

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРКАСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ім. Б. ХМЕЛЬНИЦЬКОГО

Факультет обчислювальної техніки,
інтелектуальних та управляючих систем
Кафедра інформаційних технологій

Царик Т. Ю., Веретельник В. В

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ

до виконання та оформлення курсової роботи з
дисципліни:

**«Проектування автоматизованих інформаційних
систем та технологій»**

(для здобувачів вищої освіти денної та заочної форм навчання)

Затверджено
на засіданні кафедри
інформаційних технологій
Протокол № 16 від 26.05.2023 р

Черкаси 2023

УДК 372.862

Методичні рекомендації до виконання та оформлення курсової роботи з дисципліни «Проектування автоматизованих інформаційних систем та технологій» для здобувачів ОС «магістр» спеціальність 122 «Комп'ютерні науки» ООП «Інформаційні управляючі системи та технології» 2023 р.

Укладачі:

Т. Ю. Царик, к. т. н., старший викладач кафедри інформаційних технологій
В. В. Веретельник, к. т. н., доцент кафедри інформаційних технологій

Рецензенти:

Фауре Е. В., д. т. н., професор кафедри інформаційної безпеки та комп'ютерної інженерії Черкаського державного технологічного університету.

Осауленко І. А., д. т. н., доцент, доцент кафедри інтелектуальних систем прийняття рішень Черкаського національного університету імені Богдана Хмельницького.

*Рекомендовано до друку Вченою радою
Черкаського національного університету імені Богдана Хмельницького
(Протокол № 1 від 31.08.2023 р.)*

Методичні рекомендації до виконання та оформлення курсової роботи з дисципліни «Проектування автоматизованих інформаційних систем та технологій». У методичних вказівках розкрито мету й задачі проектування та розробки, докладно охарактеризовані вимоги до структури, змісту, оформлення документації, також описані порядок, вимоги та критерії оцінки захисту курсової роботи. Матеріал методичних вказівок призначений для здобувачів ос «магістр, наукових керівників та рецензентів, членів комісій із захисту курсових робіт спеціальності 122 «Комп'ютерні науки» ОПП «Інформаційні управляючі системи і технології».

© Царик Т. Ю., укладання, 2023
© Веретельник В. В., укладання, 2023
© ЧНУ ім. Б. Хмельницького, укладання, 2023

ЗМІСТ

1. ЗАГАЛЬНІ ВІДОМОСТІ	4
2. СТРУКТУРА ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ КУРСОВОЇ РОБОТИ	4
3. ПРАВИЛА ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ	7
3.1. Побудова і рубрикація тексту	7
3.2. Оформлення рисунків.....	8
3.3. Оформлення таблиць	8
3.4. Оформлення посилань на джерела.....	9
4. ЗАВДАННЯ ДЛЯ ВИКОНАННЯ КУРСОВОЇ РОБОТИ	9
5. ОЦІНЮВАННЯ ТА ЗАХИСТ КУРСОВОЇ РОБОТИ.....	11
6. НАВЧАЛЬНО-МЕТОДИЧНА ЛІТЕРАТУРА.....	12
6.1. Базова.....	12
6.2. Додаткова	13
6.3. Електронні ресурси.....	13
ДОДАТОК А. Приклад титульного листа	14
ДОДАТОК В. Приклад листа оцінювання.....	16
ДОДАТОК Д. Приклад можливого оформлення змісту	17
ДОДАТОК Ж. Приклад оформлення курсової роботи	18

1 ЗАГАЛЬНІ ВІДОМОСТІ

Курсова робота – це комплект документації, який включає пояснювальну записку (текстові частини) і програмний продукт (курсове завдання та його вирішення – розроблена програма). Курсова робота виконується студентом індивідуально і самостійно під керівництвом і при консультації керівника курсової роботи.

Мета курсової роботи – закріплення теоретичних знань і практичних навиків, одержаних на лекціях, лабораторних заняттях та при виконанні самостійної роботи, які полягають в освоєнні технології ADO.NET в рамках середовища Microsoft Visual Studio .Net та мови програмування C# та використання бази даних Microsoft SQL Server при розробці програмних додатків.

Тематика курсових робіт визначається та затверджується на засіданні кафедри. Під час розробки тем курсових робіт можуть враховуватись побажання студентів, замовлення окремих викладачів або кафедр університету, а також сторонніх організацій ті підприємств. Курсова робота може бути частиною кафедральної комплексної курсової роботи, що виконується декількома студентами. Обмежень щодо обсягу пояснювальної записки немає.

Загальні вимоги щодо оформлення пояснювальної записки курсової роботи описані в Стандарті установи „Випускні, курсові проекти і роботи. Загальні вимоги та правила оформлення”.

2 СТРУКТУРА ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ КУРСОВОЇ РОБОТИ

Структура розміщення матеріалу в записці до курсової роботи повинна бути такою:

- титульний лист (додаток А);
- лист оцінювання (додаток В);
- зміст (додаток С);
- основний зміст курсової роботи по відповідних розділах;
- висновки;
- перелік використаних літературних джерел;
- додатки (якщо такі є).
- вклеєний конверт із CD-диском, на який записано: текст пояснювальної записки, програмний код розробленого програмного забезпечення.

Загальний вигляд оформленої курсової роботи наведено у додатку D.

Титульний лист курсової роботи оформлюється згідно з додатком А, форма якого встановлюється діючим державним стандартом ведення документообігу у вищих навчальних закладах. При зміні встановленої форми титульного листа студенти повідомляються додатково.

Лист оцінювання оформлюється згідно з додатком С. Містить критерії оцінювання для отримання максимального значення кількості балів за курсову роботу – 100 балів. Для кожної курсової роботи сума балів розраховується у відповідності до критеріїв оцінювання.

Зміст – структурна одиниця пояснювальної записки, де повинні бути вказані порядкові номери (якщо вони є) і заголовки всіх структурних одиниць записки, включаючи вступ, висновки, перелік посилань і додатки, для кожної назви розділу зазначається відповідна сторінка пояснювальної записки. Приклад оформлення змісту наведено в додатку С.

Основний зміст курсової роботи по відповідних розділах – це структурна частина пояснювальної записки, що містить основний зміст курсової роботи. Основний зміст курсової роботи може бути викладено у довільній формі, але в обов'язковому порядку містити наступні пункти:

1. Опис предметної області, що повинно включати:

1.1. Мета та доцільність створення АІС. Визначення задач, що підлягають автоматизації в обраній предметній області. Зазначення мети та доцільності створення АІС;

1.2. Схема документообігу, описання документів вхідної інформації – описується загальна схема взаємодії об'єктів предметної області та інформаційні потоки, що супроводжують цю взаємодію. Наводяться форми документів. Здійснюється опис вмісту електронних документів (їх атрибутів, кількості) майбутньої АІС.

1.3. Функціональна модель предметної області.

1.4. Об'єктна модель предметної області. Описання об'єктів предметної області, їх атрибутів, взаємозв'язків та взаємодії – Описуються об'єкти предметної області, із зазначенням їх властивостей та взаємозв'язки між ними. Здійснюється опис створення абстракцій для розв'язання поставленого завдання (ви- бір значущих об'єктів та їх властивостей), що будуть складати нормативно- довідникову інформацію системи.

2. Структура меню АІС – наводяться основні складові частини АІС та їх вміст (НДІ, вхідна інформація, вихідна інформація).

3. Розробка бази даних. В цій частині наводиться логічна та фізична структура розробленої бази даних. Наводиться текст SQL-запитів формування вихідної інформації. Розробка БД виконується з використання норм нормалізації даних в рамках СУБД MS SQL Server. Scripts створення БД наводяться в додатку

А.

- 3.1. Логічна структура баз даних
- 3.2. Фізична модель бази даних
- 3.3. SQL-запити вихідної інформації

4. Інтерфейсна частина АІС.

- 4.1. Профілі користувачів. Механізми розмежування прав доступу до даних.
- 4.2. Екранні форми АІС з описом діалогу користувача з формою додатка.

До опису функціонального призначення додається список можливих дій користувача в рамках екранної форми.

5. Програмна реалізація. Містить опис застосованих програмних механізмів валідації даних, обробки виключних подій, механізми підтримки роботи користувача (підказки, повідомлення), механізми доступу до даних БД. Лістинги наводяться у відповідності до переліку екранних форм у додатку Б.

Висновки – структурна одиниця пояснювальної записки, де стисло формулюють зміст усіх частин записки і одержані результати, включаючи характеристику прийнятих проектних та програмних рішень та їх ефективність з технічної, економічної точок зору.

Перелік посилань – структура одиниця пояснювальної записки, що оформлюється у вигляді бібліографічного опису джерел, що були використані, розміщують перед додатками, починаючи з нового аркуша. Джерела розміщують у переліку в порядку їх згадування в тексті і позначають порядковими номерами арабськими цифрами, які вказують перед бібліографічним описом джерела, відділяючи від нього крапкою

Додатки – структурна одиниця пояснювальної записки, що містить лістинги створення створення БД та програмного забезпечення АІС.

3 ПРАВИЛА ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Текст пишеться державною мовою. В тексті (крім цитат) не допускається:

- вживати зврати розмовної мови;
- вживати застарілі та жаргонні терміни і вислови;
- вживати скорочені слова, крім встановлених чинними стандартами і загальноприйнятих в літературній мові.

Основний текст: TimesNewRoman 14; інтервал 1,5; абзац 1,25; розташування – по ширині. Текст лістингу програм: TimesNewRoman 10; інтервал 1; абзац 1,25 можна в 2 колонки;

3.1 Побудова і рубрикація тексту

1. Текстовий документ розділяють на частини (розділи). Частини (розділи) за необхідності поділяють на підрозділи, а їх, в свою чергу, на пункти і підпункти. У цьому випадку кожна з зазначених структурних одиниць повинна включати не менше двох структурних одиниць більш низького рівня. Частини та їх підрозділи повинні мати заголовки. Пункти і підпункти можуть не мати заголовків.

2. Частини (розділи) повинні мати порядкові номери в межах усього документу. Вступ, висновки і перелік посилань не нумеруються. Номери частин (розділів) позначають арабськими цифрами і записують перед заголовком частини (розділу) без крапки.

Номер підрозділу, пункту чи підпункту складається з номера структурної одиниці більш високого рівня і номера даної структурної одиниці, поділених крапкою. В кінці номера крапку не ставлять. Номер структурної одиниці зазначають перед її заголовком, а при його відсутності – перед початком тексту даної структурної одиниці.

3. Кожну частину починають з нового аркуша. Текст кожної структурної одиниці починають з абзацу.

4. Заголовки друкують без крапки в кінці, не підкреслюючи. Якщо заголовок складається з кількох речень, їх поділяють крапкою. Заголовки розділів друкують великими літерами грубим шрифтом і розміщують симетрично тексту. Заголовки підрозділів, пунктів та підпунктів (якщо вони мають заголовки) друкують малими літерами (крім першої великої) грубим шрифтом і розміщують з абзацу. Перенесення слів в заголовках не допускається.

3.2 Оформлення рисунків

1. Рисунки нумерують в межах кожної частини (розділу) двома числами номером частини, розділу або позначенням додатку і порядковим номером рисунка – поділеними крапкою.

2. На всі рисунки повинні бути посилання в тексті, наприклад, ... наведено на рисунку 4.2.

3. Кожний рисунок повинен мати найменування. Слово "Рисунок", його номер і найменування розміщують під рисунком і розташовують таким чином:

Рисунок 2.1 – Деталі приладу

Після найменування рисунка крапку не ставлять.

4. За необхідності над номером і найменуванням рисунка розміщують пояснювальні дані (так звану "легенду": розшифровку номерів позицій на кресленнях або ескізах, позначення кривих на графіках тощо).

3.3 Оформлення таблиць

1. Таблиці нумерують в межах кожної частини (розділу) двома числами - номером частини, розділу або позначенням додатка і порядковим номером таблиці, поділеними крапкою.

