

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРКАСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені БОГДАНА ХМЕЛЬНИЦЬКОГО**

**Факультет обчислювальної техніки,
інтелектуальних та управляючих систем**

**Кафедра програмного забезпечення
автоматизованих систем**

Ю. Є. Гребенович, Б. О. Онищенко, О. О. Супруненко

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ

**ДО ВИКОНАННЯ ТА ОФОРМЛЕННЯ
КУРСОВИХ РОБІТ**

з дисциплін

***“Основи програмування”,
“Програмування та алгоритмічні мови”,
“Алгоритмізація та програмування”***

***для студентів напрямів підготовки
050101 – „Комп’ютерні науки”,
050103 – „Програмна інженерія”,
040303 – „Системний аналіз”
усіх форм навчання***

Черкаси 2015

УДК 004.42 + 004.021
ББК 32.97

Гребенович Ю.Є., Онищенко Б.О., Супруненко О.О. Методичні рекомендації до виконання та оформлення курсових робіт з дисциплін «Основи програмування», «Програмування та алгоритмічні мови», «Алгоритмізація та програмування» для студентів напрямів підготовки 050101 – «Комп'ютерні науки», 050103 – «Програмна інженерія» та 040303 – «Системний аналіз» усіх форм навчання. – Черкаси: Вид. від. ЧНУ імені Богдана Хмельницького, 2015. – 33 с.

Рецензенти:

М.М. Орлова, кандидат технічних наук, доцент кафедри системного програмування та спеціалізованих комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут»,

В.С. Авраменко, кандидат фізико-математичних наук, завідувач кафедри інформаційних систем та медичних технологій Черкаського національного університету імені Богдана Хмельницького.

Профільна кафедра: кафедра системного програмування та спеціалізованих комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут»;

Затверджено на засіданні кафедри програмного забезпечення автоматизованих систем, як методичний посібник для організації роботи студентів над завданнями першої курсової роботи студентів денної, заочної та екстернатної форм навчання, які проходять підготовку за напрямками «Комп'ютерні науки», «Програмна інженерія» та «Системний аналіз», протокол № 12 від 20.04.2015.

УДК 004.42 + 004.021

ББК 32.97

Рекомендовано до друку Вченою радою Черкаського національного університету імені Богдана Хмельницького
(протокол № 9 від «26» травня 2015 р.)

ЗМІСТ

Вступ	4
1. Індивідуальне завдання на курсову роботу	4
2. Етапи виконання курсової роботи.....	5
3. Структура та зміст пояснювальної записки до курсової роботи	6
3.1. Загальні рекомендації до оформлення розділів	8
3.2. Рекомендації до змісту першого розділу курсової роботи.....	12
3.3. Приклади оформлення інформаційних джерел	13
3.4. Побудова блок-схем алгоритмів та структурної схеми програми (до другого розділу курсової роботи).....	15
3.5. Приклади оформлення третього розділу з реалізації програмного продукту	18
3.6. Порядок підготовки матеріалів у курсовій роботі.....	22
4. Написання доповіді та створення презентації до захисту курсної роботи	23
5. Порядок рецензування курсової роботи	25
6. Перелік типових тем курсових робіт	25
Список рекомендованої літератури.....	27
Internet-посилання	28
Додаток А. Індивідуальне завдання на курсову роботу.....	30
Додаток Б. Титульний лист курсової роботи.....	31
Додаток В. Приклад відгуку керівника на курсову роботу	32

ВСТУП

Виконання програмних проектів у фаховій підготовці інженерів-програмістів займає центральне місце. **Курсова робота** є *видом самостійної роботи студента*, що спрямована на набуття практичних навичок по підборі алгоритмів реалізації поставленого завдання і створенню на їх основі програмних продуктів. Вона передбачає розвиток творчого мислення та навичок кодування для реалізації практичних задач на основі теорії алгоритмів, з використанням сучасних засобів розробки програмного забезпечення. Написання курсової роботи дозволяє розширити об'єм знань студентів в галузі програмування і створити реальну основу використання своїх знань для вирішення на ЕОМ задач з інших дисциплін, а також використати набуті навички у своїй подальшій практичній діяльності. Курсова робота оформлюється у вигляді пояснювальної записки та оптичного диску з кодом програмного продукту.

Таким чином, **метою виконання курсової роботи** є набуття практичних навичок з *початкової обробки завдання* (1), *проектування* (2) та *реалізації* (3) програмного проекту, що передбачений темою курсової роботи, на мові програмування високого рівня. За результатами проекту створюється технічна документація, якою в курсовій роботі є **пояснювальна записка**. Дані методичні вказівки призначені для допомоги студентам у виконанні всіх цих завдань.

1. Індивідуальне завдання на курсову роботу

Підставою для виконання курсової роботи є «**Індивідуальне завдання на курсову роботу**» (додаток А). Воно містить тему курсової роботи, вимоги до програмного компоненту (функціонал), вимоги до інтерфейсу, перелік розділів пояснювальної записки до курсової роботи, перелік наочних матеріалів та календарний план виконання задач курсового проектування. Індивідуальне завдання видається студенту керівником. *Перед початком роботи керівником і*

студентом обговорюються всі необхідні вимоги до проекту та зовнішні представлення компонентів програмного продукту.

2. Етапи виконання курсової роботи

Робота виконується за чітко визначеними етапами:

- 1) студент *отримує тему та індивідуальне завдання* до курсової роботи і *обговорює завдання* з керівником курсової роботи;
- 2) проводиться огляд технічної інформації за темою роботи, це може бути інформація з Internet-сайтів, фахових журналів та книг (див. список літератури), аналогічні програмні продукти в демоверсії чи доступних версіях; у цих джерелах *знаходяться алгоритми для реалізації поставленого завдання та приклади вдалих і невдалих реалізацій програм*, подібних до отриманого завдання;
- 3) виконується *аналіз розглянутої технічної інформації*, *обираються алгоритми* та програмні рішення для реалізації отриманого завдання;
- 4) розробляється структурна чи функціональна схема програмного продукту, описуються її елементи;
- 5) розробляються блок-схеми роботи алгоритмів основних елементів програмного продукту;
- 6) розробляється інтерфейс програмного продукту;
- 7) проводиться *кодування та тестування* програмного продукту;
- 8) по вище виконаним завданням *оформлюється пояснювальна записка*;
- 9) для написання відгуку *програма і пояснювальна записка представляються керівнику курсової роботи*;
- 10) одночасно з попереднім пунктом *готується доповідь та створюється презентація* до захисту курсової роботи;
- 11) проводиться *корегування роботи* (доповіді і презентації) за зауваженнями керівника (*щоб зауважень було мінімум, потрібно всі незрозумілі питання по ходу виконання роботи вирішувати з керівником!*);
- 12) захист курсової роботи.

В ході виконання курсової роботи студенти мають навчитися 1) виконувати огляд алгоритмів та методів реалізації поставленого завдання, 2) обирати вдалі проектні рішення, проводити проектування невеликих програмних продуктів, 3) створювати за завданнями програмні продукти з використанням мов програмування високого рівня, 4) грамотно оформлювати та представляти отримані результати.

3. Структура та зміст пояснювальної записки до курсової роботи

Пояснювальна записка до курсової роботи *оформлюється* на листах формату А4, формується у текстовому редакторі Word (14 кегль, полуторний інтервал, всі поля по 20 мм). Титульний лист оформлюється за зразком, наведеним у додатку Б.

