для забезпечення кращої взаємодії підрозділів підприємства.

Ключові слова: ІНФОРМАЦІЙНА СИСТЕМА ДОКУМЕНТООБІГУ, ПІДВИЩЕННЯ КОНКУРЕНТОСПРОМОЖНОСТІ ПІДПРИЄМСТВА, ПРОГРАМНИЙ КОМПЛЕКС, ОПТИМІЗАЦІЯ ПОТОКІВ ДОКУМЕНТАЦІЇ, ВЗАЄМОДІЯ ПІДРОЗДІЛІВ ПІДПРИЄМСТВА.

ANNOTATION

The modern market of the informative systems of circulation of documents develops very intensively and a requirement in their development becomes more actual. At the same time, the task of optimal choice of the best variant of the informative system taking into account the specific features of circulation of documents and co-operation of subdivisions of concrete enterprise stands before developers.

In this diploma project the informative system of circulation of documents of enterprise is worked out «AURORA». A project was developed on the basis of the data, collected in the period of to diploma practice on informative, technical, mathematical and programmatic to providing of the operating system of circulation of documents of enterprise.

A theme is actual, because decides the question of increase of efficiency of activity of all enterprise and increase of his competitiveness, creations of new class of the informative systems for small and midsize businesses at the maximal use of the model programmatic foods realized on modern scientific basis.

A study of existent variants of the informative systems of circulation of documents is undertaken, their walkthrough is given. The projected and worked out programmatic complex, making basis of the informative system, provides optimization of streams of documentation and her registration for providing of the best co-operation of subdivisions of enterprise.

Keywords: INFORMATIVE SYSTEM OF CIRCULATION OF DOCUMENTS, INCREASE OF COMPETITIVENESS OF ENTERPRISE, PROGRAMMATIC COMPLEX, OPTIMIZATION OF STREAMS OF DOCUMENTATION, CO-OPERATION OF SUBDIVISIONS OF ENTERPRISE.

ДОДАТОК Д – Бланк відгуку керівника на дипломну роботу**Черкаський національний університет****імені Богдана Хмельницького**

Факультет _____

Кафедра _____

Спеціальність _____

Освітній ступінь _____

Форма навчання _____ курс _____ група _____

ВІДГУК**КЕРІВНИКА НА ДИПЛОМНУ РОБОТУ**

(прізвище, ім'я, по батькові)

Тема роботи _____

1. Ступінь обґрунтування актуальності теми: _____

2. Обсяг та структура роботи, характеристика її розділів: _____

3. Відповідність роботи меті та завданням _____

Продовження додатку Д

4. Оцінка повноти використання фактичних даних: _____

5. Використання в роботі сучасних методів та методик: _____

6. Результати дипломної роботи (теоретичні, практичні): _____

7. Оцінка наукового апарату і стилю роботи: _____

8. Ставлення студента до роботи, зауважень і побажань наукового керівника:

9. Недоліки: _____

10. Загальна оцінка дипломної роботи та висновок про можливість допуску роботи до захисту: _____

Науковий керівник _____

(прізвище, ім'я, по батькові)

(учене звання, науковий ступінь, місце роботи і посада)

« ____ » _____ 20 __ року **Підпис** _____

ДОДАТОК Ж – Бланк рецензії на дипломну роботу студента**Черкаський національний університет
імені Богдана Хмельницького**

Факультет _____

Кафедра _____

Спеціальність _____

Освітній ступінь _____

Форма навчання _____ курс _____ група _____

**РЕЦЕНЗІЯ
НА ДИПЛОМНУ РОБОТУ**

*(прізвище , ім'я, по батькові)*Тема роботи _____

_____Обсяг дипломної роботи: кількість сторінок _____, таблиць _____,
рисуноків _____, додатків _____, використаної літератури _____.1. Висновок щодо відповідності дипломної роботи меті й завданням

_____2. Коротка характеристика виконання дипломної роботи * _____

* Коротка характеристика повинна вміщувати:

- а) характеристику виконання кожного розділу дипломної роботи і ступінь використання студентом останніх досягнень науки і техніки;
- б) оцінку якості виконання графічної частини роботи та пояснювальної записки;
- в) перелік позитивних рис дипломної роботи та її основних недоліків.

ДОДАТОК К – Зразок оформлення списку використаних джерел

Один чи більше авторів без редактора

1 Федорова Л.Д. З історії пам'яткоохоронної та музейної справи у Наддніпрянській Україні. 1870-ті – 1910-ті рр. Київ, 2013. 373 с.

2 Пичугина Т.С., Баранов П.Ю., Пичугин С.А. Модель возведения комплекса объектов (расчет эффективности капитальных вложений). Харьков, 1985. 136 с.

3 Лусь В.І., Киркач Т.Є., Мандріченко О.Є., Радченко А.О. Практикум з нарисної геометрії: навч. посіб. Харків: ХНУМГ ім. О.М. Бекетова, 2014. 118 с.