2. На всі таблиці повинні бути посилання в тексті, наприклад: ... наведено в таблиці 6.1.

3. Таблиця повинна мати найменування, яке вказують над таблицею. Перед найменуванням таблиці пишуть слово "Таблиця" та її номер, який відділяють від найменування за допомогою тире, наприклад:

Таблиця 4.1 – Вихідні дані

4. Розташування підпису – по лівій стороні.

3.4 Оформлення посилань на джерела

1. В тексті повинні бути посилання на всі джерела, що були використані. Посилатися слід, як правило, на джерело в цілому. За необхідності допускаються посилання на розділи, таблиці, ілюстрації чи сторінки джерела.

2. Використання запозичених даних без посилання на джерело не допускається і розглядається як плагіат.

3. Посилання на джерело наводиться у вигляді його порядкового номеру в переліку посилань, узятого в квадратні дужки

Нумерація сторінок пояснювальної записки повинна бути наскрізною, починаючи з титульного аркуша, включаючи вкладки і додатки. Номер сторінки ставлять в середині верхньої частини аркуша над текстом. На титульному аркуші, завданні, оціночному аркуші і вкладках номери сторінок не пишуть.

4 ЗАВДАННЯ ДЛЯ ВИКОНАННЯ КУРСОВИХ РОБІТ

1. Розробка проекту АІС та пакету прикладних програм „Контроль видачі супровідних документів при транспортуванні сировини на виробничому підприємстві”
2. Розробка проекту АІС та пакету прикладних програм „Облік відпрацьованого часу співробітників підприємства”
3. Розробка проекту АІС та пакету прикладних програм „Облік витрати продуктів в їдальні виробничого підприємства”
4. Розробка проекту АІС та пакету прикладних програм „Облік виробленої готової продукції на виробничому підприємстві”
5. Розробка проекту АІС та пакету прикладних програм „Облік основних засобів на виробничому підприємстві”
6. Розробка проекту АІС та пакету прикладних програм „Контроль виконання дого-

вірних зобов'язань з постачальниками сировини на виробничому підприємстві”

7. Розробка проекту АІС та пакету прикладних програм „Контроль проведення інструктажів по техніці безпеки на виробничому підприємстві”
8. Розробка проекту АІС та пакету прикладних програм „Формування портфелю замовлень на виробничому підприємстві”
9. Розробка проекту АІС та пакету прикладних програм „Контроль витрат палива на виробничому підприємстві”
10. Розробка проекту АІС та пакету прикладних програм „Контроль оплати за відвантажену готову продукцію на виробничому підприємстві”
11. Розробка проекту АІС та пакету прикладних програм „Облік руху грошових коштів в касі виробничого підприємства”
12. Розробка проекту АІС та пакету прикладних програм „Облік транспортних засобів на виробничому підприємстві”
13. Розробка проекту АІС та пакету прикладних програм „Облік кадрів по підрозділам організаційної структури виробничого підприємства”
14. Розробка проекту АІС та пакету прикладних програм „Облік руху сировини на складах виробничого підприємства”
15. Розробка проекту АІС та пакету прикладних програм „Облік руху готової продукції між виробництвом та складами виробничого підприємства”
16. Розробка проекту АІС та пакету прикладних програм „Контроль виконання договірних зобов’язань зі споживачами готової продукції на виробничому підприємстві”
17. Розробка проекту АІС та пакету прикладних програм „Облік малоцінних швидкозношуваних предметів на виробничому підприємстві”
18. Розробка проекту АІС та пакету прикладних програм „Контроль проведення ремонтів обладнання на виробничому підприємстві”
19. Розробка проекту АІС та пакету прикладних програм „Облік руху грошових коштів на розрахунковому рахунку виробничого підприємства”
20. Розробка проекту АІС та пакету прикладних програм „Облік товарно-матеріальних цінностей на складі виробничого підприємства”
21. Розробка проекту АІС та пакету прикладних програм „Контроль виконання договорів із підрядними організаціями за проведення ремонтних робіт”
22. Розробка проекту АІС та пакету прикладних програм „Контроль витрат сировини при виробництві готової продукції на виробничому підприємстві”.
23. Розробка проекту АІС та пакету прикладних програм „Контроль оплати за поставлену сировину на виробничому підприємстві”.
24. Розробка проекту АІС та пакету прикладних програм „Інвентаризація по підрозділам на виробничому підприємстві”.

25. Розробка проекту АІС та пакету прикладних програм „Контроль виплат за комунальні послуги на виробничому підприємстві”.
26. Розробка проекту АІС та пакету прикладних програм „Формування графіку ремонтів обладнання на виробничому підприємстві”.
27. Розробка проекту АІС та пакету прикладних програм „Формування графіку відпусток працівників виробничого підприємства”
28. Розробка проекту АІС та пакету прикладних програм „Планування постачання на виробничому підприємстві”
29. Розробка проекту АІС та пакету прикладних програм „Планування збуту на виробничому підприємстві”.
30. Розробка проекту АІС та пакету прикладних програм „Планування виробництва на виробничому підприємстві”.
31. Розробка проекту АІС та пакету прикладних програм „Нарахування заробітної плати за відпрацьований час працівників виробничого підприємства”
32. Розробка проекту АІС та пакету прикладних програм „Облік прибутку на виробничому підприємстві”.
33. Розробка проекту АІС та пакету прикладних програм „Облік руху товарів в автосалоні”
34. Розробка проекту АІС та пакету прикладних програм „Облік призовників у воєнкоматі”
35. Розробка проекту АІС та пакету прикладних програм „Контроль виконання договірних зобов’язань рекламним агентством”
36. Розробка проекту АІС та пакету прикладних програм „Автосервіс”
37. Розробка проекту АІС та пакету прикладних програм „Організація роботи агентства нерухомості”
38. Розробка проекту АІС та пакету прикладних програм „Готельний комплекс”
39. Розробка проекту АІС та пакету прикладних програм „Облік товарів у книгарні”
40. Розробка проекту АІС та пакету прикладних програм „Продаж квитків в аеропорту”
41. Розробка проекту АІС та пакету прикладних програм „Формування журналу відвідувань у салоні краси”
42. Розробка проекту АІС та пакету прикладних програм „Продаж квитків на залізниці”.

5 ОЦІНЮВАННЯ ТА ЗАХИСТ КУРСОВОЇ РОБОТИ

Результатом виконання курсової роботи є оформлена у відповідності до вимог пояснювальна записка, що містить проект розроблюваної АІС та функціонуюче програмне забезпечення АІС, створене засобами мови програмування С# в рамках програмного середовища Microsoft Visual Studio .Net із використанням технології ADO.NET та СУБД MSSQL Server. Невід'ємною частиною оцінки за курсову роботу є прилюдний захист курсової роботи. Захист курсової роботи може відбуватися в присутності комісії, викладачів та студентів кафедри, при необхідно – за наявності фахівців з розробки та тестування програмного забезпечення сторонніх організацій.

Оцінка за курсову роботу є комплексною та складеною у відповідності до таблиці 1.

Таблиця 1 – Критерії оцінювання

Критерій оцінювання	Максимальна оцінка, балів	Оцінка, балів
Опис предметної області	30	
- вимоги до АІС	5	
- схема документообігу	5	
- функціональна модель предметної області	10	
- об'єктна модель предметної області	15	
Розробка структури меню АІС	5	
Розробка бази даних АІС	15	
- логічна структура бази даних	10	
- фізична модель бази даних	5	
Розробка інтерфейсу користувача	15	
- екранні форми АІС	10	
- застосування елементів підтримки користувача (підказки, інструкції, повідомлення).	5	
Програмна реалізація	30	
- механізми валідації	5	
- механізми обробки виняткових ситуацій	5	
- механізми графічного подання інформації	10	
- формування звітної інформації, друк	5	
- механізми розподіленого доступу до даних	5	
Оформлення пояснювальної записки	5	
Всього	1006	

6 НАВЧАЛЬНО-МЕТОДИЧНА ЛІТЕРАТУРА

1. М.Ф.Бондаренко, С.И. Маторин, Е.А. Соловьева "Моделирование и проектирование бизнес-систем: методы, стандарты, технологии": Учебное пособие. Харьков: Компания СМИТ, 2004, - 304с.

2. Ананьев О.М. Інформаційні системи і технології в комерційній діяльності: підручник / О.М. Ананьев, В.М. Білик, Я.А. Гончарук. - Львів: Новий Світ, 2006. - 583 с.

3. Глівенко С.В. Інформаційні системи в менеджменті: навч. посіб. / С.В.Глівенко, Є.В. Лапін, О.О. Павленко, С.С. Слабко, В.М. Лебідь. - Суми: Університетська книга, 2005. - 407 с.

Міністерство освіти і науки України
Черкаський національний університет ім.Б. Хмельницького
Кафедра Інформаційних технологій

КУРСОВА РОБОТА

з дисципліни „Проектування автоматизованих інформаційних систем та технологій”

на тему: “Розробка інформаційного та програмного забезпечення АІС „Бібліотека”

Студента (ки) курсу групи

Керівник

Національна шкала _____

Кількість балів: _____ Оцінка: ECTS _____

Члени комісії: _____

(підпис)

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

м. Черкаси – 20 _____ рік

ДОДАТОК С. Приклад листа оцінювання

Міністерство освіти і науки України

Черкаський національний університет ім. Б. Хмельницького

Студент . Іванов І.І групи _____

Тема курсової роботи “Розробка інформаційного та програмного забезпечення АІС „Бібліотека”

Оцінювання

Критерій оцінювання	Максимальна оцінка, балів	Оцінка, балів
Опис предметної області	30	
- вимоги до АІС	5	
- об’єктна модель предметної області	15	
- схема документообігу	5	
- функціональна модель предметної області	10	
Розробка структури меню АІС	5	
Розробка бази даних АІС	15	
- логічна структура бази даних	10	
- фізична модель бази даних	5	
Розробка інтерфейсу користувача	15	
- екранні форми АІС	10	
- застосування елементів підтримки користувача (підказки, інструкції, повідомлення).	5	
Програмна реалізація	30	
- механізми валідації	5	
- механізми обробки виняткових ситуацій	5	
- механізми графічного подання інформації	10	
- формування звітної інформації, друк	5	
- механізми розподіленого доступу до даних	5	
Оформлення пояснювальної записки	5	
Всього	100б	

Захист відбувся _____

Загальна оцінка _____

Керівник курсової роботи _____

ЗМІСТ

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ.....	
1.1 Мета та доцільність створення АІС.....	
1.2 Схема документообігу.....	
1.3 Функціональна модель предметної області.....	
1.4 Об'єктна модель предметної області.....	
2 СТРУКТУРА МЕНЮ АІС	
3 РОЗРОБКА БАЗИ ДАНИХ	
3.1 Логічна структура БД.....	
3.2 Фізична структура БД.....	
3.3 SQL-запити вихідної інформації.....	
4 ІНТЕРФЕЙСНА ЧАСТИНА АІС.....	
4.1 Профілі користувачів.....	
4.2 Екранні форми АІС.....	
5 ПРОГРАМНА РЕАЛІЗАЦІЯ.....	
ВИСНОВКИ.....	
ПЕРЕЛІК ПОСИЛАНЬ	
ДОДАТОК А. Лістинг створення бази даних.....	
ДОДАТОК Б. Лістинг створення програмного забезпечення.....	