Обсяг пояснювальної записки обумовлюється її змістом і орієнтовно складає 18-25 сторінок друкованого тексту (без врахування додатків) з формулами, малюнками і таблицями, що займають менше 1-ї сторінки (всі інші малюнки та таблиці виносяться у додатки). Структура та зміст курсової роботи оформлюється у наступній послідовності:

- 1) **вступ** (1-2 сторінки), в якому обґрунтовується актуальність теми, визначається мета та завдання роботи, наводяться базові поняття, що стосуються обраної теми;
- 2) **розділ 1*** – **огляд інформаційних джерел** (5-7 сторінок), в якому проводиться огляд алгоритмів та методів реалізації отриманого завдання, проводиться їх аналіз та вибір конкретних алгоритмів та методів реалізації програми, робляться висновки в яких формулюється ідея програмного проекту (див. п. 3.1);

* При оформленні розділів **назва кожного розділу формується студентом у залежності від викладеного матеріалу** (!!! не допускається такі назви розділів, як теоретична частина, чи практична частина !!!).

- 3) **розділ 2*** – **проектування програмного продукту** (5-7 сторінок), в якому проводиться опис алгоритмів для реалізації задачі, розробляється блок-схема основного та допоміжних алгоритмів; розробляється узагальнена структурна чи функціональна схема програмного продукту; робиться висновок, в якому ставиться задача на розробку програмного продукту;
- 4) **розділ 3*** – **реалізація програмного продукту** (5-7 сторінок): проводиться опис реалізації завдання у програмному коді, який може доповнюватися схемами, діаграмами та невеликими ділянками лістингу програми з необхідними коментарями (на всі ілюстративні матеріали мають бути посилання в основному тексті, наприклад: див. рис. 4, де зображено блок-схеми основних алгоритмічних конструкцій); опис результатів роботи програми та тестування доповнюється копіями екрану; у висновку описується реалізований функціонал та можливості щодо його застосування;
- 5) **висновки** (1 сторінка) – у висновках висвітлюється функціонал програмного продукту, який був створений для реалізації поставленого завдання; коротко описується суть програмних рішень, що були застосовані; наводяться сфера застосування та шляхи розвитку запропонованих програмних рішень;
- 6) **список використаних джерел** (1 сторінка) – список літературних та Internet-джерел (при посиланні на Internet-джерела вказується назва джерела, його адреса та дата перегляду матеріалу (див. пункт 3.3)).
- 7) **додатки** – за необхідності громіздкі малюнки, таблиці, граф-схеми алгоритмів та інше, що *займають одну цілу сторінку формату А4 чи більше* та не несуть інформації, пов'язаної безпосередньо з темою роботи, виносяться у додатки; кожен додаток починається з нової сторінки, нумерується великими українськими літерами (за винятком букв *Г, Є, З, І, Ї, Й, О, Ч, Б*) і має назву в залежності від вмісту (див. додатки А-В).

* При оформленні розділів **назва кожного розділу формується студентом у залежності від викладеного матеріалу** (!!! не допускається такі назви розділів, як теоретична частина, чи практична частина !!!).

Наприклад: Зміст курсової роботи:

Вступ.....	2
Розділ 1. Огляд алгоритмів роботи з циклічними списками та сортування великих обсягів вхідних даних.....	3
1.1. Алгоритми роботи з циклічними списками.....	4
1.2. Аналіз алгоритмів для реалізації задачі складання поліномів.....	7
1.4. Висновок до першого розділу	8
Розділ 2. Проектування модуля для операцій над поліномами	9
2.1. Блок-схема алгоритму розщеплення циклічного списку	11
2.2. Блок-схема алгоритму множення поліномів.....	14
2.3. Структура модуля для операцій над поліномами.....	16
2.4. Висновок до другого розділу	18
Розділ 3. Реалізація та тестування модуля для роботи з поліномами	19
3.1. Реалізація основних функцій програмного модуля.....	19
3.2. Реалізація користувацького інтерфейсу.....	21
3.3. Тестування програмного модуля для операцій над поліномами	24
3.4. Висновок до третього розділу	26
Висновки.....	27
Список інформаційних джерел	28
Додаток А. Блок-схема модуля перетворення поліномів	29
Додаток Б. Узагальнена структурна схема програмного модуля для операцій над поліномами	30

3.1. Загальні рекомендації до оформлення розділів

Кожен *розділ* роботи починається з окремого листа. Опис завдань проводиться у текстовому редакторі Word, 14 кегль, полуторний інтервал, всі поля по 20 мм. Заголовки розділів набираються 16 кеглем напівжирним шрифтом, інші заголовки – 14 кеглем напівжирним шрифтом. Відступ перед і після заголовку – 12 пт.

Формули набираються у редакторі формул Microsoft Equation і нумеруються – нумерація в пояснювальній записці наскрізна. Формули розміщуються по центру листа, мають відступ 3 інтервали (1 пустий рядок з полуторним інтервалом) до та після формули, номер формули ставиться у дужках по лівому краю листа (рис. 1).

Рисунки виконуються у окремій області (наприклад: «Рисунок Word» чи в окремій області малюнка), можуть виконуватися у інших додатках типу Microsoft Visio, CorelDraw і т.і.



Рис.1. Правила оформлення формул у тексті курсової роботи [14].

Рисунки нумеруються та підписуються (нумерація наскрізна у межах розділу, наприклад: рис. 2.3 – це 3-й малюнок 2-го розділу), у тексті документу мають бути посилання на кожний виконаний в роботі рисунок. Після підпису рисунка залишається простір (3 інтервали чи 1 пустий рядок з полуторним інтервалом) до наступного за ним тексту (рис. 2). Якщо рисунок не поміщається на тій сторінці, на якій на нього посилаються, його розміщують на початку наступної сторінки, заповнюючи залишений простір текстом з наступної сторінки.

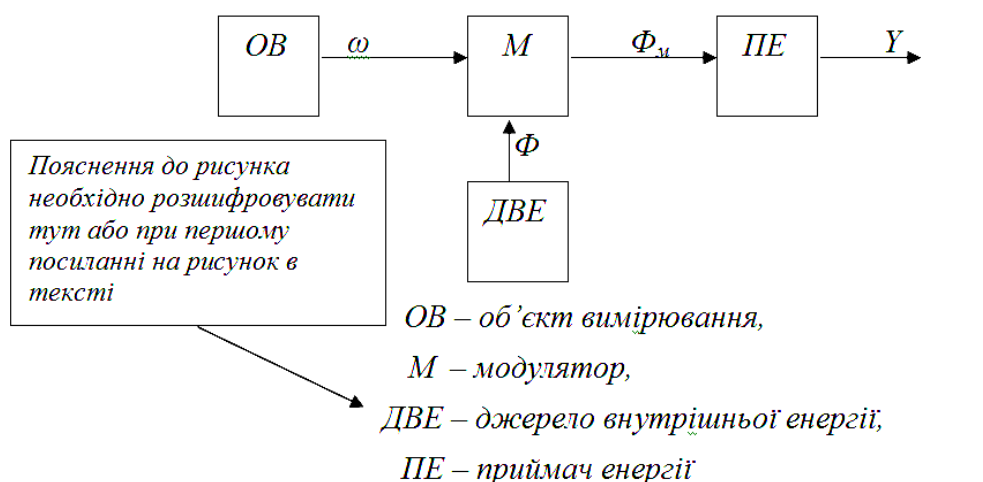


Рис 3. Приклад оформлення рисунка

3 інтервали

Рис.2. Правила оформлення рисунків у тексті курсової роботи [14].