4 Zhovinsky E.Ya., Kryuchenko N.O., Paparyha P.S. Geochemistry of Environmental Objects of the Carpathian Biosphere Reserve. Kyiv, 2013. 100 p.

Один чи більше авторів із редактором

5 Воробей П.А. Кримінальна відповідальність за незаконну торговельну діяльність: монографія / за ред. В.К. Матвійчука. Київ: Укр. академія внутр. справ, 1996. 116 с.

6 Закон України «Про Національну поліцію»: наук.-практ. коментар / О.І. Безпалова, К.Ю. Мельник, О.О. Юхно та ін.; за заг. ред. В.В. Сокурєнка; передм. В.В. Сокурєнка. Харків, 2016. 408 с.

Без автора (збірники, матеріали конференцій, книги за редакцією, укладачі, упорядники)

7 Україна в цифрах. 2007: стат. зб. / Держ. комітет статистики України. Київ, 2008. С. 185–191.

8 Софія Київська: Візантія. Русь. Україна. Вип. II. Київ, 2012. 464 с.

9 Правова основа діяльності органів державної влади: зб. нормат. актів / упоряд. П.М. Любченко. Харків: ФІНН, 2010. 303 с.

10 Експлуатація і технічне обслуговування газорозподільчих станцій магістральних газопроводів / заг. ред. А.А. Руднік. Київ, 2003. 370 с.

11 Electrodes of conductive metallic oxides / J.M. Honig et al. Amsterdam: Elsevier, 1980. 260 p.

12 Ресурсосбережение и энергоэффективность инженерной инфраструктуры урбанизированных территорий и промышленных предприятий: материалы II Международной научно-технической интернет-конференции (2–27 февраля 2016 г., Харьков). Харьков, 2016. 150 с.

Багатотомне видання

13 Ушинський К.Д. Людина як предмет виховання. Спроба педагогічної антропології: вибрані твори. Київ: Рад. школа, 1983. Т. 1. 480 с.

14 Юридична енциклопедія: в 6 т. / за ред. Ю.С. Шемшученка. Київ: «Укр. енциклопедія» ім. М.П. Бажана, 1999. Т. 2. 784 с.

15 Франко І. Твори: у 50 т. Т. 45. Київ, 1986. 480 с.

16 Енциклопедія історії України: у 10 т. / ред. рада: В.М. Литвин (гол.) та ін.; Ін-т історії України НАН України. Київ: Наук. думка, 2005. Т. 9. С. 36–37.

Частина видання

Розділ книги

17 Наумов М.С. Напрями впливу інтелектуалізації економіки на розвиток ринкових відносин в Україні. Трансформаційні процеси в суспільстві в умовах інформаційної економіки: монографія / В.П. Решетило, М.С. Наумов, Ю.В. Федотова. Харків, 2014. С. 213–241.

Стаття в журналі, газеті

18 Яцків Я.С., Радченко А.І. Про ефективність видання наукових журналів в Україні. Вісник НАН України. 2012. № 6. С. 62–67.

19 Мозоль А.П. Злочинність мігрантів (кримінологічна характеристика). Вісник Університету внутрішніх справ. Серія «Право». 1999. Вип. 7. Ч. 2. С. 112–119.

20 Аврамцев О. Хронограф: події в історії. Слобідський край. 2015. 27 січня. С. 8.

Тези доповідей у матеріалах конференцій

21 Скальський В.Р. Становлення методу акустичної емісії в установах Західного наукового центру. Теорія і практика раціонального проектування, виготовлення і експлуатації машинобудівельних конструкцій: праці 2 міжнар.

наук.-техн. конф. (Львів, 11–13 листопада 2010 р.). Львів, 2010. С. 9–10.

22 Борисова В.І. Зміст заповіту. Проблеми цивільного права та процесу: матеріали наук.-практ. конф., присвяч. пам'яті проф. О.А. Пушкіна (Харків, 27 травня 2016 р.). Харків: ХНУВС, 2016. С. 20–24.

Електронні ресурси

23 Наукові публікації і видавнича діяльність НАН України. Київ, 2007. URL: <http://www.nas.gov.ua/publications> (дата звернення: 19.03.2014).

24 Вільчик Т.Б. Відповідальність адвоката перед клієнтом: напрями гармонізації законодавства України до європейських стандартів. Форум права. 2016. № 1. С. 30–36. URL: http://nbuv.gov.ua/j-pdf/FP_index.htm_2016_1_7.pdf (дата звернення: 20.09.2016).

25 Харківський національний університет внутрішніх справ // Вікіпедія: вільна енциклопедія. URL:

https://uk.wikipedia.org/wiki/Харківський_національний_університет_внутрішніх_справ (дата звернення 15.09.2016).

26 Axak N. Development of multi-agent system of neural network diagnostics and remote monitoring of patient. Eastern-European Journal of Enterprise Technologies. 2016. Vol. 4, № 9(82). P. 4–11. DOI: 10.15587/1729-4061.2016.75690.