ДОДАТОК Ж. Приклад оформлення курсової роботи

Черкаський національний університет ім. Б. Хмельницького
Кафедра „Інформаційних технологій”

КУРСОВА РОБОТА

з дисципліни „Проектування автоматизованих інформаційних систем
та технологій” на тему: „Розробка інформаційного та програмного
забезпечення АІС
„Облік витрат палива на ділянці промислового підприємства”

Студента (ки) курсу групи

Іванова Олега Івановича

Керівник

Національна шкала _____

Кількість балів: _____ Оцінка: ECTS _____

Члени комісії: _____

(підпис)

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

м. Черкаси – 20 _____ рік

Міністерство освіти і науки України

Черкаський національний університет ім. Б. Хмельницького

Кафедра «Інформаційних технологій»

Студент . Іванов І.І. групи _____

Тема курсової роботи: Розробка інформаційного та програмного забезпечення АІС „Облік витрат палива на ділянки промислового підприємства”

Оцінювання

Критерій оцінювання	Максимальна оцінка, балів	Оцінка, балів
Опис предметної області	30	
- вимоги до АІС	5	
- функціональна модель предметної області	15	
- схема документообігу	5	
- об'єктна модель предметної області	10	
Розробка структури меню АІС	5	
Розробка бази даних АІС	15	
- логічна структура бази даних	10	
- фізична модель бази даних	5	
Розробка інтерфейсу користувача	15	
- екранні форми АІС	10	
- застосування елементів підтримки користувача (підказки, інструкції, повідомлення).	5	
Програмна реалізація	30	
- механізми валідації	5	
- механізми обробки виняткових ситуацій	5	
- механізми графічного подання інформації	10	
- формування звітної інформації, друк	5	
- механізми розподіленого доступу до даних	5	
Оформлення пояснювальної записки	5	
Всього	100б	

Захист відбувся _____

Загальна оцінка _____

Керівник курсової роботи _____

ЗМІСТ

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ.....	
1.1 Мета та доцільність створення АІС.....	
1.2 Схема документообігу.....	
1.3 Функціональна модель предметної області.	
1.4 Об'єктна модель предметної області.....	
2 СТРУКТУРА МЕНЮ АІС.....	
3 РОЗРОБКА БАЗИ ДАНИХ.....	
3.1 Логічна структура БД.....	
3.2 Фізична структура БД.....	
3.3 SQL-запити вихідної інформації.....	
4 ІНТЕРФЕЙСНА ЧАСТИНА	
4.1 Профілі користувачів.....	
4.2 Екранні форми АІС	
5 ПРОГРАМНА РЕАЛІЗАЦІЯ.....	
ВИСНОВКИ.....	
ПЕРЕЛІК ПОСИЛАНЬ.....	
ДОДАТОК А. Лістинг створення бази даних.....	
ДОДАТОК Б. Лістинг створення програмного забезпечення.....	

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

Процес розробки будь-якої АІС складається з наступних етапів: аналіз предметної області (аналіз вимог), проектування, реалізація, тестування, експлуатація. Під аналізом предметної області розуміється вид дослідження, при якому реальний або уявний об'єкт розділяється на складові частини (елементи) і досліджуються ці елементи і зв'язки між ними. Аналіз предметної області (ПрО) представляє особливий вид наукової діяльності, у результаті якої будується об'єктна та функціональна моделі предметних знань (у широкому змісті).

В основі різних методологій моделювання предметної області АІС лежать принципи послідовної деталізації абстрактних категорій. Зазвичай, моделі будуються на трьох рівнях: на зовнішньому рівні (визначенні вимог), на концептуальному рівні (специфікації вимог) і внутрішньому рівні (реалізації вимог). Так, на зовнішньому рівні модель відповідає на запитання, що повинна робити система, тобто визначається склад основних компонентів системи: об'єктів, функцій, подій, організаційних одиниць, технічних засобів. На концептуальному рівні модель відповідає на запитання, як повинна функціонувати система? Інакше кажучи, визначається характер взаємодії компонентів системи одного і різних типів. На внутрішньому рівні модель відповідає на запитання: за допомогою яких програмно-технічних засобів реалізуються вимоги до системи?

Система моделей предметної області повинна відображати структурний (об'єктна модель) і функціональний (функціональна модель) аспект предметної області.

До моделей предметних областей пред'являються наступні вимоги [1]:

- формалізація, що забезпечує однозначний опис структури предметної області;
- зрозумілість для замовників і розроблювачів на основі застосування графічних засобів відображення моделі;
- можливість реалізування, що припускає наявність засобів фізичної реалізації моделі предметної області в рамках пакету прикладних програм АІС;
- забезпечення оцінки ефективності реалізації моделі предметної області на основі визначених методів і показників, що обчислюються.

1.1 Мета та доцільність створення АІС

Облік – одна з функцій управління. Облік – процес збору даних, що циркулюють в предметній області. Основне завдання обліку – надання достовірної інформації про хід та результати діяльності, яка є основою прийняття управлінсь-

ких рішень. Мета розробки АІС полягає в реалізації автоматизованого обліку палива на дільниці промислового підприємства, з такими завданнями:

- збір, обробка та відображення первинних даних про господарську діяльність;
- систематизація даних з метою отримання та узагальнення підсумкової інформації про господарську діяльність;
- створення вихідної інформаційної бази для планування і здійснення контролю за виконанням планів.

Задачі, що вирішуються в розробленій АІС:

1) Облік витрат палива по водіям. Реалізація автоматизованого обліку витрат палива по водіям здійснюється на основі документу „Подорожній лист”.

Вхідними параметрами запиту є:

- ПІБ водія;
- табельний номер водія.

В результаті виконання запитів до БД користувачеві видається інформації стосовно:

- кількості ходок, здійснених водієм;
- всього витраченого палива;
- кількість кілометрів, що проїхав водій.

2) Облік витрат палива по подорожньому листу. Реалізація автоматизованого обліку палива по подорожньому листу здійснюється на основі документу „Подорожній лист”. Вхідними параметрами запиту є:

- початкова дата вибірки даних;
- кінцева дата вибірки даних;
- ПІБ водія;
- номер державної реєстрації машини

В результаті виконання запитів до БД користувачеві видається інформації стосовно:

- кілометражу;
- виданого палива.

3) Облік витрат палива по техніці. Реалізація автоматизованого обліку палива по машині здійснюється на основі документу „Подорожній лист”. Вхідними параметрами запиту є:

- державний номер машини;

В результаті виконання запитів до БД користувачеві видається інформації стосовно:

- витрати палива за паспортом на 100 км;
- пройденого кілометражу;

- кількості ходок;
- загальної кількості витраченого палива.

Доцільність створення АІС полягає в забезпеченні отримання економічного ефекту від впровадження та використання АІС. Кінцевий ефект від використання АІС „Облік витрат палива технікою на дільниці доменного цеху” полягає:

- у зменшенні витрат часу за рахунок оперативного оновлення бази даних інформацією з документів „Подорожний лист”;
- у спрощенні механізму внесення нової та редагування вже наявної інформації нормативно-довідникової інформації;
- у підвищенні продуктивності праці співробітників відповідальних за облік витрат палива на дільниці доменного цеху;
- у підвищенні ефективності обліку та контролю за витратами палива зі урахуванням виду техніки, виду палива, паспортного нормативу витрати палива.

Таким чином, розробку АІС „Облік витрат палива технікою на дільниці доменного цеху” слід вважати доцільною та актуальною.

1.2 Схема документообігу

Взаємодія об’єктів предметної області фіксується в документах, що є юридичним свідомством здійснення видачі палива, обліку кілометражу та використаного палива. Документами предметної області, що використовуються як основне джерело інформації АІС – це „Подорожний лист” (рис. 1), що заповнюється начальником гаражу та водієм.

Подорожній лист – це документ, що визначає разом з товарно-транспортною накладною під час перевезення товарних вантажів показники для обліку роботи рухливого складу і водія, а також для нарахування заробітної плати водієві і здійснення розрахунків за перевезення вантажів. У документі „подорожній лист”, форма якого встановлена ДержКомСтатом України агрегується інформація про:

- 1) автомобілі, причепі, колони, бригади, атрибутами яких є: марка, номер державної реєстрації, номер колони, номер бригади, номер державної реєстрації причепа, марка причепа, гаражні номери;
- 2) водії (ПШБ, табельний номер), особи що супроводжують (ПШБ, табельний номер);
- 3) графіки-завдання для виконання (час оправлення, час прибуття, пробіг, показання спідометра);
- 4) паливо (марка, залишок при виїзді, залишок при прибутті);
- 5) товари (назва товару, відкіля взяти, куди привезти, у тонах);

ЗАТВЕРДЖЕНО
 наказом Мінітрансу, Міністату України
 від 29 грудня 1995 р. N 488/346
 Типова форма N 2-ТН

Подорожній лист N _____ вантажного автомобіля
 " _____ " _____ 20__ р.

Місце для підпису підприємства		Ремонт роботи		Код	
		Колоса	Бригада	Гар. N	Бригада
Автомобіль	марка _____	держ. N _____	Гар. N _____		
Водій	прізвище, ім'я, по батькові _____	Таб. N _____	Гар. N _____		
Привіт 1	марка _____	держ. N _____	Гар. N _____		
Привіт 2	марка _____	держ. N _____	Гар. N _____		
Супроводжуючі особи					
Робота вантажів та автомобіля					
операция	час за графіком год.	хв.	пул. пробіг. км	показ спідометра	час, фактичний час, мін., год., хв.
1	2	3	4	5	6
вїзд із гаража					
поверн. в гараж					
Рух вантажів, літріє					
марок вантажів	код марок	визначено	завишок при вїзді	повернений спеціаліст.	час роб. год.
7	8	9	10	11	12
дефекти					
заправок					
літрів					
марок					
дефекти					

в час розпорядження		час		кількість		куди доставити		найменування		кільк. ілок з		перевезти	
		прибуття	вїбуття	годин	вантаж	вантаж	вантаж	вантаж	вантаж	вантаж	тонн		
14	15	16	17	18	19	20	21	22	23				
Завантаження вантажів													
Разом													

Посвідчення водія, перевіряє, завдання вказує, _____
 водія вантажного автомобіля _____ літріє _____
 Підпис диспетчера _____
 Водій за станом здоров'я до управління _____
 допущений підпис _____ штами _____

Водій дозволений, підпис механіка _____
 Автомобіль прийняв, підпис водія _____
 При поверненні автомобіль справний/ не справний _____
 Здав водій _____
 Підпис механіка _____

Особливі відмітки _____

a)

Послідовність виконання завдань							
NN ідекс	номер приставки товарно-транспортних накладних і талонів заповника	кід працьовано, год., хв.	перевезено, тонн	виконано, ткм	підпис та печатка вантажовідправника	маршрут руху (заповнюється заповником)	
						відси	куди
24	25	26	27	28	29	30	31

ТТН у кількості _____ шт. Здав водій _____ Приймав диспетчер _____
прописом

ТАКСУВАННЯ _____

±

Результати роботи автомобіля і прилепі																				
внтрага пального (літрів)		час в марші, год., хв.					кількість ідекс з вантажем	пробіг, км				перевезено		виконано		зарплата				
		всього		в т.ч. автомобіля				загальний		в т.ч. з вантажем		тонни		ткм		код	сума			
за нормою	фактично	автомобіля	прилепі	у руці	у простій		автомобіля	прилепі	автомобіля	прилепі	всього	в т.ч. в прилепах	всього	в т.ч. в прилепах	48			49		
32	33	34	35	36	на лінії	по тех. неспр.										37	38		39	40
Коды марок автомобіля				примечіє				Автомобіле-дні у роботі												

б)

Рисунок 1.1 – Форма документу Подорожній лист:
а – перша сторінка; б – друга сторінка

б) шляхи проходження (у ходок, кілометраж).

Оскільки в рамках поставленої задачі вирішується завдання обліку палива машинами дільниці заводу, то структура документа „Подорожній лист” значно спрощується і в остаточному своєму вигляді містить інформації стосовно:

- 1) техніки дільниці;
- 2) водіїв (ПІБ, табельний номер);
- 3) графіки-завдання для виконання (час оправлення, час прибуття, пробіг, показання спідометра);
- 4) паливо (марка, залишок при виїзді, залишок при прибутті);
- 5) шляхи проходження (у ходок, кілометраж).