Таблиці розміщуються по центру листа симетрично до тексту після першого посилання на даній сторінці або на наступній, якщо на даній вона не поміщається і таким чином, щоб зручно було її розглядати без повороту, або з поворотом на кут 90° за годинниковою стрілкою [14]. Номер таблиці пишеться з вирівнюванням по правому краю, назва таблиці – по центру на наступному рядку. Наприклад:

Таблиця 1.4

Елементи граф-схеми алгоритму та їх характеристики.

--	--	--	--	--

У верхній (чи лівій) частині таблиці розміщують заголовок таблиці, в яком вказують назви граф. Діагональний поділ комірок таблиці не допускається. Назви граф пишуть малими буквами, починаючи перше слово графи з великої літери, у підграфах назви повністю пишуть малими літерами, в кінці крапку не ставлять. Текст в таблицях пишуть з *одинарним інтервалом*.

Якщо всі параметри величин, які наведені в таблиці, мають одну й ту ж одиницю фізичної величини, то наприкінці назви таблиці після коми розміщують її скорочене позначення, наприклад *мм*. Якщо ж параметри мають різні одиниці фізичних величин, то позначення одиниць записують в заголовках граф після коми (наприклад: *Довжина, мм*).

Дані, що наводяться в таблиці, можуть бути словесними і числовими. Слова записують в графах з однієї позиції. Якщо рядки таблиці не розділені лініями, то текст, який повторюється і складається з одного слова дозволяється замінювати лапками (,). Якщо текст складається з одного і більше слів, то при першому повторенні його замінюють словами “*те ж*”, а далі лапками. При розділенні таблиці горизонтальними лініями – ніякої заміни не виконують.

Числа записують посередині графи ***так, щоб їх однакові розряди по всій графі були точно один під одним***, за винятком випадку, коли вказують інтервал. Інтервал вказують від меншого числа до більшого з тире між ними: *12 – 35 або 122 – 450*. Дробові числа наводять у вигляді десяткових дробів, з однаковою кількістю знаків після коми в одній графі. Ставити крапки замість цифр чи математичних символів, які повторюються, не можна. Якщо цифрові чи інші дані в таблиці не наводяться, то ставиться прочерк [14].

Якщо таблиця не поміщається на одній сторінці, її переносять на наступну, на якій повторюють заголовок таблиці, або під заголовком на початку таблиці нумерують рядки і цю цифрову нумерацію рядків розміщують в заголовку продовженої таблиці (рис. 3). Надписують продовження таблиці: «Продовження таблиці 1.4», вирівнюють по правому краю листа.

Назва таблиці.

Таблиця 1.4

Найменування елемента	Позначення елемента	Вхідні зв'язки	Вихідні зв'язки	Примітки
1	2	3	4	5

<на наступній сторінці>

Продовження таблиці 1.4

1	2	3	4	5

Рис. 3. Оформлення перенесення таблиць

У розділі роботи, який стосується реалізації програмного продукту, описуючи особливості реалізації алгоритму можна наводити **фрагменти коду**, які доповнюються необхідними коментарями. Якщо є необхідність навести частину лістингу, що перевищує розмір сторінки, такі ділянки коду розміщуються у додатках, на кожен з яких мають бути посилання в тексті роботи, де мова йде про елементи цих лістингів. Посилання наводяться у дужках після відповідного пояснення у такій формі: (див. додаток В).

При виконанні курсової роботи студент користується індивідуальним завданням, інформаційними джерелами, рекомендованими керівником курсової роботи, конспектом лекцій з дисципліни «Основи програмування», «Програмування та алгоритмічні мови», «Алгоритмізація та програмування», рекомендованою літературою (див. п. «Список рекомендованої літератури») та Internet-посиланнями (див. п. «Internet-посилання»). При використанні літературних та Internet-ресурсів обов'язково в тексті роботи наводяться посилання, наприклад: [4, с.36-37]. У даному посиланні використовується 4-е джерело, в якому наведені факти розміщуються на сторінках 36-37. У списку використаних джерел книги та Internet-ресурси нумеруються в порядку їх використання в тексті курсової роботи.

Виконана робота роздруковується, підшивається і захищається студентом на засіданні комісії з захисту курсової роботи за розкладом, визначеним кафедрою.

Оцінка за курсову роботу приєднується до загального рейтингу оцінок студента, який враховується при видачі диплому бакалавра, також є складовою рейтингу при вступі на навчання до магістратури.

3.2. Рекомендації до змісту першого розділу курсової роботи

Перший розділ курсової роботи починається з формулювання *мети* та *задач** курсової роботи



Далі проводиться *огляд інформаційних джерел* чітко у відповідності з метою та сформульованими задачами. У висновках до розділу обираються алгоритми для реалізації завдання курсової роботи. Розберемо все послідовно.

При *формулюванні мети* та *постановці задач* слід керуватися визначеннями:

Мета роботи – це кінцевий результат, на досягнення якого вона спрямована. Мета має адекватно відображатись у темі роботи, містити в узагальненому вигляді очікувані результати.

Задачі підпорядковуються меті роботи і спрямовані на послідовне (поетапне) її досягнення.

Приклад формулювання мети курсової роботи:

Метою даної курсової роботи є обрання алгоритмів та реалізація програмного модуля для операцій над поліномами.

Приклад постановки задач курсової роботи:

1. Виконати аналіз алгоритмів для операцій над поліномами.
2. Побудувати блок-схеми алгоритмів складання та множення поліномів.
3. Скласти структурну схему програмного модуля для операцій над поліномами.

4. Реалізувати у вигляді програмного продукту модуль для операцій над поліномами.
5. Зробити висновки щодо можливостей застосування розробленого програмного модуля для операцій над поліномами.

Під час огляду літературних та Internet-джерел студенту рекомендується:



- 1) переглянути обрані за списком рекомендованої літератури та за Internet-пошуком джерела, які містять інформацію про розв'язання поставлених задач або про певний алгоритм чи метод,
- 2) переглянувши обраний матеріал, виділити джерела, у яких найбільш повно розкриті питання, що стосуються завдання курсової роботи,
- 3) за обраними джерелами провести опис (своїми словами) основного теоретичного матеріалу з посиланнями на оброблені джерела (виклад матеріалу студент має робити таким чином, щоб була зрозуміла його думка щодо викладеного матеріалу), можна використовувати схеми і малюнки з оглянутих інформаційних джерел, але обов'язково треба зробити посилання на ці джерела після назви рисунка чи схеми у формі: [5, с.304] (рисунок з 5-го джерела, зі сторінки 304), подібно до того, як це зроблено на стор. 9, рис 2;
- 4) по закінченні 1-го розділу потрібно зробити короткий висновок про доцільність обрання певних алгоритмів та методів для розв'язання поставленого завдання;
- 5) короткий висновок до розділу можна використовувати при написанні загального висновку до курсової роботи.

3.3. Приклади оформлення інформаційних джерел

Список інформаційних джерел (літературних та Internet-джерел) складають за такими правилами:

1) джерело (книгу, журнал, електронний ресурс) наводять у списку мовою, якою Ви його читали (українською, англійською чи ін.),

2) на початку опису джерела вказують основного автора, його ініціали, назву джерела,

3) в залежності від того, наводиться друковане видання чи електронне джерело, вказується його походження, відповідно [Текст]/[Електронний ресурс],

4) після похилої риски вказують перелік інших авторів, якщо такі є,

5) вказують місто, видавництво, рік видання та загальну кількість сторінок (для книг); назву журналу (після //), номер, рік видання та початкову-кінцеву сторінки, на яких розміщено матеріал (для статей з журналів), дату, коли матеріал опрацьовувався (для електронних ресурсів).