27 Авраменко В.С. Операційні системи [Електронний ресурс]: практикум / Черкаський національний ун-т ім. Б. Хмельницького. Електрон. текст, дані. Черкаси, 2014. 1 електрон. опт. диск (CD-R).

Дисертації та автореферати

(може бути не вказане місце захисту та повна назва спеціальності)

28 Черевко П.П. Створення юридичних осіб приватного права: дис. канд. юрид. наук: 12.00.03. Київ, 2008. 225 с.

29 Савченко Л.А. Особисті права та обов'язки батьків і дітей за сімейним законодавством України: автореф. дис. ... канд. юрид. наук: 12.00.03. Київ, 1997. 27 с.

30 Дутко А.О. Юридичні конструкції та їх використання в законотворчій практиці України: автореф. дис. ... канд. юрид. наук: спец. 12.00.01 «Теорія та

історія держави і права; історія політичних і правових учень»; Львівський держ. ун-т внутр. справ. Львів, 2010. 20 с.

Закони та інші нормативні документи

31 Конституція України: Закон України від 28 червня 1996 р. № 254к/96-ВР / Верховна Рада України. Відомості Верховної Ради України. 1996. № 30. Ст. 141.

32 Про Національну поліцію: Закон України від 2 липня 2015 р. № 580-VIII / Верховна Рада України. URL: <http://zakon.rada.gov.ua/laws/show/580-19> (дата звернення: 26.08.2016).

33 Про Стратегію сталого розвитку «Україна – 2020»: Указ Президента України від 12 січня 2015 р. № 5/2015 / Президент України. Офіційний вісник Президента України. 2015. № 2. С. 14. Ст. 154.

34 Загальна декларація прав людини: прийнята і проголош. Резолюцією 217 А (III) Генеральної Асамблеї ООН від 10 грудня 1948 р. // База даних «Законодавство України». URL: http://zakon.rada.gov.ua/laws/show/995_015 (дата звернення: 26.08.2016).

35 Гражданский кодекс Республики Беларусь. Минск: Национальный центр правовой информации республики Беларусь, 1999. 442 с.

36 Гражданский кодекс Российской Федерации от 30 ноября 1993 г. № 51-ФЗ. Российская газета. 1994. 8 декабря. № 238–239. Ст. 1551.

Архівні документи

37 Науковий архів Інституту історії України НАН України. Ф. 2. Оп. 3. Спр. 170. № 1. Арк. 5.

38 Приветственный адрес в честь 50-летия творческой и научно-педагогической работы А.Н. Бекетова от ректора ХИИКСа. Музейный комплекс Харьков. нац. ун-та гор. хоз-ва им. А.Н. Бекетова. Осн. Ф. 125. Вх. инв. № 225.

39 Заява приват-доцента по кафедрі філософії Київського університету П. Светлова від 23.10.1919 про необхідність читання курсу гносеології. Державний архів м. Києва. Ф. 16. Оп. 469. Спр. 381. 13 арк.

Стандарти, патенти, препринти, каталоги

40 ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. Київ, 2016. 16 с. ГОСТ Р 517721-2001.

41 Аппаратура радиоэлектронная бытовая. Входные и выходные параметры и типы соединений. Технические требования. М., 2001. 27 с.

42 Люмінісцентний матеріал: пат. 25742 Україна: МПК6 C09K11/00, G 01T1/28, G 21H3/00. № 200701472; заявл. 12.02.07; опубл. 27.08.07, Бюл. № 13. 4 с.

43 Панасюк М.І., Скорбун А.Д., Сплошной Б.М. Про точність визначення активності твердих радіоактивних відходів гамма-методами. Чорнобиль: Ін-т пробл. безпеки АЕС НАН України, 2006. 7 с. (Препринт. НАН України, Ін-т пробл. безпеки АЕС; 06-1).

44 Національна академія наук України. Анотований каталог книжкових видань 2008 року. Київ: Академперіодика, 2009. 444 с.

ДОДАТОК Л – Орієнтовна тематика дипломних робіт

Тематика ДР повинна відповідати профілю спеціальності підготовки фахівців, враховувати практичні потреби сучасних виробничих підприємств, державних організацій і соціальних служб.

Орієнтовні теми дипломних робіт для студентів кафедри ПЗАС спеціальності **121 Інженерія програмного забезпечення**.