Оскільки в предметній області обрано лише один документ, який може бути застосований в якості вхідного документу АІС, то схема документообігу не наводиться.

1.3 Функціональна модель предметної області

Ключовим моментом створення АІС з метою автоматизації інформаційних процесів організації є аналіз функціональної взаємодії об'єктів автоматизації [2]. Щоб зрозуміти, як функціонує предметна область, виконується її функціональний аналіз. В основі функціонального аналізу лежить принцип декомпозиції дій. Синонімами поняття „дія” є: процес, задача, функція. Результатом аналізу в цьому контексті є функціональна модель, що дає представлення про предметну область у термінах функцій і груп даних, що супроводжують виконання цих функцій. Результати такого аналізу багатогранні. Системні аналітики представляють їх у вигляді функціональної моделі предметної області. У найбільш узагальненому виді, функціональна модель – це опис предметної області, заснований на аналізі семантики об'єктів і явищ, виконане без орієнтації на використання програмних або технічних комп'ютерних засобів, у якій акцентується функціональний аспект моделювання предметної області [3].

В якості засобу розробки функціональної моделі предметної області обрано методологія IDEF0, яка передбачає представлення результатів аналізу предметної області у вигляді діаграм: контекстної діаграми, діаграм декомпозиції та діаграми дерева робіт. форми вимог і специфікацій), змодельована в одній із графічних нотацій або з застосуванням формальних мов моделювання.

Контекстна діаграма. Модель IDEF0 завжди починається з представлення системи як єдиного цілого – одного функціонального блоку з інтерфейсними дугами, що простираються за межі розглянутої області. Така діаграма з одним функціональним блоком називається контекстною діаграмою.

Визначаємося зі складовими частинами контекстної діаграми. У контекстній діаграмі вхід – паливо; керування – норми витрати палива, правила оформлення подорожніх листів; механізм – водії та операціоніст; вихід – заповнений подорожній лист, визначена перевитрата палива. Завершений вигляд контекстної діаграми має вигляд, як показано на рисунку 1.2.

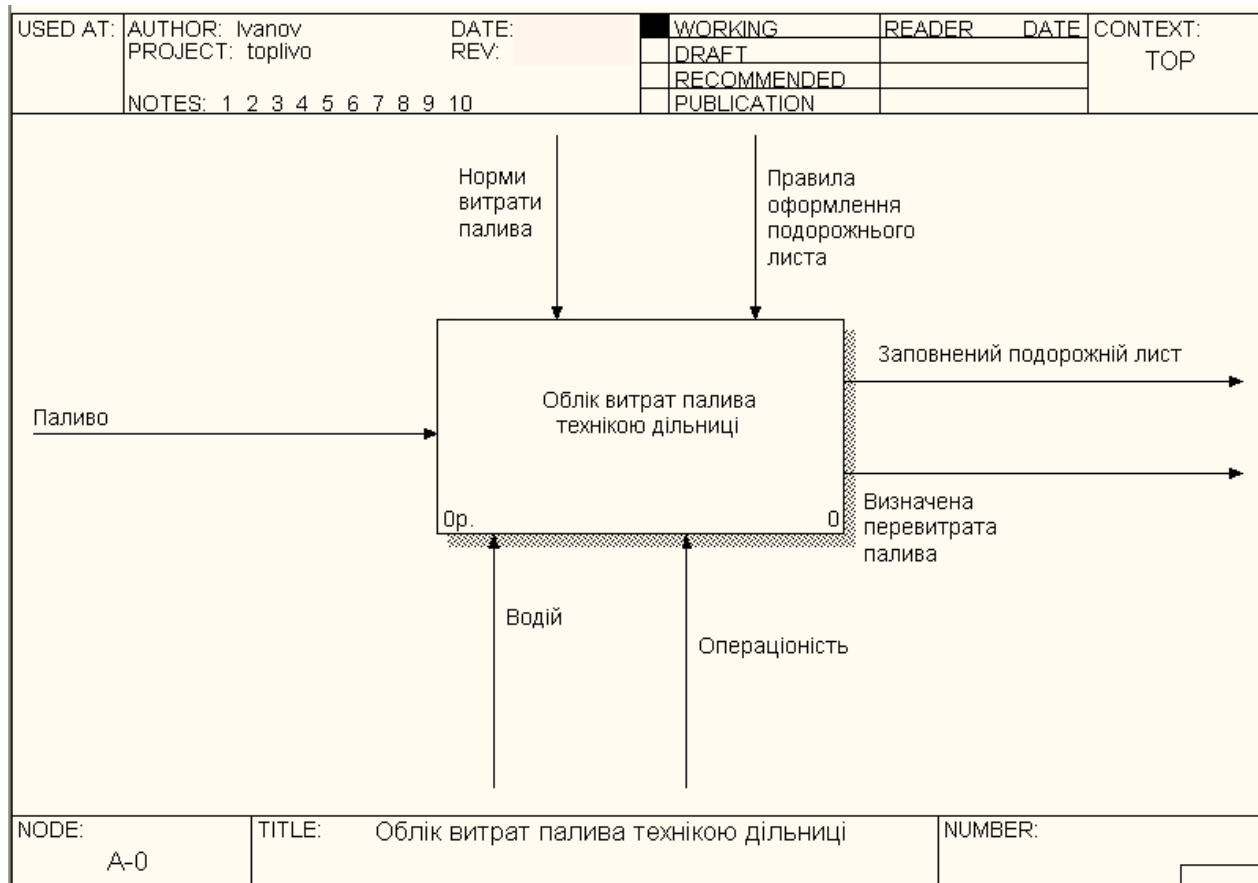


Рисунок 1.2 – Контекстна діаграма предметної області

Діаграми декомпозиції першого рівня. Кожна робота декомпозується на елементи наступного рівня, і так відбувається доти, поки не буде отримана релевантна структура, що дозволяє відповісти на питання, сформульовані в меті створення функціональної моделі предметної області. Кожна підфункція моделюється окремим блоком. Кожен батьківський блок докладно описується дочірньою діаграмою на більш низькому рівні. Діаграма декомпозиції першого рівня показана на рис. 1.3. У випадку, коли дії в предметній області складні та містять набір операцій, рекомендується створювати діаграми декомпозиції другого рівня.

Для функціонального блоку 3 та функціонального блоку 4 необхідно створити діаграми декомпозиції другого рівня (рис. 1.4 та рис. 1.5 відповідно) для опису складових частин цих дій.