При оформленні списку друкованих інформаційних джерел повну кількість сторінок вказують за зразком: 352 с. При вказівці вибраних сторінок журналу, на яких розміщується стаття вказівку оформлюють за зразком: С. 45-52.



Наприклад, 1) так оформлюється у списку літератури книга трьох авторів:

1. Ахо А. Структуры данных и алгоритмы [Текст] / А.Ахо, Дж. Хопкрофт, Дж. Ульман. – М.: Издательский дом «Вильямс», 2010. – 384 с.

2) так оформлюється у списку літератури стаття одного та двох авторів:

2. Драпак Л.С. Використання мови C# при проектуванні людино-машинного інтерфейсу [Текст] / Л.С. Драпак // Восточно-европейский журнал передовых технологий. – 2012. – № 4/2 (58). – С. 39-42.

3. Марковский А.П. Метод доопределения частично-заданных булевых функций для обеспечения их лавинных свойств. [Текст] / А.П. Марковский, Н.Н. Романец // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. наук. пр. – К.: Век+, – 2009. – № 51. – С. 56-62.

При оформленні Internet-джерел важливо зафіксувати назву сторінки ресурсу, автора, якщо він вказаний, та електронну адресу сторінки ресурсу. Формувати у списку літератури посилання на Internet-джерел можна за прикладом:



4. Гавва А. «Адское» программирование. Ada-95. Компилятор GNAT. [Електронний документ]. Режим доступу: <http://ada-ru.org/V-0.4w/index.html>. Перевірено: 18.07.2012.

Дату перегляду Internet-джерела важливо вказувати, оскільки Internet-сторінки можуть оновлюватися, і потрібно знати, коли ви останній раз її переглядали. При посиланні на іноземні Internet-джерела можна керуватися одним з наступних прикладів:

5. Ada Reference Manual ISO/IEC 8652:2007(E) Ed. 3. [Електронний документ]. Режим доступу: <http://ocw.unican.es/ensenanzas-tecnicas/programacionen-lenguaje-ada/otros-recursos-2/RM-Final.pdf>. Перевірено: 18.07.2012.

6. Ada Reference Manual ISO/IEC 8652:2007(E) Ed. 3. Available at: <http://www.adapower.com/rm95/index.html> (accessed 18 July 2012).

3.4. Побудова блок-схем алгоритмів та структурної схеми програми (до другого розділу курсової роботи)

Графічні засоби відображення структури та функцій програмного продукту служать для представлення і аналізу проектних рішень і є вдалим наочним засобом для командної роботи над програмним проектом.

Блок-схема (англ. block scheme) – графічне представлення задачі для її аналізу або розв'язування за допомогою спеціальних символів [16]. У таблиці 1 представлені елементи блок-схеми алгоритмів, які відповідають окремим елементам алгоритму.

Приклади представлення основних алгоритмічних конструкцій блок-схемами наведені на рис.4. Серед них:

- лінійна ділянка алгоритму (рис. 4, а),
- цикл з передумовою (цикл типу while) (рис. 4, б),

Таблиця 1.

Основні елементи блок-схеми алгоритму [16]

Назва 1	Позначення 2	Функція 3
Терміна- тор		Елемент відображає вхід із зовнішнього середовища або вихід з неї (найчастіше застосування - початок і кінець програми). Всередині фігури записується відповідна дія.
Процес		Виконання однієї або кількох операцій, обробка даних будь-якого виду (зміна значення даних, форми подання, розташування). Всередині фігури записують безпосередньо самі операції.
Рішення		Показує рішення або функцію перемикального типу з одним входом і двома або більше альтернативними виходами, з яких тільки один може бути обраний після обчислення умов, визначених всередині цього елемента. Вхід в елемент позначається лінією, що входить зазвичай у верхню вершину елемента. Якщо виходів два чи три то зазвичай кожен вихід позначається лінією, що виходить з решти вершин (бічних і нижній). Якщо виходів більше трьох, то їх слід показувати однією лінією, що виходить з вершини (частіше нижній) елемента, яка потім розгалужується.
Зумовле- ний процес		Символ відображає виконання процесу, що складається з однієї або кількох операцій, що визначені в іншому місці програми (у підпрограмі, модулі). Всередині символу записується назва процесу і передані в нього дані.
Дані		Перетворення у форму, придатну для обробки (введення) або відображення результатів обробки (виведення). Цей символ не визначає носія даних (для вказівки типу носія даних використовуються специфічні символи).
Межа циклу		Символ складається з двох частин - відповідно, початок і кінець циклу - операції, що виконуються всередині циклу, розміщуються між ними. Умови циклу і збільшення записуються всередині символу початку або кінця циклу - в залежності від типу організації циклу. Часто для зображення на блок-схемі циклу замість цього символу використовують символ рішення, вказуючи в ньому умову, а одну з ліній виходу замикають вище в блок-схемі (перед операціями циклу).
З'єднувач		Символ відображає вихід в частину схеми і вхід з іншої частини цієї схеми. Використовується для обриву лінії та продовження її в іншому місці (приклад: поділ блок-схеми, що не поміщається на листі). Відповідні сполучні символи повинні мати одне (при тому унікальне) позначення.
Коментар		Використовується для детальнішої інформації про кроки, процесу або групи процесів. Опис поміщається з боку квадратної дужки і охоплюється нею по всій висоті. Пунктирна лінія йде до описуваного елемента, або групи елементів (при цьому група виділяється замкнутою пунктирною лінією). Також символ коментаря слід використовувати в тих випадках, коли обсяг тексту в будь-якому іншому символі (наприклад, символ процесу, символ даних та ін) перевищує його обсяг.