1. Шаблони ініціалізації, завершення, синхронізації, обміну даними для багатопоточних програм на мові Java.
2. Сервер розподіленої програмної системи для реалізації двовимірної Real-Time гри.
3. Web-орієнтоване середовище моделювання задач маршрутизації.
4. Побудова програми за її автоматною моделлю.
5. Створення засобів опису програми для побудови її автоматної моделі.
6. Машинно-залежна оптимізація програми.
7. Система комп'ютерного супроводу розв'язання задач лінійної алгебри.
8. Система комп'ютерного супроводу розв'язання задач інтегрального числення.
9. Модуль захисту програмного продукту від несанкціонованого використання.
10. Інформаційно-пошукова система "Ресторація" для мобільних пристроїв на базі iOS.
11. Система безпроводної суддівської сигналізації для силових видів спорту на платформі Android з використанням BlueTooth .
12. Програмне забезпечення логістичної служби аграрного виробництва.
13. Web-орієнтована система організації та проведення змагань з пауерліфтингу.
14. Система автоматичного складання розкладу із застосуванням методу Ганта.
15. Система автоматизації роботи фінансової служби підприємства.
16. Автоматизація оформлення і обліку трудових договорів у освітній установі.

17. Автоматизація складського обліку і відпуску готової продукції на прикладі малого підприємства.
18. Автоматизація електронного документообігу на підприємстві.
19. Автоматизована система розрахунку і обліку оплати комунальних послуг.
20. АРМ економіста планово-економічного відділу підприємства.
21. АРМ менеджера по роботі з клієнтами фірми.
22. Інформаційна система автоматизації центру комп'ютерних курсів.
23. Розробка статистико-аналитической інформаційної системи.
24. Аналіз стану випадкових тимчасових рядів.
25. Розробка програмного комплексу для автоматизації процесу тестування знань.
26. Статичне виведення типів в динамічних мовах програмування на прикладі Python.
27. Система підтримки процесу атестації співробітників компанії.
28. Розробка синтаксичного аналізатора текстів для створення онтологій предметних областей.
29. Розробка системи інформаційної безпеки для комп'ютерних мереж малого бізнесу.
30. Розпізнавання рукописного тексту з використанням апарату нейронної мережі.

Орієнтовні теми дипломних робіт для студентів кафедри інформаційних технологій спеціальності **122 Комп'ютерні науки**.

1. Інформаційна система підприємства житлово-комунального господарства.
2. Інформаційна система автотранспортного підприємства.
3. Інформаційна система з продажу комп'ютерної техніки.
4. АРМ бухгалтера по заробітній платі навчального закладу.
5. Автоматизована інформаційна система підбору ліків.
6. Комп'ютерна система розпізнавання тріщин і переломів на основі рентгенівських знімків.
7. Комп'ютерна система планування харчування при лікуванні органів

шлунково-кишкового тракту.

8. Проектування інформаційної системи для прогнозування серцево-судинних захворювань.
9. Проектування інформаційної системи для прогнозування захворювань нервової системи.
10. Інформаційна система стаціонару медичної установи.
11. Інформаційна система обліку пацієнтів міської поліклініки.
12. Програмний модуль діагностування психомоторного стану людини.
13. Web-орієнтована система фінансового обліку медичного закладу.
14. Програмний симулятор клінічних ситуацій.
15. Інформаційний Інтернет-портал служби психологічної допомоги.
16. Комп'ютерний модуль контролю больового стресу.
17. Інформаційно-довідкова система аналізу якості продуктів харчування.
18. Інформаційна система автоматичного формування плейлистів музичних творів.
19. Інформаційна система вибору лікарських засобів на основі характеристик певної хвороби.
20. Розробка комплексу віртуальних комп'ютерних класів для підтримки навчального процесу.
21. Інтелектуальна система контролю стану здоров'я на мобільній платформі Android.
22. Веб-орієнтована система медико-соціального центру.
23. Інформаційна система правильного харчування.
24. Анатомічний атлас-довідник людини.
25. Інформаційна система визначення спадковості людини.
26. Інформаційна система управління лікувальним процесом.
27. Автоматизована система підбору товарів Інтернет-магазину.
28. Комп'ютерна діагностика захворювання туберкульозом.
29. Комп'ютерна діагностика захворювань гіпертонічного типу в залежності від навколишнього середовища.

30. Адаптивна інформаційна система формування раціону харчування у реабілітаційний період.
31. Автоматизована система формування НМКД.
32. Автоматизована інформаційна система планogram мережі магазинів.
33. Інформаційна система забезпечення процесу управління мережею автомобільних салонів.
34. Інформаційна система управління економічною діяльністю торгівельної фірми.
35. Автоматизована система оцінки якості роботи викладача ЗВО.
36. Web-орієнтована інформаційна система обласного відділення судинної хірургії.
37. Web-орієнтована інформаційна система обласного алергологічного відділення.
38. Web-орієнтована інформаційна система обласного відділення гемодіалізу.
39. Інформаційно-пошукова та довідкова система з працевлаштування.
40. Інформаційна система «Електронний журнал».
41. Інформаційна система сервісного центру по ремонту комп'ютерної техніки.
42. Автоматизоване робоче місце медичного працівника дошкільного закладу.
43. Браузерний редактор діаграм.
44. Система пошуку спамерів на основі аналізу користувачів сервісу Youtube.
45. Система управління контентом веб-сайту з урахуванням потреб користувача.
46. Інформаційна система кластеризації комерційних даних на основі нейромереж.
47. Система підтримки лояльності клієнтів з конструктором правил.
48. Автоматизована інформаційна система розрахунку і обліку оплати послуг зв'язку.
49. Інформаційна система роботи страхової компанії.
50. Інформаційна система автоматизації приватної юридичної фірми.