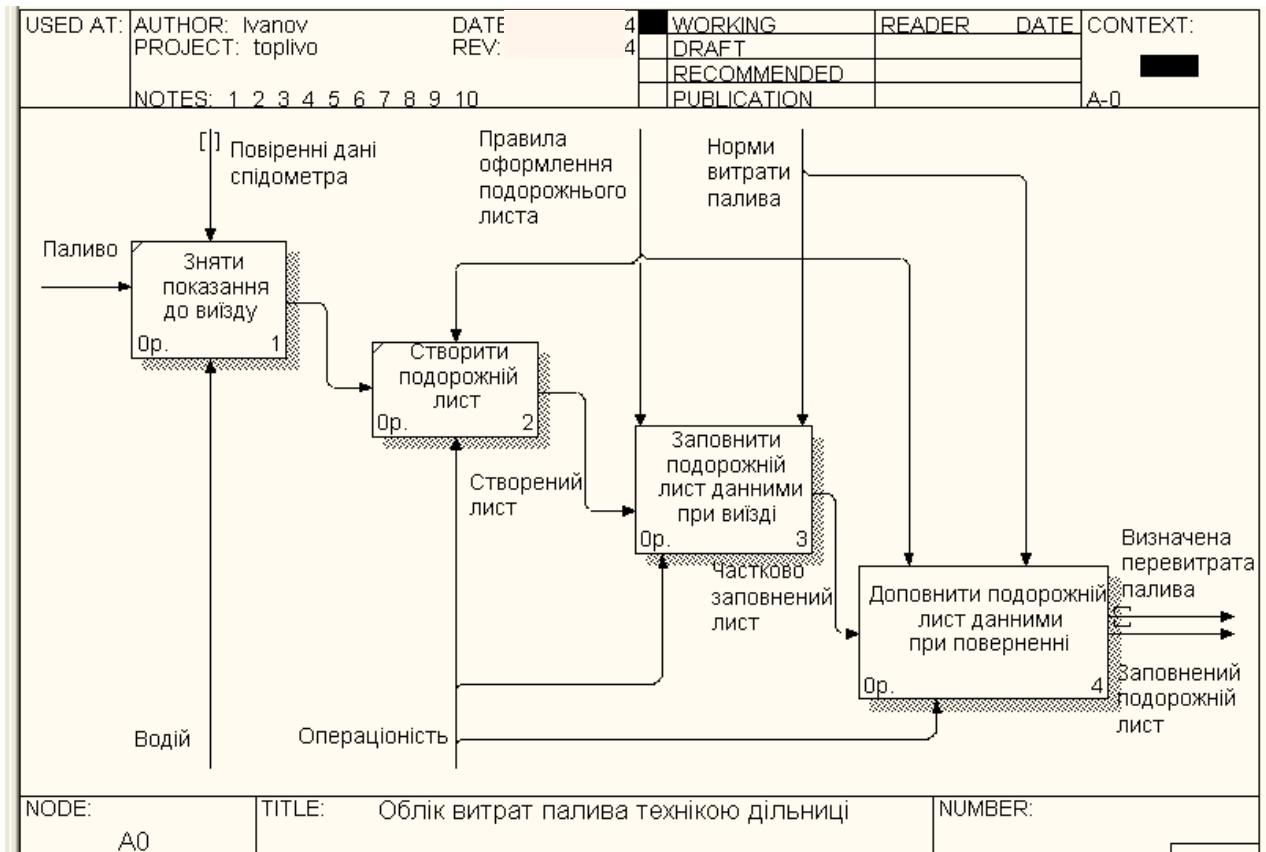


Рисунок 1.3 – Діаграма декомпозиції першого рівня предметної області

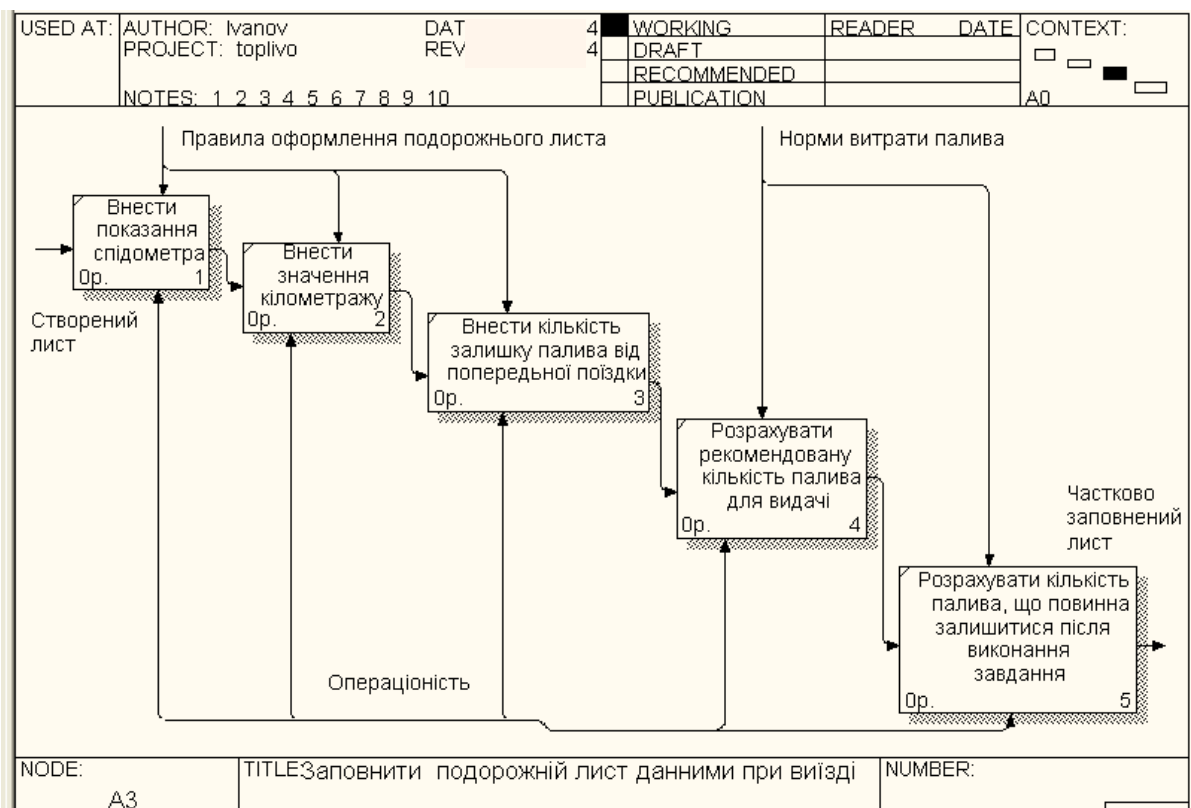


Рисунок 1.4 – Діаграма декомпозиції другого рівня 3-го функціонального блоку

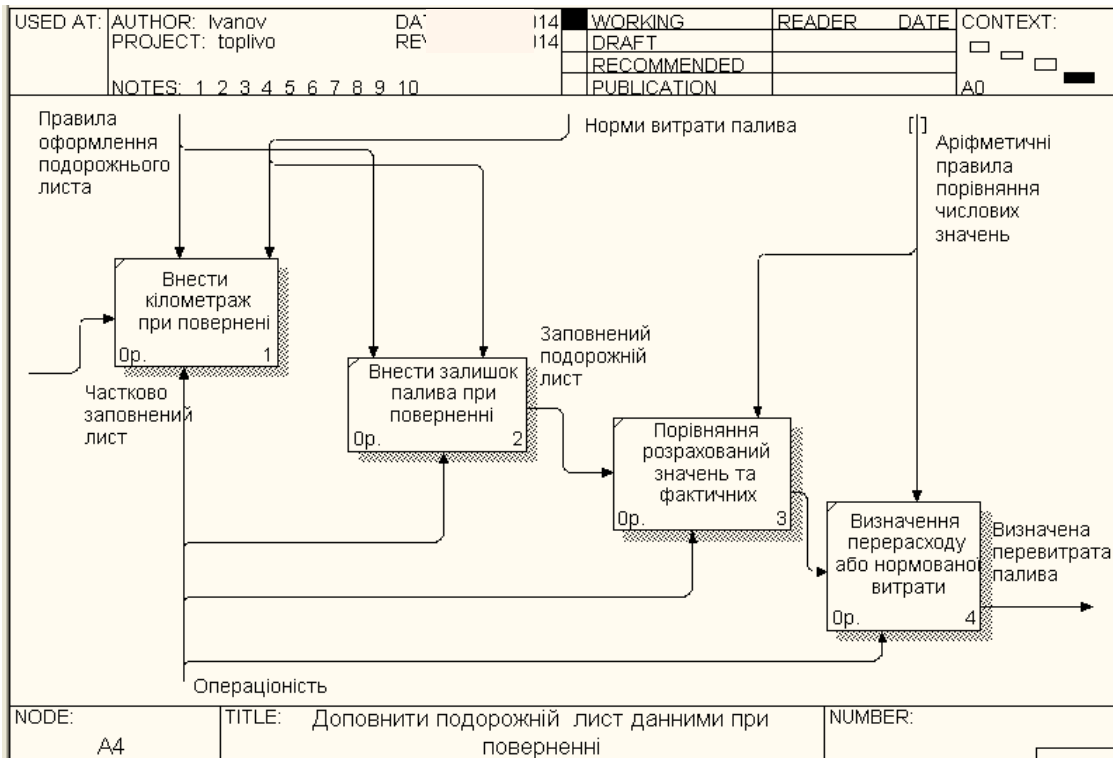


Рисунок 1.5 – Діаграма декомпозиції другого рівня 4-го функціонального блоку

Побудова дерева робіт. Діаграма дерева вузлів (рис. 1.6) показує ієрархічну залежність робіт предметної області.

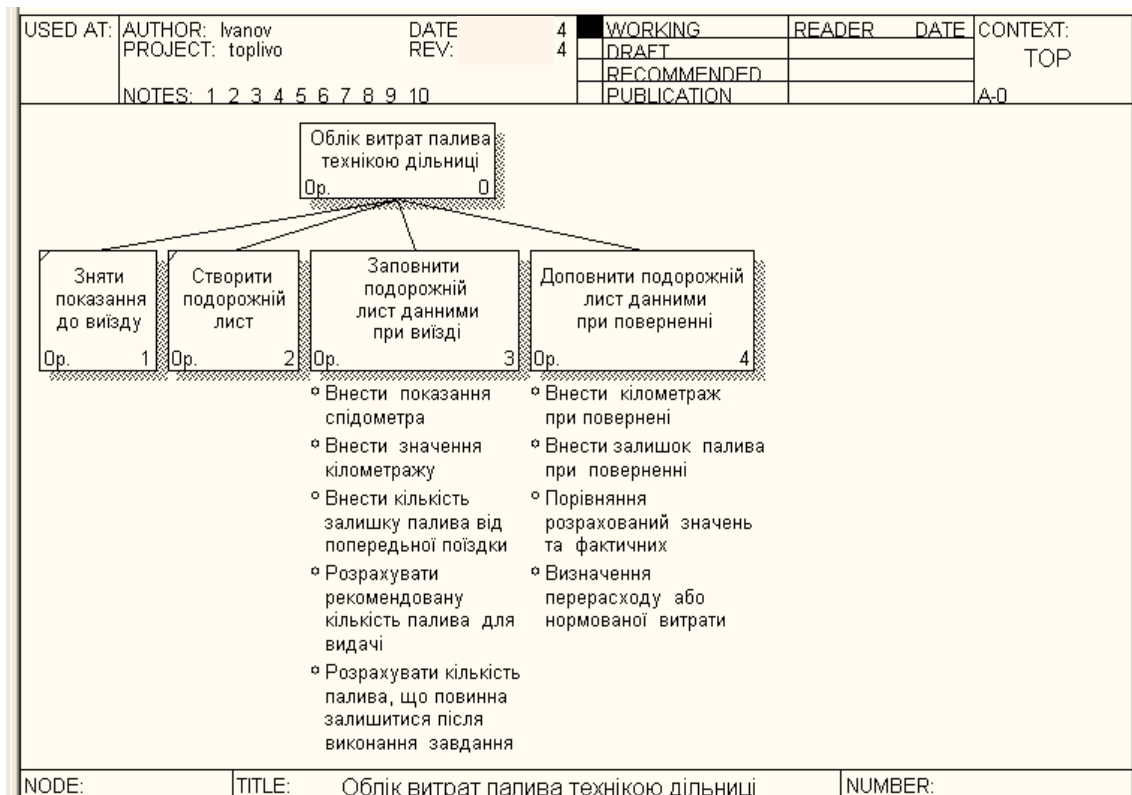


Рисунок 1.6 – Діаграма дерева робіт предметної області

1.4 Об'єктна модель предметної області

Об'єкт предметної області – це сутність, що використовується при виконанні деякої функції або операції (перетворення, обробки, формування і т.д.). Об'єктна модель предметної області – це відображення статичної структури її складових частин [4]. Об'єкта модель повинна відображати набір сутностей (об'єктів) предметної області, набір їх властивостей, набір їх структурних зв'язків та зв'язків відношення між ними. Об'єкти предметної області можуть мати динамічну або статичну природу: динамічні об'єкти використовуються в одному циклі відтворення, наприклад замовлення на продукцію, рахунки на оплату, платежі; статичні об'єкти використовуються в багатьох циклах відтворення, наприклад, устаткування, персонал, запаси матеріалів.

Використовуючи об'єктно-орієнтований метод аналізу предметної області були виділені статичні та динамічні об'єкти предметної області їх атрибути, взаємозв'язки і структурні особливості. В якості інструментарію відображення об'єктної моделі предметної області обрано діаграму сутність-зв'язок. Враховуючи вимоги розробки діаграми сутність-зв'язок необхідно описати специфікацію об'єктів (табл. 1.1) предметної області, специфікацію атрибутів сутностей (табл. 1.2) та специфікацію зв'язків між об'єктами (табл. 1.3).

Статичними об'єктами предметної області є техніка дільниці, паливо та водії дільниці. Техніка дільниці – це будь-який транспорт, що закріплений за дільницею. До кожної одиниці техніки закріплено 2 водія. У рамках розглядуваної предметної області, об'єкт „техніка дільниці” представляється в вигляді 2-рівневої агрегації (агрегація формує об'єкт як зв'язок між іншими об'єктами) об'єктів.

Таблиця 1.1 – Специфікація сутностей предметної області

№	Назва сутності	Описання сутності
1	Тип техніки	Статична сутність, що задає групи для класифікації всієї техніки, що є на дільниці виробничого підприємства
2	Техніка	Статична сутність, яка описує будь-який транспорт, що закріплений за дільницею
3	Палива	Статична сутність предметної області, що описує види палива, що використовуються технікою на дільниці
4	Водій	Статична сутність, яка відображає інформацію стосовно кожного працівника-водія співробітник, який закріплюється за машиною дільниці. Саме на водія

		виконується видача палива, з урахування нормативної витрати палива для відповідної машини. Водій зобов'язаний вірно заповнювати розділи подорожнього листа стосовно кілометражу (за спідометром) та залишок палива за датчиком.
5	Подорожній лист	Динамічна сутність, яка містить дані стосовно кожної господарської операції з паливом, технікою та водієм

Таблиця 1.2 – Специфікація атрибутів сутностей:

Назва сутності	Назва атрибуту	Опис атрибуту
Тип техніки	Назва типу	Символьний тип даних. Задає класифікаційної групи техніки на дільниці
Техніка	Модель	Символьний тип даних. Задає назву або номер моделі техніки
	Марка	Символьний тип даних. Задає назву марки техніки.
	Рік випуску	Числовий тип даних. Задає значення року випуску техніки
	Номер держреєстрації	Символьний тип даних. Задає символну послідовність номеру державної реєстрації техніки
	Норма витрати за паспортом	Числовий тип даних. Задає нормовану кількість витрати палива в змішаному режимі роботи техніки із розрахунку на 100 км.
Паливо	Назва типу палива	Символьний тип даних. Задає назву класифікаційної групи палива, що використовується на дільницю підприємства
	Марка палива	Символьний тип даних. Задає назву або номер марки палива
Водій	ПІБ	Символьний тип даних. Об'єднана характеристика водія у форматі Прізвище, Ім'я, По батькові
	Табельний номер	Числовий тип даних. Задає значення унікального номера кожного співробітника підприємства
Подорожній лист	Номер	Чисельне значення. Порядковий номер створюваного документа
	Дата	Тип даних – дата-час. Задає дату створення документу
	Водій	Символьний тип даних. Відображає

<i>Продовження табл. 1.2</i>	
	ПІБ, табельний номер водія
Паливо	Символьний тип даних. Відображає назву типу та марку палива
Техніка	Символьний тип даних. Відображає назву марки, моделі машини, номер державної реєстрації та паспортний розмір витрати палива
Тип техніки	Відображає назву типу техніки
Показання спідометра при виїзді	Чисельне значення.
Кількість виданого палива	Чисельне значення.
Залишок палива при виїзді	Чисельне значення. Відображає значення кількості палива, яке залишилося після попередньої роботи, перед поточною видачею
Залишок при прибутті	Чисельне значення. Відображає значення кількості палива, яке залишилося після виконання поточного завдання
Кілометраж	Чисельне значення. Відображає орієнтовану відстань, що повинна пройти техніка, виконуючи поточне завдання

Таблиця 1.3 – Зв'язки між сутностями

Назва зв'язку	Назва сутності 1	Назва сутності 2	Множинність зв'язку
Містить	Подорожній лист	Тип техніки	1:1
Містить	Подорожній лист	Техніка	1:1
Містить	Подорожній лист	Паливо	1:1
Містить	Подорожній лист	Водій	1:1
Складається	Тип техніки	Техніка	1:N

Діаграма сутність зв'язок оперує наступними графічними нотаціями:

- прямокутник – це сутність предметної області. У прямокутнику зазначається назва сутності (іменник);
- овал – це атрибут сутності. В овалі записується назва атрибута сутності (іменник);
- ромб – це зв'язок між сутностями. В ромбі записується або назва зв'язку між сутностями (змістове дієслово) або порядковий номер. Загальний вигляд діаграми „сутність-зв'язок” для розглядуваної предметної області показано на рис. 1.7.