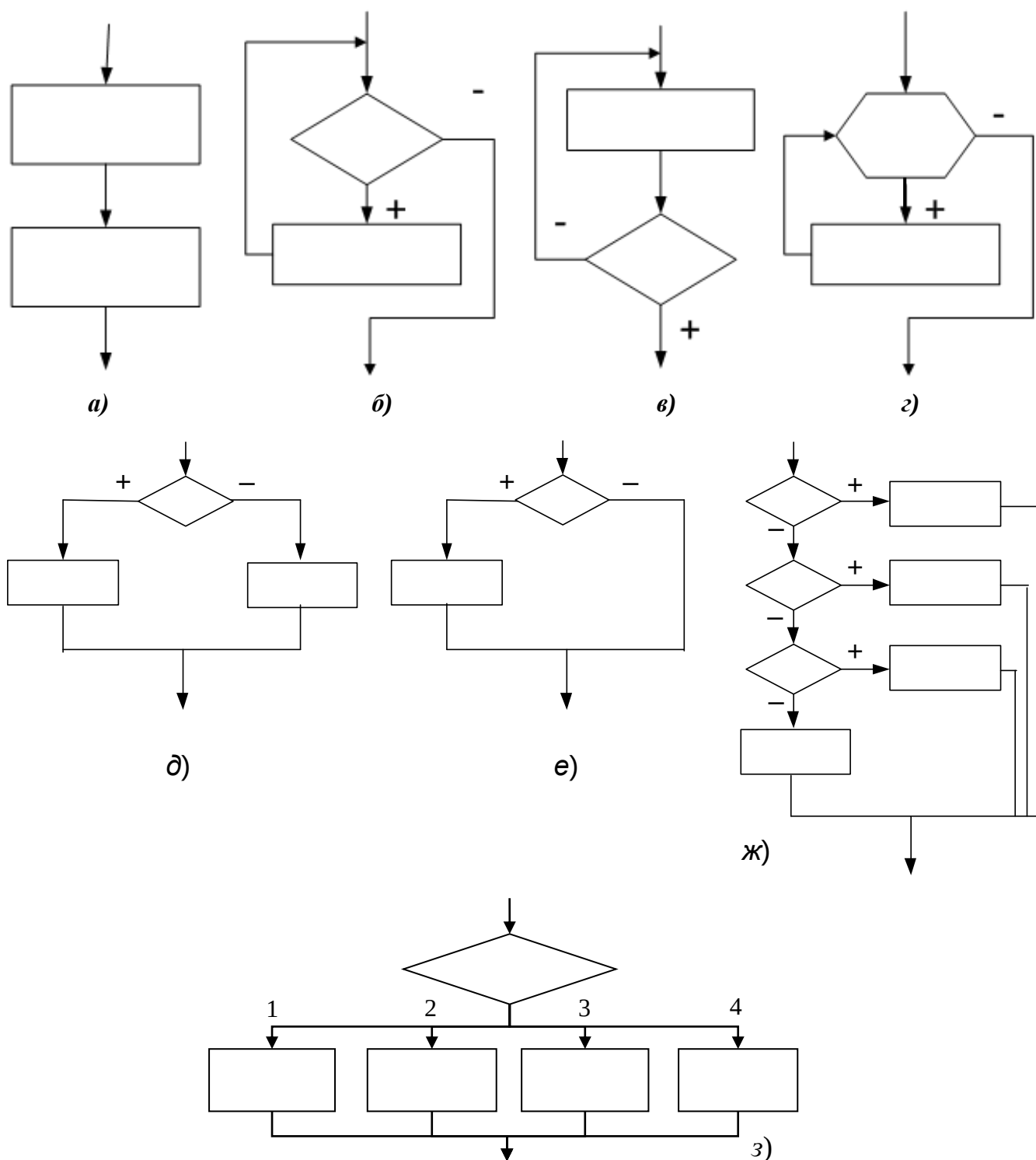


Рис. 4. Основні алгоритмічні конструкції, представлені блок-схемами:
a – лінійна ділянка алгоритму, *б* – цикл з передумовою, *в* – цикл з післяумовою,
г – цикл з параметром, *д* – розгалуження (повна форма), *е* – розгалуження (неповна
 форма), *ж* – каскадне розгалуження, *з* – оператор вибору [17].

- цикл з післяумовою (цикл типу do) (рис. 4, *в*),
- цикл з параметром (цикл типу for) (рис. 4, *г*),

- розгалуження у повній формі (розгалуження if) (рис. 4, д),
- розгалуження у неповній формі (розгалуження if-else) (рис. 4, е),
- множинне розгалуження (розгалуження if-elif-else) (рис. 4, ж),
- оператор вибору (case of або switch()) (рис. 4, з).

При виконанні курсової роботи студент має опрацювати алгоритми для реалізації завдання [2-12, Internet-джерела 1,4-5] та побудувати алгоритм (чи алгоритми) для їх подальшої практичної реалізації у вигляді програмного коду.

Проектування прикладної програми починається з **аналізу вимог**, яким вона повинна задовольняти. Такий аналіз проводиться з метою визначення призначення та умов експлуатації системи настільки, щоб скласти її початковий проект.

3.5. Приклади оформлення третього розділу з реалізації програмного продукту

В пункті „ОПИС ИНТЕРФЕЙСУ” описується процес роботи з програмним продуктом. Наводяться всі реалізовані меню та підменю, та, в разі необхідності, пояснення до них. Цей розділ бажано доповнити роздруківкою відповідних екранних форм, що реалізовані в програмі чи вікон консолей.

Тексти вихідних кодів програмного продукту повинні виконуватись з урахуванням угод що до стилю оформлення тексту програми та містити необхідні коментарі, наприклад для методів – призначення методу, опис параметрів та значення, що повертається.

Якщо студент вибрав тему курсового проекту самостійно, то вимоги до програми обговорюються з керівником та можуть відрізнятися від наведених вище.

Приклад опису основних функцій програмного модуля гри “Сапер”

Для того, щоб реалізувати спроектовані алгоритми, програма повинна мати чітку і прозору архітектуру. Щоб реалізувати таку архітектуру, в якій всі функції чітко розділені між класами, доцільно використати ідею про розділення логіки та інтерфейсу програми.

В першому наближенні ми повинні мати певну структуру для представлення мінного поля в пам’яті комп’ютера. Така структура повинна

бути гнучкою і логічною, щоб з нею легко можна було зв'язати функції користувацького інтерфейсу.

Першим типом, що створений для даної програми є злічений тип `CellState`, що реалізує множину можливих станів комірки: закрита, відкрита і маркована комірка.

Для реалізації абстракції комірки використовується тип `Cell`, що містить інформацію про кількість замінованих сусідів, наявність міни а також поточний стан комірки у вигляді поля типу `CellState`.

Нарешті, для реалізації абстракції мінного поля, клас головної форми містить поле, яке являється двовимірним динамічним масивом елементів типу `Cell`.

Уся логіка програми базується на тому, що користувач, здійснюючи хід, змінює внутрішній стан мінного поля. Після кожного ходу відбувається перемальовування карти з новими даними про відкриті і закриті комірки. Увесь вивід мінного поля відбувається на компоненті малювання `GameView`, що використовує функціонал `OpenGL`.

Особливим є перший хід. Мінне поле з заданими розмірами створюється під час натиснення кнопки «Нова гра» (після чого викликається метод `NewGame` головної форми, який і ініціалізує мінне поле). Після здійснення першого кроку, коли стають відомі координати за якими був здійснений перший крок, викликається алгоритм розстановки мін, що реалізований в методі `FirstMove` головної форми. І лише після цього виконується обробка першого ходу згідно стандартних правил гри.

Реалізація різноманітних ігрових функцій описана в обробнику подій `GameView_MouseClick` головної форми. За необхідності даний метод викликає методи `FirstMove`, `GameOver`, `BFS` (від англ. `breadth-firstsearch`– пошук у ширину).

Метод `BFS` реалізує алгоритм розкриття вільної області за допомогою алгоритму на графах пошуку в ширину. Метод `GameOver` реалізує функції підрахунку і збереження рекордів, підриву усіх мін на полі і блокування

мінного поля, щоб неможливо було зробити хід після програшу або завершення гри.

Приклад опису користувацького інтерфейсу

Інтерфейс користувача в сучасних популярних реалізаціях гри передбачає використання маніпулятора «миша» для виконання ходів. Натиснення на праву кнопку означає, що гравець хоче відкрити клітинку, над якою знаходиться курсор, натиснення лівої кнопки призводить до маркування/розмаркування клітинки над якою в момент натиснення знаходився курсор.

Також в таких іграх присутнє головне меню яке дозволяє почати нову гру, вибрати налаштування мінного поля, такі як розміри і кількість мін а також можливості перегляду таблиці рекордів та імен найкращих гравців.

Алгоритми малювання мінного поля зазвичай досить складні, так як передбачають інтенсивне малювання великої кількості об'єктів – закритих, відкритих клітинок, маркування у вигляді прапорців, виведення кількості замінованих сусідніх мін і тому подібне. Один із можливих підходів – реалізувати мінне поле, як двовимірний масив стандартних Windows-кнопок, але подібна реалізація працює надзвичайно повільно. Як альтернативний варіант можна розглянути мінне поле як одну велику канву, на якій відбувається малювання. Такий підхід хоч і має коректну швидкість промальовування, але він вимагає великої кількості процесорного часу, що може бути небажаним для кінцевого користувача, що працює у багатозадачній операційній системі. В такому середовищі найефективнішим методом малювання мінного поля є використання потужності графічного прискорювача для генерування зображення.