Орієнтовні теми дипломних робіт для студентів кафедри інформаційних технологій спеціальності **123 Комп'ютерна інженерія**.

1. Програмний комплекс автоматичного аналізу медичних сигналів.
2. Мобільний кардіограф.
3. Програма «Динамічна оптика» для вивчення законів геометричної оптики в тонких лінзах.
4. Програма визначення просторової орієнтації об'єкту за афінними спотвореннями.
5. Адаптивна система керування параметрами камери спостереження.
6. Інформаційна система підтримки панорамного зору робота.
7. Комп'ютерний комплекс діагностування вестибулярної функції організму.
8. Розробка комп'ютерної системи діагностики новоутворень на основі аналізу зображень.
9. Автоматизоване робоче місце кардіолога.
10. Синтез кінцевого автомату Мілі з використанням нейронних мереж.
11. Синтез кінцевого автомату Мура з використанням нейромереж.
12. Програмне забезпечення для виявлення механічних домішок в інфузійних розчинах.
13. Система дистанційного керування підтримкою тиску в чистих приміщеннях.
14. Система інтерпретації математичних виразів засобами контролерних систем.
15. Програма віддаленого керування функціями мобільного пристрою на платформі Android.
16. Програма прогнозування стану об'єктів за даними спостереження.
17. Інформаційна система руху товарів на оптовому складі комп'ютерної техніки.
18. 3-D атлас анатомії людини.
19. Моделювання біосигналів в біомедичних комп'ютерних системах.
20. Інформаційна система керування мережею пошукових агентів.

21. Інформаційна система пошукового агента.
22. Інформаційна система підбору комплексу фізичних вправ для травмованих хворих.
23. Інформаційна система навчально-тренувальних занять для футболістів.
24. Мобільна інформаційна система зняття показників у велоспорті.
25. Система підрахунку частоти пульсу за допомогою камери мобільного телефону.
26. Комп'ютерна модель робота.
27. Моніторинговий алгоритм стану мережі передачі даних.
28. Програмування системи управління робота.
29. Програмний комплекс багатокритерійної оптимізації систем на основі мультихромосомних моделей і генетичних алгоритмів.
30. Автоматизована система управління проектами для малих підприємств.
31. Розробка програмного забезпечення підтримки процесів закупівлі і обліку зберігання товарів на складі.
32. Стегоаналіз цифрових звукових файлів.
33. Система перерахунку зображення з урахуванням проектних спотворень.
34. Побудова автономного віртуального робота на основі автоматного підходу на прикладі гри «CoclekaPy».

Орієнтовні теми дипломних робіт для студентів кафедри ІСПР спеціальності **122 Комп'ютерні науки, 124 Системний аналіз.**

1. Інформаційна система моніторингу зовнішніх динамічних об'єктів.
2. Інформаційна система управління навчанням тактиці волейболістів.
3. Система розпізнавання термінів у текстових файлах формату php – html.
4. Підсистема формування статистичного опису трафіка інтернет-сайту.
5. Система підтримки прийняття рішення у виборі фінансової установи для розміщення депозитного вкладу.
6. Система автоматизованого розрахунку збитків підприємства від

забруднення довкілля.

7. Автоматизована система кадастрової оцінки земельних ділянок сільськогосподарського призначення та населених пунктів.
8. Синтезатор індуктивних моделей з динамічною кількістю вхідних даних.
9. Первинна обробка даних в інформаційних системах прогнозування стану довкілля.
10. Система підтримки прийняття рішень при лікуванні захворювань шкіри.
11. Комп'ютерна система виділення меж у графічних зображеннях.
12. Система комп'ютерного моніторингу розвитку мислення дітей дошкільного віку на основі ігор.
13. Система підтримки прийняття рішень у підборі конфігурацій комп'ютерів.
14. Проектування інформаційної системи для прогнозування курсу акцій.
15. Середовище візуальної розробки баз знань діагностичних експертних систем.
16. Проектування інформаційної системи для прогнозування інвестиційних ризиків.
17. Оцінка ефективності застосування різних методів профілактики онкологічних захворювань засобами імітаційного моделювання.
18. Інформаційна система секціонування фотографічних зображень.
19. Інформаційна система моніторингу проектів.
20. Інформаційна система побудови оптимального транзитного шляху на залізниці.
21. Нейромережевий алгоритм розпізнавання номерного знака автомобіля.
22. Система прийняття рішень при фінансових операціях.
23. Інформаційна система моніторингу готелю.
24. Інформаційна система прогнозування банкрутства підприємства.
25. Інтелектуальна система логістики.
26. Інформаційна система моніторингу міграційного руху населення.
27. Система підтримки прийняття рішень з відбору кадрів для робіт підвищеної небезпеки.