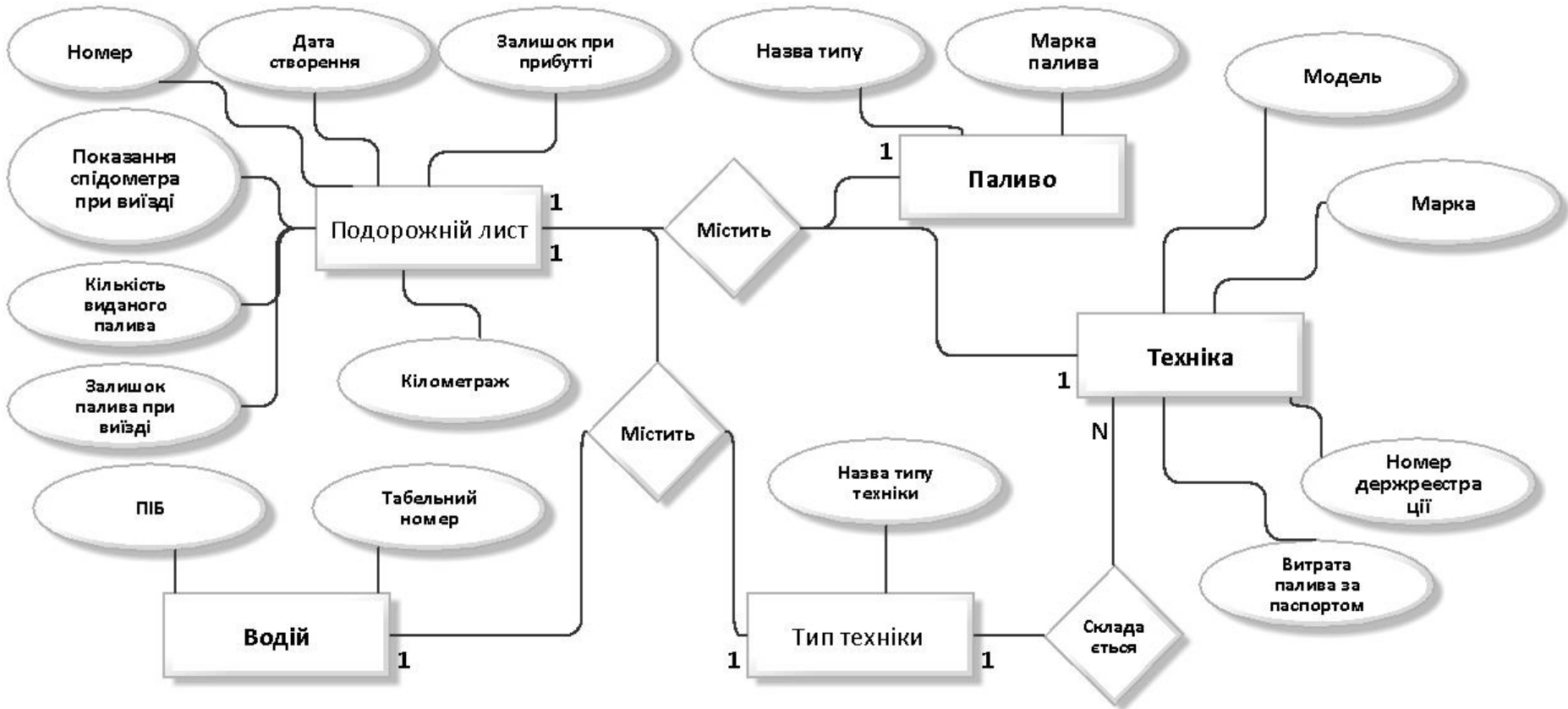


Рисунок 1.7 – Діаграма „сутність-зв’язок ” предметної області

АІС складається з нормативно-довідникової, вхідної та вихідної інформації. Для того, щоб виділити складові частини АІС (нормативно-довідникову, вхідну і вихідну інформацію) необхідно проаналізувати поставлену задачу, відповідаючи на питання:

- яку саме інформацію повинна видавати АІС для рішення конкретної задачі у визначеній предметній області;
- які вхідні параметри запитів до БД (тобто яку інформацію необхідно буде ввести користувачем, щоб одержати результат роботи АІС);
- з яких документів предметної області можна набути необхідну інформацію;
- яку інформацію предметної області можна віднести до умовно незмінної, що можна зберегти в нормативно-довідникову інформацію для забезпечення швидкості та достовірності введення даних. Вихідна інформація АІС – це результат виконання динамічних запитів до БД. До складу вихідної інформації АІС відносять задачі, для рішення яких створюється АІС. Вихідна інформація складається з даних 3 видів:

- вхідні параметри запиту до БД – умови, що обмежують пошук і обсяг вибірки даних;
- результати вибірки даних із БД (з таблиць документів БД);
- результати автоматичних розрахунків.

Визначивши склад вихідної інформації потрібно вибрати з документообігу предметної області мінімальний набір документів, що зберігають у собі максимальну кількість даних, придатних для рішення поставленої задачі. Вхідна інформація АІС – це основне джерело даних в БД, зазвичай паперові документи предметної області перетворюють до електронних документів та відносять до пункту меню АІС „Вхідна інформація” і збережені у вигляді таблиць БД.

Умовно-постійні атрибути документів вхідної інформації складають основу для розробки нормативно-довідникової інформації системи. Нормативно-довідникова інформація (НДІ) слугує для збереження умовно незмінних даних сутностей предметної області. Мета використання НДІ:

- скорочення часу для введення даних оператором АІС;
- скорочення помилок при введенні даних;
- забезпечення цілісності даних БД.

При розробці структури довідників приймаються до уваги умови:

- одна класифікаційна група повинна мати не більш, ніж 100 елементів (не більше 100 рядків таблиці);

- структура довідника повинна відповідати структурі об'єкта предметної області;

- зазвичай, один довідник – це опис одного об'єкта предметної області.

АІС „Облік витрат палива технікою на дільниці” повинна містити нормативно-довідникову інформацію (рис. 2.1):

- техніка дільниці;
- паливо;
- водії.

До складу вхідної інформації системи необхідно віднести документ предметної області:

- подорожній лист.

В якості вихідної інформації необхідно представити інформацію стосовно:

- обліку витрат палива по водіям;
- обліку витрат палива по подорожньому листу;
- обліку витрат палива по техніці.

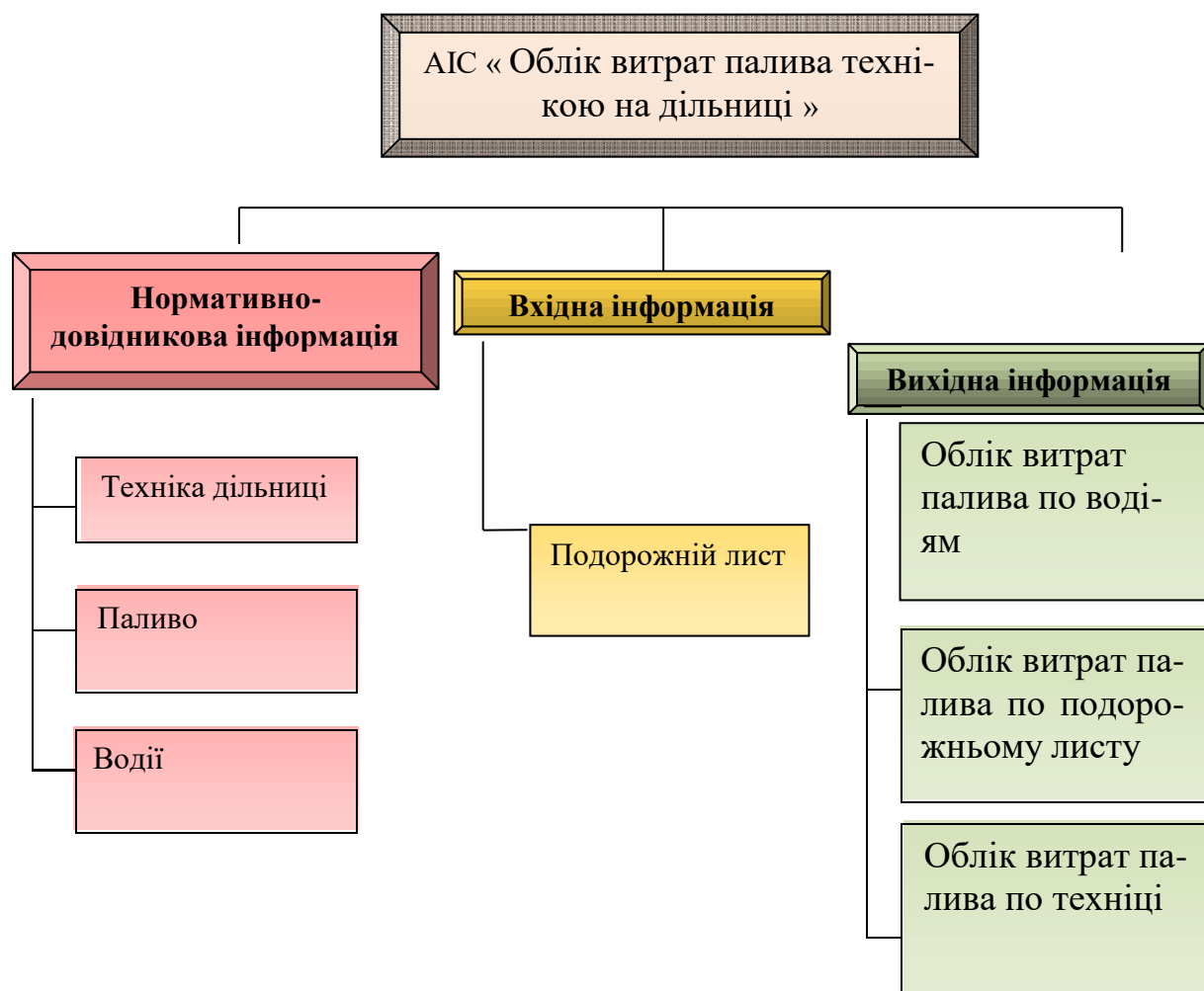


Рисунок 2.1 – Структура меню АІС

Розробка бази даних складається з розробки логічної та фізичної моделі БД. Логічна структура БД – узагальнений, табличний опис сутностей предметної області, зроблений із урахуванням нормалізації даних та незалежних від засобів програмної реалізації БД. Логічна структура бази даних може бути представлена декількома способами, найбільш інформативним з яких є повна атрибутивна модель БД. Фізична модель БД – це схема БД, яка розроблена в рамках обраної СУБД. Оскільки вихідна інформація АІС – це результат вибірок даних із БД, що генерується в залежності від запитів користувача АІС, то в БД вихідна інформація не зберігається.

3.1 Логічна структура БД

Побудова повної атрибутивної моделі БД включає в себе відображення таблиць бази даних у вигляді таблиці, що містить 3 рядки, в яких описуються атрибути сутності:

- природна назва атрибутів сутності;
- синтетична назва атрибутів сутності;
- тип даних та розмір атрибута сутності.