Саме тому для реалізації цієї задачі була вибрана відкрита бібліотека OpenGL. OpenGL є набором портованих на .NET відкритих бібліотек, що дозволяють повніше використовувати апаратні можливості обладнання, на якому працює користувач: бібліотека OpenGL, що дозволяє напряму програмувати графічний автомат сучасних графічних процесорів, OpenAL – для програмування апаратного звуку і OpenCL – бібліотека, що являє собою

Тюрінг-повне розширення для програмування на графічних процесорах. Малювання мінного поля здійснюватиметься з використанням бібліотеки OpenGL.

Користувацький інтерфейс реалізований у вигляді головної форми, яка має головне меню, у якому відображаються пункти меню «Нова гра», «Налаштування», «Список рекордів» та «Вихід». У нижній частині вікна розміщено рядок стану, на якому відображається кількість маркованих мін.

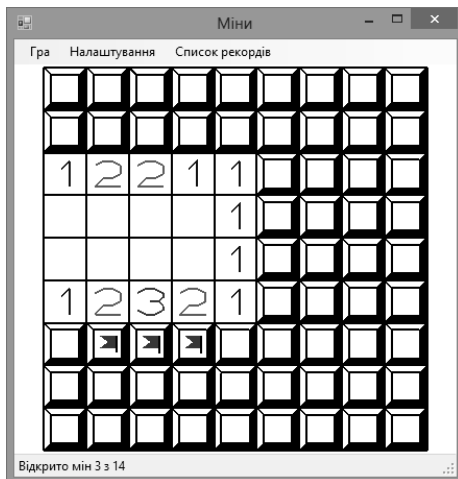


Рис. 5. Вигляд головного вікна програми

Всередині головного вікна розміщено палітру OpenGL, на якій і відбувається малювання мінного поля. Реалізація алгоритмів малювання елементів міститься в методах DrawClosedCell, DrawFlag, DrawNumber, DrawOpenedCell, DrawBomb. Координує малювання елементів обробник подій GameView_Paint. Відповідні елементи меню викликають меню налаштувань і таблиці рекордів.

Приклад опису керівництва користувача програмного продукту

Після першого ходу гравця повинна відкритись вільна від мін область.

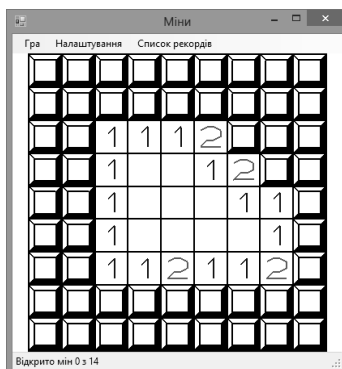


Рис. 6. Розкриття вільної області після першого ходу.

Рівні складності реалізуються зміною розмірів мінного поля і кількості мін. Гра має три рівні складності.

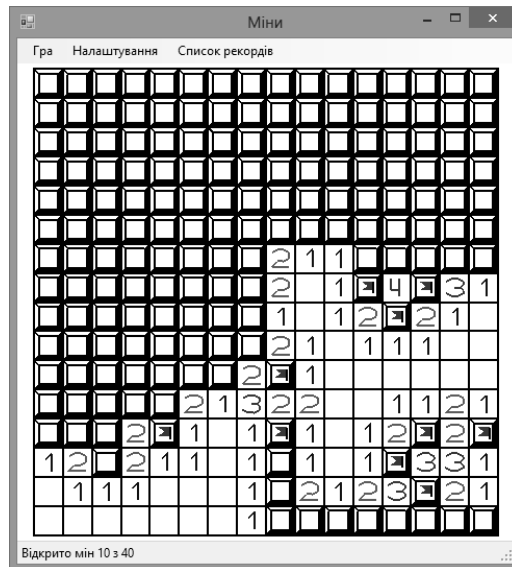


Рис. 7. Вигляд мінного поля при грі на вищому рівні складності

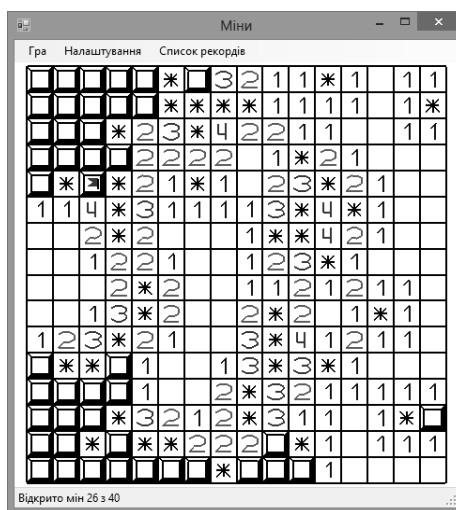


Рис. 8. Закінчення гри і відображення всіх мін.

Також, коли гравець програє, все мінне поле вибухає, також показуються неправильно зроблені ходи (маркування клітинок, що не містять мін).

3.6. Порядок підготовки матеріалів у курсовій роботі

При підготовці матеріалів курсової роботи до підшивання у папку потрібно скласти їх у такому порядку та пронумерувати сторінки (нумеруються не всі сторінки – див. список нижче по тексту):

- 1) титульний лист курсової роботи (не нумерується),
- 2) індивідуальне завдання на курсову роботу (не нумерується),
- 3) вступ (нумерація сторінок починається з 3 сторінки);
- 4) розділ 1 з відповідними пунктами, розміщеними у порядку нумерації (нумерація продовжується);
- 5) розділ 2 (і наступні ...) з відповідними пунктами, розміщеними у порядку нумерації (нумерація продовжується);
- 6) висновки (нумерація продовжується);
- 7) список використаних джерел (нумерація продовжується);
- 8) додатки у порядку, що визначений їх нумерацією: А, Б, В ... (сторінки додатків нумеруються окремо, починаючи з першої, але після номера вказується буквене позначення додатку, наприклад: 1А, 2А і т.д.);

Пояснювальна записка до курсової роботи доповнюється CD-диском, який розміщується у конверті, приклеєному скотчем до внутрішньої сторони обкладинки наприкінці роботи.

4. Написання доповіді та створення презентації до захисту курсової роботи

Для представлення наукових та практичних результатів потрібно підготувати *доповідь*. Зміст доповіді залежить від характеру роботи. Так, для представлення курсової роботи на захисті доповідь має таку структуру:

- *Шановні члени комісії та присутні, Вашій увазі пропонується курсова робота на тему: «...». Виконавець ..., керівник ...*
- короткий виклад мети та задач роботи,
- послідовний виклад результатів курсової роботи: обрані алгоритми та методи реалізації задач, граф-схеми алгоритмів з коментарями їх особливостей, структурна схема програмного продукту (всі положення ілюструються схемами, алгоритмами та ін. з пояснювальної записки),
- коротка характеристика створеного програмного продукту (результату курсової роботи), функціонал, інтерфейс користувача,
- практичне застосування результатів роботи, а також перспективи подальшого розвитку програмного продукту (якщо такі передбачаються).