- 28.Інтелектуальна система обробки та аналізу статистичної інформації про користувачів соціальних мереж.
- 29.Комп'ютерна навчаюча система пошуку оптимуму функції за допомогою генетичного алгоритму.
- 30.Підсистема забезпечення процесу селекції для генетичного алгоритму.
- 31.Система Web-підтримки інструментально-контролюючої програми Master of Logic.
- 32.Розпізнавання зашумлених двійкових даних з використанням нейронних мереж.
- 33.Розпізнавання зашумлених двійкових даних з використанням МГУА.
- 34.Розпізнавання зашумлених текстових даних з використанням нейромереж.
- 35.Інтелектуальна система підтримки прийняття рішення як backend мобільного додатку .
- 36.Автоматизована підсистема імітаційного моделювання для автоматизованої системи управління складами.
- 37.Адекватність і об'єктивність моделювання інформаційної системи управління.
- 38.Система збору і моніторингу інформації з віддалених інформаційних ресурсів.
- 39.Моделювання систем передачі інформації особливої важливості.
- 40.Витягання інформації з текстів природної мови для розробки онтологій.
- 41.Система морфемного аналізу словоформ української мови.
- 42.Система аналізу роботи підрозділів на основі бази знань.

ДОДАТОК М – Приклад вступу до дипломної роботи

Тема дипломної роботи:

СИСТЕМА ДЛЯ ЗАХИСТУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗА ДОПОМОГОЮ ТЕХНОЛОГІЇ ЦИФРОВИХ ВОДЯНИХ ЗНАКІВ

ВСТУП

Задача захисту інформації від несанкціонованого доступу вирішувалося у всі часи упродовж історії людства. Два основні напрями, існуючих з давніх часів: криптографія і стеганографія. Криптографічні методи дозволяють приховувати інформацію за допомогою шифрування, а стеганографічні методи приховують сам факт наявності прихованої інформації [1].

Темою дипломної роботи є розробка системи для захисту програмного забезпечення за допомогою технології цифрових водяних знаків.

Актуальність теми. Існує велика кількість різних методів, що дозволяють приховати факт наявності зашифрованої інформації. Класичними прикладами є крапчасті колоди, в якій карти мають бути розташовані в певному порядку для відновлення інформації. Також були поширені шифри, що використовують форму інформаційного носія. Інформація відновлювалася з використанням трафаретів або геометричних форм. З розвитком фотографії стала застосовуватися технологія мікроточок – маленьких знімків, що вставляються в текст листів [2].

Методи сучасної стеганографії можна розділити за наявності обробки цифрових сигналів. За відсутності такої обробки виникає питання надійності приховання даних в заголовках файлів або при передачі пакетів, адже такі дані порівняно легко виявити або знищити. Відповідно, на застосовність цих методів накладаються обмеження.

З іншого боку, відбувається все більше досліджень в області застосовності стеганографії при обробці цифрових сигналів, що дозволяє говорити про

цифрову стеганографію. Наприклад, при передачі голосового повідомлення з використанням IP- телефонії в деякі пакети можуть вбудовуватися секретні дані. Стандартний приймач розпізнає такий пакет як пошкоджений і проігнорує його. Але якщо той, що приймає, знає про наявність існування таких пакетів, то він може настроїти приймач так, щоб ці пакети не видалялися, а зберігалися, і робити розшифровку даних.

Іншим поширенішим прикладом є цифрові дані, що, як правило, мають аналогову природу, наприклад, аудіозаписи, зображення, відео. Відомі також приклади вбудови інформації в текстові файли і у виконувані файли програм.

Розробка ефективного методу захисту є одним з ключових завдань при розробці комерційного програмного забезпечення. Методи захисту програмного забезпечення дозволяють розробникам звести до мінімуму можливість нелегального використання своєї продукції.

Також актуальною проблемою є обмеження на використання шифрування в деяких країнах і відсутність єдиних стандартів в області захисту цифрових даних. Тому дослідження в цій області нині є актуальними. Одним з методів є використання цифрових водяних знаків [3].

Мета і задачі розробки. Метою цієї роботи є проектування і розробка додатку для захисту програмного забезпечення за допомогою технології цифрових водяних знаків.

Для досягнення поставленої мети необхідно розв'язати такі задачі:

- виконати аналіз предметної області;
- вивчити існуючі рішення і зробити постановку задачі;
- визначити функціональні вимоги до системи;
- спроектувати логічну і фізичну структуру системи;
- розробити додаток для захисту ПЗ;
- провести тестування системи.

Об'єкт і предмет розробки. Об'єктом розробки є криптографічні і стеганографічні методи захисту інформації від несанкціонованого доступу за допомогою шифрування і приховування інформації.