В таблиці виділяються первинні та зовнішні ключі, таблиця отримує природну назву та в дужках записується назва таблиці в БД.

Нормативно-довідникова інформація АІС

Довідник «Техніка дільниці»

Таблиця 3.1 – Логічна структура таблиці БД «Тип техніки» (TipMashun)

Код типу	Назва типу
ID_TIPM	Name_TMash
I4	C50

Таблиця 3.2 – Логічна структура таблиці БД «Техніка» (Mashunu)

Код техніки	Код типу	Номер держреєстрації	Модель	Марка	Витрата палива за паспортом
ID_Mash	ID_TIPM	Nomer	Model	Marka	Rashod
I4	I4	C20	C50	C50	N2,2

Таблиця 3.3 – Логічна структура таблиці бази даних „Паливо” (Toplivo)

Код палива	Назва типу палива	Марка палива
ID_toplivo	Nazv	marka
I4	C30	C20

Таблиця 3.4 – Логічна структура таблиці бази даних „Водії” (Voditeli)

Код водія	Прізвище	Ім'я	По батькові	Табельний номер
Kod_d	Fam	Im	Ot	Tab_nomer
I4	C25	C15	C30	C15

Вхідна інформація

Таблиця 3.5 – Логічна структура таблиці бази даних „Подорожній лист” (pytevoy):

Код	Номер	Дата	Код типу	Код техніки	Код водія
Kod_a	Nomer_a	Data_a	ID_TIPM	ID_Mash	Kod_v
I4	I4	D8	I4	I4	I4
Код палива	Показання спідометра на виїзді	Кілометраж	Кількість виданого палива		
ID_toplivo	Pokaz	Rastoanie	VudelinoTop		
I4	I4	I4	I4		
Залишок палива при виїзді	Залишок при прибутті				
Ostat_v	Ostat_p				
N2,2	N2,2				

3.2 Фізична структура БД

Фізична модель даних представляє собою опис об'єктів предметної області в рамках засобів конкретної СУБД. У фізичній моделі описується вся інформація про конкретні фізичні об'єкти – таблиці, колонки, індекси та збережені процедури (рис. 3.1).

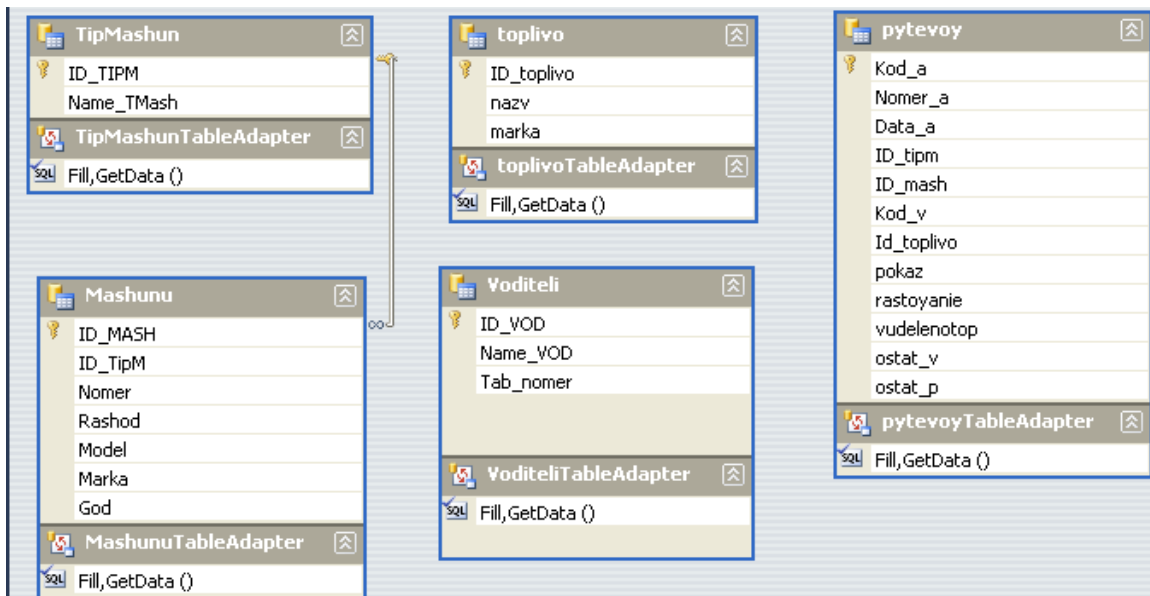


Рисунок 3.1 – Фізична модель БД

3.3 SQL-запити вихідної інформації

Для забезпечення можливості виконання динамічних запитів користувачів систем можна використовувати механізм запитів до представлень бази даних. Представлення є тимчасовими наборами даних, існуючими тільки під час роботи з ними. Важливо розуміти, що представлення є не копією даних, а лише посиланням на реальні записи. Тобто при зміні даних у представленні будуть змінені дані й у фізичній таблиці. І при зміні даних у фізичній таблиці автоматично перебудовуються дані в представленні.

Представлення дозволяють повертати набори даних, що задовольняють запитам конкретних користувачів або груп. Після створення представлення до нього можна звертатися точно так само, як і до звичайної таблиці. Представлення може будуватися на базі однієї або декількох таблиць, або навіть на основі інших представлень.

1. Представлення „Облік витрат палива по водіям”. Схема набору таблиць та полів представлення „Облік витрат палива по водіям” наведено на рис. 3.2.

SQL-запит для формування вибірки даних на основі представлення „Облік палива по водіям”:

```
SELECT SUM(dbo.pytevoy.rastoyanie) AS Expr1,
SUM(dbo.pytevoy.vudelenotop) AS Expr2, dbo.Voditeli.Name_VOD,
dbo.Voditeli.Tab_nomer, dbo.toplivo.nazv, dbo.toplivo.marka, dbo.pytevoy.Data_a
FROM dbo.pytevoy
INNER JOIN dbo.toplivo ON dbo.pytevoy.Id_toplivo = dbo.toplivo.ID_toplivo
INNER JOIN dbo.Voditeli ON dbo.pytevoy.Kod_v = dbo.Voditeli.ID_VOD
```

GROUP BY dbo.Voditeli.Tab_nomer, dbo.Voditeli.Name_VOD,
 dbo.topливо.nazv, dbo.topливо.marka, dbo.pytevoy.Data_a

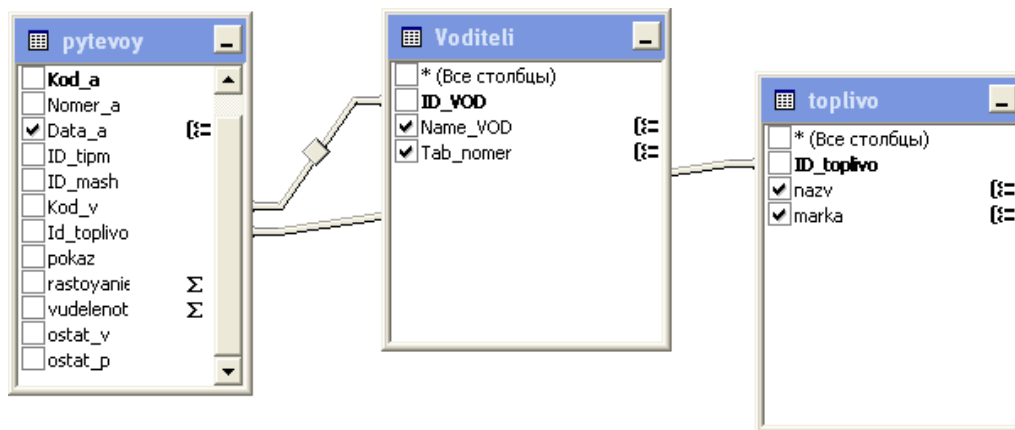


Рисунок 3.2 – Схема представлення „Облік палива по водіям”

2. Представлення „Облік витрат палива по подорожнім листам”. Схема набору таблиць та полів представлення „Облік витрат палива по подорожнім листам” наведено на рис. 3.3.

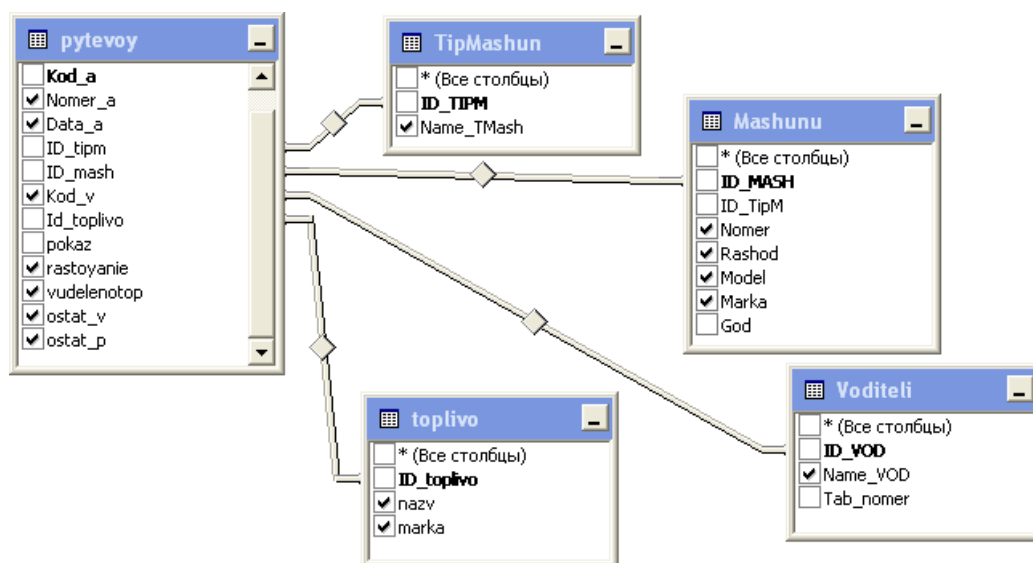


Рисунок 3.3 – Схема представлення „Облік витрат палива по подорожнім листам”

SQL-запит для формування вибірки даних на основі представлення „Облік витрат палива по подорожнім листам”:

```
SELECT dbo.pytevoy.Nomer_a, dbo.pytevoy.Data_a, dbo.pytevoy.rastoyanie,
  dbo.pytevoy.vudelenotop, dbo.pytevoy.ostat_v, dbo.pytevoy.ostat_p,
  dbo.Mashunu.Marka, dbo.Mashunu.Model, dbo.Mashunu.Rashod,
  dbo.Mashunu.Nomer, dbo.Voditeli.Name_VOD, dbo.topливо.nazv,
  dbo.topливо.marka AS Expr1, dbo.TipMashun.Name_TMash, dbo.pytevoy.Kod_v
```

```

FROM dbo.pytevoy
INNER JOIN dbo.toplivo ON dbo.pytevoy.Id_toplivo = dbo.toplivo.ID_toplivo
INNER JOIN dbo.Voditeli ON dbo.pytevoy.Kod_v = dbo.Voditeli.ID_VOD
INNER JOIN dbo.TipMashun ON dbo.pytevoy.ID_tipm =
dbo.TipMashun.ID_TIPM
INNER JOIN dbo.Mashunu ON dbo.pytevoy.ID_mash =
dbo.Mashunu.ID_MASH

```

3. Представлення „Облік витрат палива по техніці”. Схема набору таблиць та полів представлення „Облік витрат палива по техніці” наведено на рис. 3.4.