- *Доповідь закінчена. Дякую за увагу!*

Доповідь розраховують на 4-5 хвилин. Доповідач має підготуватися до ймовірних запитань по курсовій роботі. Деякі з них він викликає сам, коротко говорячи про результати (але інакше в межах виділеного часу неможливо). Інші виникають по причині неякісного опрацювання теоретичного матеріалу чи неповного виконання завдання. Студент має підготуватися до відповіді на такі питання. Якщо не знаєте відповіді на поставлене запитання і цього не вимагалось індивідуальним завданням, можна сказати (там де це коректно), що така задача не ставилася.

Для викладення доповіді доповідач створює презентацію, яка містить ілюстративний матеріал до доповіді. Вона складається з послідовності слайдів, які можуть викладатися у наступній послідовності:

- ✓ **1 слайд.** Тема курсової роботи, виконавець та керівник.
- ✓ **2 слайд.** Мета та задачі, які були поставлені на курсову роботу.
- ✓ **3 ÷ n слайд.** Демонстрація отриманих результатів (за розділами доповіді), це можуть бути текстові блоки, таблиці, блок-схеми алгоритмів, структурні схеми та ін., на одному слайді не слід розміщувати більше 8-10 рядків тексту та 1-2 схем (блок-схем), кожен слайд повинен мати заголовок.
- ✓ **Останній слайд.** Висновки за отриманими результатами (коротко).
- ✓ **Додатковий слайд** (не обов'язковий). Перспективи і проблеми, які опрацьовує виконавець для подальшого розвитку теми роботи.

5. Порядок рецензування курсової роботи

Після повного виконання та оформлення пояснювальної записки до курсової роботи студент надає готовий матеріал на рецензування (для написання відгуку) керівнику курсової роботи.

Рецензія – (від лат. *recensio* – розгляд, обслідування) – це вид наукової, літературної і художньої критики, науково-критична стаття, що дає оцінку досліджуваному твору [14].

Рецензуючи курсову роботу, керівник враховує ряд чинників: 1) відповідність змісту курсової роботи тому плану, який був зазначений у індивідуальному завданні, 2) ступінь виконання роботи, 3) якість ілюстрування роботи блок-схемами, схемами, рисунками, формулами, 4) якість оформлення пояснювальної записки (уміння підбирати і аналізувати фахову інформацію, керуватися нею при виконанні практичних завдань, а також грамотність викладу матеріалу), 5) зазначає недоліки, якщо такі виявлені, 6) робить загальний висновок щодо допуску чи не допуску студента до захисту, виставляє оцінку керівника. Зразок відгуку наведений у додатку В.

6. Перелік типових тем курсових робіт

Теми курсових робіт з дисциплін «Основи програмування», «Програмування та алгоритмічні мови», «Алгоритмізація та програмування», мають відповідати ряду вимог. Це перша курсова робота в навчальних планах ІТ-спеціальностей. Дана курсова робота має на меті перевірити навички студентів у створенні простих програмних продуктів та оформленні супровідної документації. Тема має формулюватися з врахуванням того, що студент має обрати та реалізувати не менше двох основних алгоритмів для реалізації завдання, а також побудувати простий інтерфейс програми (10-12 елементів). Нижче наведений перелік типових тем курсових робіт.

1. Гра «Хоніх». (Створити спрощений аналог популярної гри, у якій ігрове поле поділене на «сушу» і «море» а гравець, керуючи курсором намагається захопити якомога більше території під «сушу». При цьому гравцеві заважають «морські» та «сухопутні» монстри).

2. Гра «Меморі». (Створити гру, що являє собою поле на якому розташовані картки з картинками. Картки лежать у випадковому порядку, і кожна з них представлена на поле 2 рази. Спочатку картки закриті, в кожен момент часу можуть бути відкриті тільки дві картки (не рахуючи вже знайдених пар). При відкритті двох однакових картинок, пари залишаються на полі у відкритому вигляді. Мета гри – послідовно відкрити все поле).
3. Програма візуалізації методів сортування. (Створити програмний продукт, який за допомогою візуальних образів, наприклад, ліній різної довжини чи комірок з числами демонстрував покрокову роботу різних алгоритмів сортування).

Список рекомендованої літератури

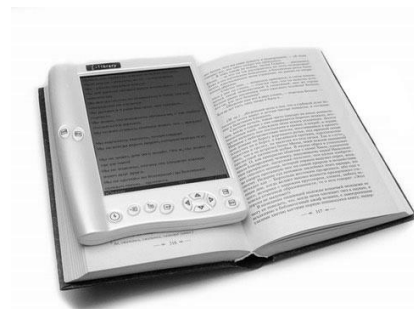
1. Андерсон Д. Дискретная математика и комбинаторика: Пер. с англ. – М.: Издательский дом «Вильямс», 2003. – 960 с.
2. Вирт Н. Алгоритмы и структуры данных.— М.: ДМК_Пресс, 2011. – 272 с.
3. Левитин А.В. Алгоритмы: введение в разработку и анализ. : Пер. с англ. – М.: Издательский дом «Вильямс», 2006. – 576 с.
4. Кормен Т., Лейзерсон Ч., Риверст Р., Штайн К. Алгоритмы: построение и анализ, 2-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2005. – 1296 с.
5. Макконнелл Дж. Основы современных алгоритмов. – М.: Техносфера, 2004. – 368с.
6. Кнут Д. Искусство программирования, том 1. Основные алгоритмы, 3-е изд. : Пер. с англ. – М.: Издательский дом «Вильямс», 2007. – 720 с.
7. Кнут Д. Искусство программирования, том 2. Получисленные алгоритмы, 3-е изд. : Пер. с англ. – М.: Издательский дом «Вильямс», 2005. – 832 с.
8. Кнут Д. Искусство программирования, том 3. Сортировка и поиск. – М.: Издательский дом «Вильямс», 2011. – 824 с.
9. Ахо А., Дж. Хопкрофт, Дж. Ульман. Структуры данных и алгоритмы. – М.: Издательский дом «Вильямс», 2010. – 384 с.
10. Седжвик Р. Фундаментальные алгоритмы на C++. ч. 1-5. Анализ / Структуры данных / Сортировка / Поиск : Пер. с англ. / Роберт Седжвик. - К.: Издательство «ДиаСофт», 2003. – 688 с.
11. Окулов С. М. Программирование в алгоритмах. – М.: БИНОМ, Лаборатория знаний, 2002. – 341 с.
12. Гудман С., С. Хидетниemi. Введение в разработку и анализ алгоритмов. – М.: Мир, 1981.
13. Романовский И.В. Вычислительная математика и структура алгоритмов. – М.: Изд-во МГУ. 2006. – 112 с.



14. Методичні рекомендації щодо оформлення курсових і магістерських робіт для студентів факультету інформатики НАУКМА – К.: Вид. НАУКМА, 2010. – 38 с.
15. Шейко В.М., Кушнарєнко Н.М. Організація та методика науково-дослідницької діяльності: Підручник. – 2-е вид., перероб. і доп. – К.: Знання-Прес, 2002. – 295 с.
16. Блок-схема. [Електронний документ]. Режим доступу: <http://uk.wikipedia.org/wiki/Блок-схема>. Перевірено: 14.12.2012.
17. Старикова М.Е. Составление блок-схем. [Электронный документ]. Режим доступа: <http://festival.1september.ru/articles/588437/>. Проверено: 16.12.2012.
18. Алгоритмические структуры. [Электронный документ]. Режим доступа: <http://inf1.info/flowchart>. Проверено: 16.12.2012.
19. Мейер Б. Основы объектно-ориентированного программирования [Электронный документ]. Режим доступа: <http://www.intuit.ru/department/se/oopbases/>. Проверено 20.01.2013.
20. Греди Буч. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. – СПб: Невский Диалект, 1998. – 560 с.