Предметом розробки є алгоритми і розробка системи для захисту програмного забезпечення за допомогою технології цифрових водяних знаків.

Практичне значення одержаних результатів. Практична значущість роботи полягає в створенні процедур і функцій для захисту програмного забезпечення за допомогою технології цифрових водяних знаків. Розроблену систему захисту ПЗ можна використовувати в багатьох сферах програмної інженерії, зокрема в сфері розробки програмного забезпечення.

Апробація роботи. Теоретичні і практичні результати, отримані у процесі розробки, доповідалися і обговорювалися на Першій міжнародній науково-технічній конференції «Комп'ютерні інформаційні системи і технології» (Львів, 23-24 квітня 2017 р.).

Публікації. За темою дипломної роботи опубліковано дві наукових роботи, у тому числі одна – у виданнях, рекомендованих МОН України. З них одна робота [4] опублікована у співавторстві.

Структура та обсяг роботи. Дипломна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел (25 найменувань), 5 додатків. Загальний обсяг роботи становить 78 сторінок основного тексту, 15 рисунків та 6 таблиць.

Продовження додатку М

Тема дипломної роботи:

ІНФОРМАЦІЙНО-АНАЛІТИЧНА СИСТЕМА ДЕРЖАВНОЇ ПОДАТКОВОЇ СЛУЖБИ

ВСТУП

Податкова служба – це сукупність державних органів, які організують і контролюють надходження податків, податкових і окремих видів неподаткових платежів. Темою дипломної роботи є розробка автоматизованої інформаційно-аналітичної системи державної податкової служби.

Актуальність теми. Податкова служба є головною організацією у зборі податків, реєстрації нових підприємців та занесення їх у головний реєстр, створення звітності тощо [1].

Система податкових органів – це розгалужена мережа спеціалізованих структур. Вони здійснюють контроль надходженням до бюджету податкових платежів. Податкова служба забезпечує надходження до бюджету всіх відрахувань, які згідно закону включені до податкової системи.

Надійність та ефективність податкової системи залежить від двох чинників: раціональності побудови податкової системи і рівня організації податкової служби держави [2].

При роботі податкової служби присутня велика кількість інформації, а саме: дані про клієнтів (фізичних та юридичних осіб), підрахунки збору для платника податку тощо. Для обробки даної інформації необхідна сконцентрованість та увага. При умові, що людський фактор також грає велику роль, можна зробити висновок, що створення інформаційно-аналітичної системи державної податкової служби є необхідністю в даний час.

Актуальність створення інформаційно-аналітичної системи державної податкової служби в окремих містах обумовлена сьогодні необхідністю використання великих і постійно зростаючих обсягів інформації при здійсненні функції з електронної обробки податкових документів [3].

Мета і задачі розробки. Метою роботи є проектування і розробка інформаційно-аналітичної системи, що дозволить автоматизовано рахувати заборгованості платників податків, створення звітів та збереження їх у зручному форматі.

Для досягнення цієї мети мають бути розв'язані такі задачі:

- виконати порівняльний аналіз існуючих аналогів;
- виконати огляд можливих засобів реалізації;
- вивчити існуючі рішення і зробити постановку задачі;
- визначити функціональні вимоги до системи;
- спроектувати логічну і фізичну структуру інформаційної системи;
- реалізувати і протестувати додаток.

Об'єктом розробки є інформаційна система податкової служби з функціями збереження цілісності даних.

Предметом розробки є метод та алгоритми програмного забезпечення для обрахунку сплати податку та захисту цілісності даних.

Практичне значення одержаних результатів. Практична значущість роботи полягає в розробці алгоритмів та їх програмної реалізації для збереження цілісності податкової інформації з функціями підвищеної стійкості для підрахунку податку.

Апробація роботи. Основні теоретичні і практичні результати, отримані в процесі розробки, доповідалися і обговорювалися на V науково-технічній конференції «Проблеми інформатизації» (Київ, 16-17 квітня 2017).

Публікації. По темі дипломної роботи опубліковано дві наукових роботи, у тому числі одна – у виданнях, рекомендованих МОН України. З них одна робота [4] опублікована у співавторстві.

Структура та обсяг роботи. Дипломна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел (20 найменувань), 5 додатків. Загальний обсяг роботи становить 76 сторінок основного тексту, 12 рисунків та 6 таблиць.

Продовження додатку М

Тема дипломної роботи:

ПОБУДОВА АРХІТЕКТУРИ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ ЗАДАЧІ ФІЛЬТРАЦІЇ МІСЬКИХ ШУМІВ У ЗВУКОВОМУ ПОТОЦІ

ВСТУП

Проблема сучасної обробки звуку тісно пов'язана з однією з проблем машинного навчання – проблемою розпізнавання мови. При розпізнаванні мови застосовуються ті ж механізми, що і при фільтрації звукового потоку. Завдання фільтрації шумів у процесі розпізнавання мови полягає в тому, щоб видалити з початкового сигналу всі звуки, що не належать до людської мови: порожні проміжки, проміжки між словами і небажані шуми.

При обробці цифрового звуку пристроєм, що здійснює фільтрацію звуку, є спеціалізована комп'ютерна програма, що реалізовує той або інший набір алгоритмів з обробки звуку.

Темою дипломної роботи є побудова архітектури нейронної мережі для задачі фільтрації міських шумів у звуковому потоці.

Актуальність теми. Проблема фільтрації звукового потоку є актуальною в багатьох сферах, що так чи інакше працюють зі звуком. Технології фільтрації і згладжування звуку активно використовуються в сучасній музичній індустрії, стільниковому радіозв'язку, медицині тощо.

Незважаючи на велику кількість існуючих алгоритмів для фільтрації шумів у звуковому сигналі [1], ні для одного з них не визначений критерій якості підсумкового вихідного звукового сигналу. Сучасні методи фільтрації звуку, що ґрунтуються на методі машинного навчання, використовують принципово новий алгоритм фільтрації шумів, який називається довірчим фільтром.

Довірчий фільтр забезпечує мінімально можливий довірчий інтервал для кожного початкового звукового кадру з вхідного звукового сигналу. Перевагою

роботи довірчого фільтру є те, що він самостійно налаштовується в кожному конкретному випадку і видає найкраще максимально можливе значення у процесі фільтрації звуку.

В основі алгоритму довірчого фільтру лежить метод найменших квадратів [2], який ґрунтується на апроксимації послідовності початкових даних лінійною комбінацією деяких функцій. Сума квадратів відхилень початкових даних від апроксимованих значень при цьому мінімізується. В якості апроксимуючих функцій зазвичай використовуються деякі поліноміальні функції [3].

Одним із способів апроксимації функцій з використанням нейронних мереж є здатність апроксимувати складні нелінійні функції. Нейронна мережа з будь-якою нелінійністю є універсальною апроксимацією, що доводить узагальнена теорема апроксимації [4].

Таким чином, задача побудови нейронної мережі для фільтрації шумів у звуковому потоці є актуальною.

Мета і задачі розробки. Метою цієї роботи є проектування архітектури нейронної мережі і розробка додатку для фільтрації звуку на основі спроектованої нейронної мережі.

Для досягнення поставленої мети необхідно розв'язати такі задачі:

- виконати аналіз предметної області;
- провести порівняльний аналіз існуючих аналогів, виявити їх переваги і недоліки;
- спроектувати архітектуру нейронної мережі;
- спроектувати додаток з фільтрації звуку;
- розробити додаток з фільтрації звуку;
- розробити і провести комплексне тестування системи.

Об'єкт і предмет розробки. Об'єктом розробки є нейронна мережа для фільтрації шумів у звуковому потоці зі здатністю апроксимувати складні нелінійні функції.

Предметом розробки є алгоритми і програмне забезпечення для фільтрації оцифрованого звукового сигналу на основі спроектованої нейронної мережі.

Практичне значення одержаних результатів. Практична значущість роботи полягає у створенні процедур і функцій для фільтрації і згладжування звуку на основі спроектованої нейронної мережі. Розроблену програму фільтрації звукового потоку можна використовувати у багатьох сферах, які так чи інакше працюють зі звуком.

Апробація роботи. Теоретичні і практичні результати, отримані в процесі розробки, доповідалися і обговорювалися на IV Міжнародній конференції з практичного застосування штучного інтелекту і DATA SCIENCE (Харків, 23-24 вересня 2017 р.).

Публікації. За темою дипломної роботи опубліковано дві наукових роботи, у тому числі одна – у виданнях, рекомендованих МОН України. З них одна робота [8] опублікована у співавторстві.

Структура та обсяг роботи. Дипломна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел (23 найменування), 6 додатків. Загальний обсяг роботи становить 75 сторінок основного тексту, 15 рисунків та 8 таблиць.

Навчально-методичне видання

Авраменко Валентин Семенович

Беседіна Світлана Валеріївна

ВИКОНАННЯ ТА ОФОРМЛЕННЯ ВИПУСКНИХ РОБІТ

Освітній ступінь «Бакалавр»

Навчально-методичний посібник

Комп'ютерне верстання:

А. С. Авраменко

Підписано до друку __.__.2018. Формат 60x84/16.
Ум. друк. арк. 7,96. Тираж 300 пр. Зам. № __

Видавець: Чабаненко Ю.А.

Свідоцтво про внесення до Державного реєстру видавців
серія ДК №1898 від 11.08.2004 р.

Україна, м. Черкаси, вул. Дашкевича, 39

Тел.: (0472) 56-46-66, (093) 788-99-99

e-mail: office@2upost.com

Друк ФОП Нечитайло О.Ф.

Україна, м. Черкаси, вул. Дашкевича, 39

Тел.: (096) 3-306-306, e-mail: print@306.com