Internet – посилання

1. <http://algolist.manual.ru/> – велика гарно структурована колекція алгоритмів
2. <http://lib.mexmat.ru/books/780> – електронна бібліотека опікунської ради механіко-математичного факультету Московського держуніверситету.
3. <http://www.iqlib.ru/support/about.visp> – електронна бібліотека освітніх і просвітницьких видань.
4. ips.ifmo.ru/courses/coursesinfo/index.html - курс С. Е. Столяра «Введение в алгоритмику»
5. <http://neerc.ifmo.ru/wiki/index.php> – Станкевич А.С. Дискретная математика, алгоритмы и структуры данных.



6. <http://www.diary.ru/~eek/p67723918.htm> – математика в технічному університеті – учбові матеріали МВТУ ім. Н.Е.Баумана.
7. <http://www.bookarchive.ru/computer/programming/page/1/> – безкоштовна електронна бібліотека книг і журналів з інформатики і програмування.
8. <http://www.knigonosha.net/computer/> – електронна бібліотека «Книгоноша».
9. <http://habrahabr.ru/search/> – ресурс, що має багато інформації для програмістів.
10. <http://jurnalprogman.narod.ru> – Internet-журнал «ProgMan» присвячений програмуванню, в ньому розміщено кількість статей по програмуванню під Windows та Web, є уроки програмування на Delphi, C++ та ін.
11. <http://www.mzero.by.ru> – зміст журналу «Zero» – це статті з програмування та web-дизайну, огляди ПЗ та різних версій Windows.



ДОДАТОК А. Індивідуальне завдання на курсову роботу
МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
ЧЕРКАСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ім. Б. ХМЕЛЬНИЦЬКОГО
(НАЗВА КАФЕДРИ, НА ЯКІЙ ВИКОНУЄТЬСЯ КУРСОВА РОБОТА)

ЗАТВЕРДЖУЮ
Завідувач кафедри _____,

(підпис)
„____” _____ 2015 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на курсову роботу

студента _____ групи _____ першого курсу

ТЕМА _____

Вихідні дані до курсової роботи: _____

Зміст **Пояснювальної записки** до курсової роботи:

- Індивідуальне завдання
- Вступ
- 1 Огляд ...
- 2 Розробка схеми алгоритму ...
- 3 Розробка програми ...
- Висновки
- Список літератури
- Додатки (за необхідності)

Перелік наочного матеріалу:

Програмний продукт ..., пояснювальна записка, презентація результатів роботи.

Календарний план виконання роботи:

№ п/п	Назва етапу дипломного проекту (роботи)	Термін виконання етапу: до ...	Примітка
1.	Отримання завдання на курсову роботу.	04.02.2015	
2.	Огляд джерел технічної інформації за темою роботи.	18.02.2015	
3.	Виконання аналізу методів реалізації завдання	04.03.2015	
3.	Розробка алгоритму реалізації завдання	18.03.2015	
4.	Програмування алгоритму, створення програмного продукту	08.04.2015	
5.	Тестування розробленого програмного продукту	15.04.2015	
6.	Написання пояснювальної записки	01.05.2015	
7.	Подача роботи керівнику для написання відгуку.	07.05.2015	
8.	Корегування роботи за результатами розгляду керівника. Остаточне оформлення пояснювальної роботи.	14.05.2015	
9.	Написання доповіді, створення слайдів.	15.05.2015	
10.	Захист курсової роботи	16.05.2015	

Студент _____ / _____ /
(підпис) (Прізвище, ініціали)

Науковий керівник _____ / _____ /
(підпис) (Прізвище, ініціали)

Завдання видане « ____ » _____ 2015 року

ДОДАТОК Б. Титульний лист курсової роботи

Черкаський національний університет імені Богдана Хмельницького

(повне найменування вищого навчального закладу)

(Назва кафедри, на якій виконується курсова робота)

(повна назва кафедри)

КУРСОВА РОБОТА

**з дисципліни “Назва дисципліни за індивідуальним
навчальним планом (Основи програмування,
Програмування та алгоритмічні мови,
Алгоритмізація та програмування)”**

НА ТЕМУ “...”

Студента (ки) 1 курсу, групи КС-11
напряму підготовки «Програмна інженерія»

(прізвище та ініціали)

Керівник _____

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала: _____

Кількість балів: _____ Оцінка: ECTS _____

Члени комісії

(підпис)

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

м. Черкаси – 2015 рік

ДОДАТОК В. Приклад відгуку керівника на курсову роботу

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ ЧЕРКАСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ім. Б. ХМЕЛЬНИЦЬКОГО (НАЗВА КАФЕДРИ, НА ЯКІЙ ВИКОНУЄТЬСЯ КУРСОВА РОБОТА)

Відгук на курсову роботу

студента _____ групи _____ другого курсу

1. Тема роботи: _____
2. Актуальність теми: Алгоритми сортування знаходять широке застосування при реалізації ... та інших функцій, пов'язаних з вибором оптимального варіанту розв'язання задачі. Однак, їх практичне застосування обмежене великими обчислювальними витратами при великому обсязі вхідних даних. Тому актуальним є вибір таких алгоритмів сортування, які забезпечать роботу користувача в режимі реального часу.
4. Відповідність змісту роботи її плану: зміст роботи повністю відповідає її плану.
5. Ступінь виконання роботи: робота виконана повністю. Наведено огляд існуючих алгоритмів сортування та детально описаний алгоритм Проведений опис основних етапів проектування та створення ... (програмного продукту).
6. Ілюстрованість роботи (наявність розрахунків, таблиць, схем, діаграм, тощо):
Робота ілюстрована в достатній мірі граф-схемами алгоритмів, таблицями та схемами.
7. Якість оформлення роботи:
Оформлення роботи відповідає вимогам Державного стандарту України. При оформленні використано сучасні комп'ютерні інформаційні технології (можна конкретизувати які).
9. Недоліки:
В роботі недостатньо описаний етап розробки інтерфейсу користувача та не конкретизовано причини вибору алгоритму роботи з циклічним списком.
10. Загальний висновок (допускається чи не допускається до захисту), якої оцінки заслуговує робота:
Робота виконана на _____ технічному рівні і допускається до захисту з оцінкою « _____ ».

кандидат фіз.-мат. наук,
доцент кафедри ПЗАС _____

Б.О. Онищенко

" ____ " _____ 2015 р.

Навчально-методичне видання

Гребенович Юлія Євгенівна

Онищенко Борис Олегович

Супруненко Оксана Олександрівна

**Методичні рекомендації
до виконання та оформлення
курскових робіт з дисциплін
“Основи програмування”,
“Програмування та алгоритмічні мови”,
“Алгоритмізація та програмування”
для студентів напрямів підготовки
050101 – „Комп’ютерні науки”,
050103 – „Програмна інженерія”,
040303 – „Системний аналіз”
усіх форм навчання**

*Комп’ютерне верстання
Л.Г. Любченко*

Підписано до друку 11.01.2016. Формат 60×84/16.
Ум. друк. арк. 1,3. Тираж 50 прим. Зам. № 5789.

Видавець і виготовник видавничий відділ
Черкаського національного університету імені Богдана Хмельницького.
Адреса: 18000, м. Черкаси, бул. Шевченка, 81, кімн. 117,
тел. (0472) 37-13-16, факс (0472) 37-22-33,
e-mail: vydav@cdu.edu.ua, <http://www.cdu.edu.ua>
Свідоцтво про внесення до державного реєстру
суб’єкт видавничої справи ДК № 3427 від 17.03.2009 р.