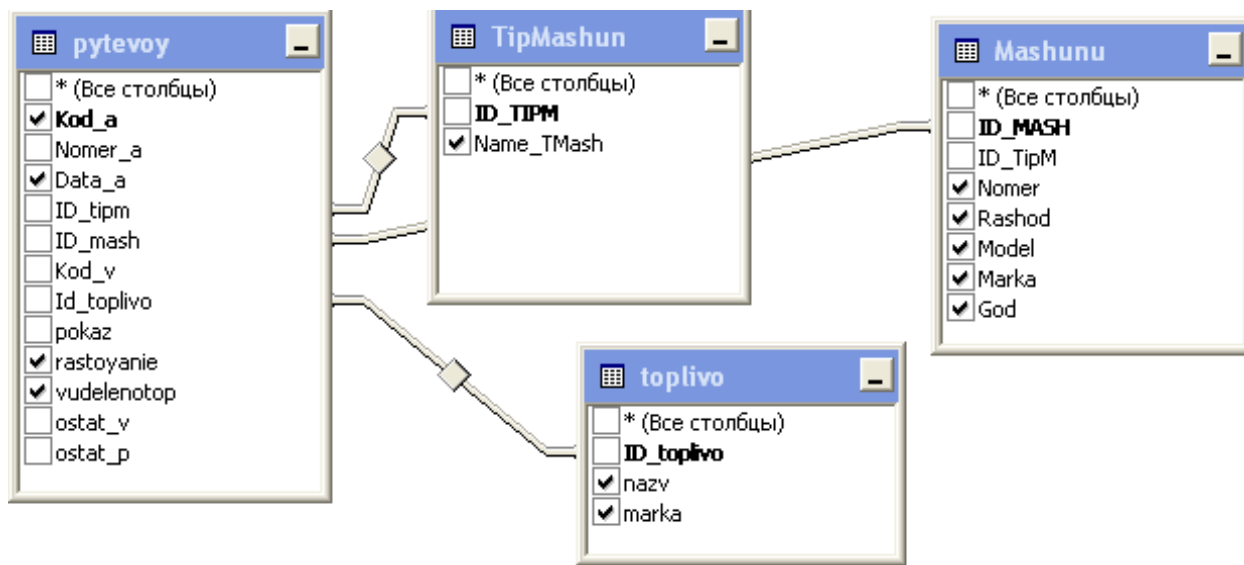


Рисунок 3.4 – Схема представлення „Облік палива по техніці”

SQL-запит для формування вибірки даних на основі представлення „Облік витрат палива по техніці”:

```

SELECT dbo.pytevoy.Data_a, dbo.toplivo.nazv, dbo.toplivo.marka,
dbo.Mashunu.Nomer, dbo.Mashunu.Rashod, dbo.Mashunu.Model,
dbo.Mashunu.Marka AS Expr1, dbo.Mashunu.God, dbo.TipMashun.Name_TMash,
dbo.pytevoy.rastoyanie AS Expr2, dbo.pytevoy.vudelenotop AS Expr3,
dbo.pytevoy.Kod_a AS Expr4
FROM dbo.TipMashun
INNER JOIN dbo.pytevoy
INNER JOIN dbo.toplivo ON dbo.pytevoy.Id_toplivo = dbo.toplivo.ID_toplivo
ON dbo.TipMashun.ID_TIPM = dbo.pytevoy.ID_tipm
INNER JOIN dbo.Mashunu ON dbo.pytevoy.ID_mash =
dbo.Mashunu.ID_MASH

```

Лістинг створення БД наведено у додатку А.

Інтерфейс користувача (UI – user interface) – сукупність засобів, за допомогою яких користувач спілкується із системою, поєднує в собі всі елементи і компоненти програми, що здатні впливати на взаємодію користувача з програмним забезпеченням.

4.1 Профілі користувачів

Розподіл прав до даних можна здійснити двома способами:

- на рівні БД;
- на рівні додатку.

В даній курсовій роботі реалізовано розподіл прав до даних на рівні додатку. В рамках предметної області визначено 3 типи користувачів із різним рівень доступу до даних.

1. Адміністратор (А) – особа з найвищим рівнем доступу до даних, якій надано можливість додавати, видаляти, редагувати та шукати дані з БД в рамках АІС.

2. Менеджер (М) – особа з середнім рівнем доступу до даних, якій надано можливість здійснювати додавання, редагування та пошук даних з БД в рамках АІС.

3. Оператор (О) – особа з низьким рівнем доступу до даних, які надано можливість лише додавати дані до БД.

Операції з даними в рамках кожної екранної форми наведено в табл. 4.1.

Таблиця 4.1 – Визначення прав доступу до даних по видах користувачів

Програмна назва сторінки	Опис функціонального призначення сторінки	Доступ для користувачів		
		А	М	О
vod2.cs	Прогляд	+	+	+
	Додавання даних	+	+	+
	Редагування даних	+	+	+
	Видалення даних	+	-	-
	Пошук даних	+	-	-
top1.cs	Прогляд	+	+	+
	Додавання даних	+	+	+
	Редагування даних	+	+	+
	Видалення даних	+	-	-
	Пошук даних	+	-	-
tex.cs	Прогляд	+	+	+

		<i>Продовження табл. 4.1</i>		
	Додавання даних	+	+	+
	Редагування даних	+	+	+
	Видалення даних	+	-	-
	Пошук даних	+	-	-
pyt_gl.cs	Прогляд	+	+	+
	Видалення	+	-	-
	Пошук	+	+	-
pyt_dob.cs	Прогляд	+	+	+
	Додавання	+	+	+
Rez1.cs	Прогляд	+	+	-
	Формування та отримання результатів виборок	+	+	-
Rez2.cs	Прогляд	+	+	-
	Формування та отримання результатів виборок	+	+	-
Rez3.cs	Прогляд	+		-
	Формування та отримання результатів виборок	+		-

4.2 Екранні форми АІС

Екранні форми призначені для того, щоб полегшити користувачеві введення даних у базу і як додаткову інформацію можуть містити тексти допомоги, що відповідають на питання, як ввести дані в деяке поле, а також зразки заповнення полів. Форми повинні бути доступними для огляду й наочними, в іншому випадку втрачається їхній сенс. Структура інтерфейсної частини вхідної інформації системи показана в таблиці 4.2.

Таблиця 4.2 – Структура інтерфейсної частини вихідної інформації системи

Екрана форма	Опис
vod2.cs	Екранна форма довідника „Водії”
topl.cs	Екранна форма довідника „Паливо”
tex.cs	Екранна форма довідника „Техніка”
pyt_gl.cs	Екранна форма перегляду документів „Подорожній лист”, що зберігаються в базі даних та їх пошуку
pyt_dob.cs	Екранна форма додавання „Подорожнього листа”
Rez1.cs	Екранна форма „Облік витрат палива по водіям”
Rez2.cs	Екранна форма „Облік витрат палива по подорожнім листам”

Rez3.cs	Екранна форма довідника „Облік витрат палива по техніці”
	Системне повідомлення про помилку введення дат запиту

Екранні форми нормативно-довідникової та вхідної інформації представлено на рис. 4.2 – рис. 4.5.

Для додавання, оновлення, видалення та пошуку даних із складових частин довідникової або вхідної інформації користувачам необхідно вибрати відповідний пункт головного меню системи (рис. 4.1).



Рисунок 4.1 – Головна екранна форма АІС

Екранні форми складових частин довідників техніки дільниці, палива та водіїв представлено на рис. 4.2 – рис. 4.5. Діалог між користувачем та системою має аналогічну схему в рамках кожної екранної форми і складається з наступних дій:

- для додавання нового значення до бази даних необхідно ввести у текстові поля, розташовані на екранних формах відповідні значення та натиснути кнопку „Додати” та „Зберегти”, внесені дані будуть відображені в табличній частині форми, а поля для введення будуть автоматично очищені;

- для видалення значення з бази даних необхідно вибрати в таблиці відповідний запис та натиснути кнопку „Видалити”, в результаті користувачеві видасться повідомлення про підтвердження операції видалення (наприклад, рис. 4.4).

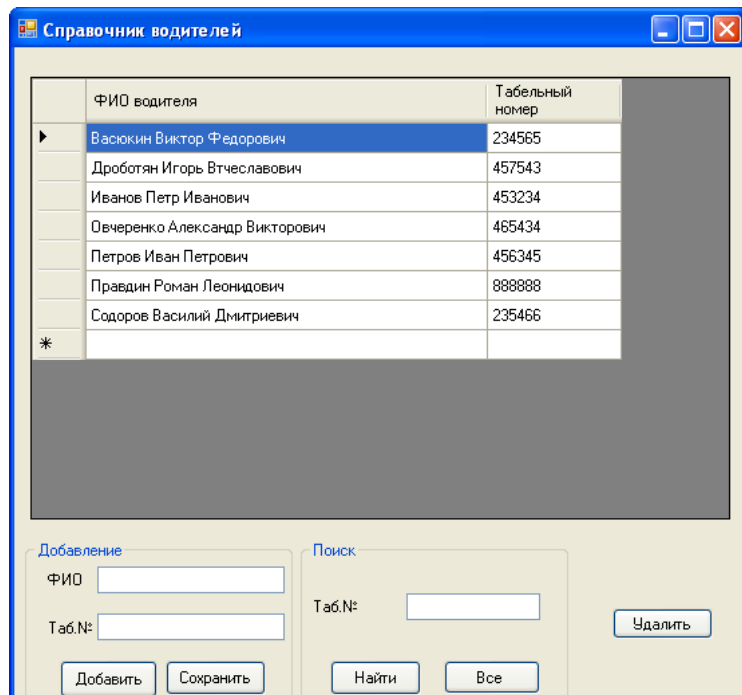


Рисунок 4.2 – Экранна форма довідника „Водії”

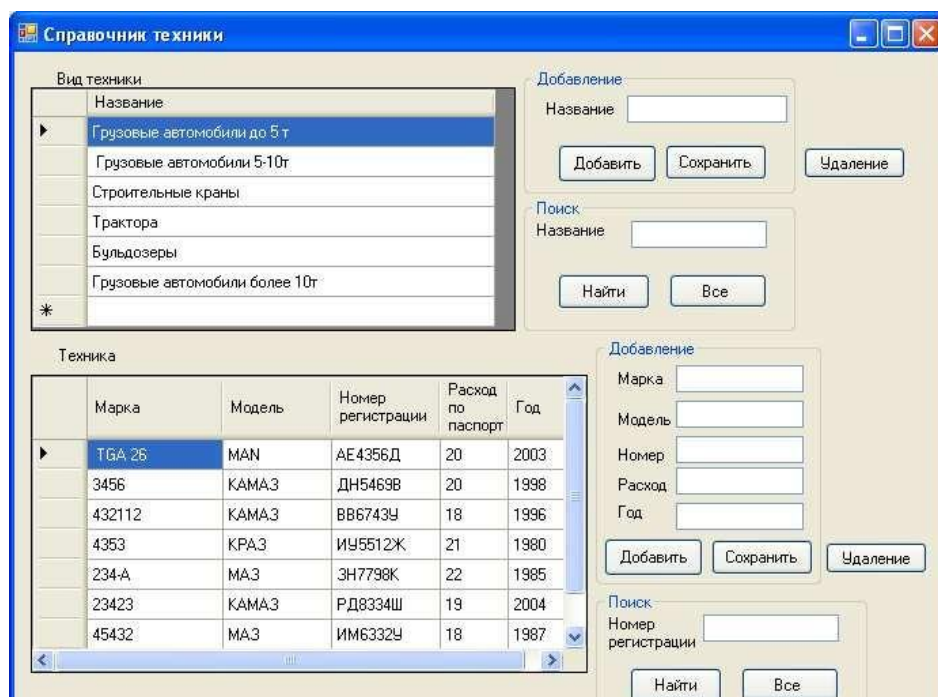


Рисунок 4.3 – Экранна форма довідника „Техніка діляниці”

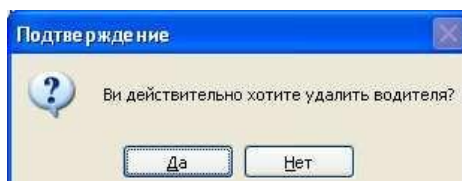


Рисунок 4.4 – Системне повідомлення для підтвердження видалення

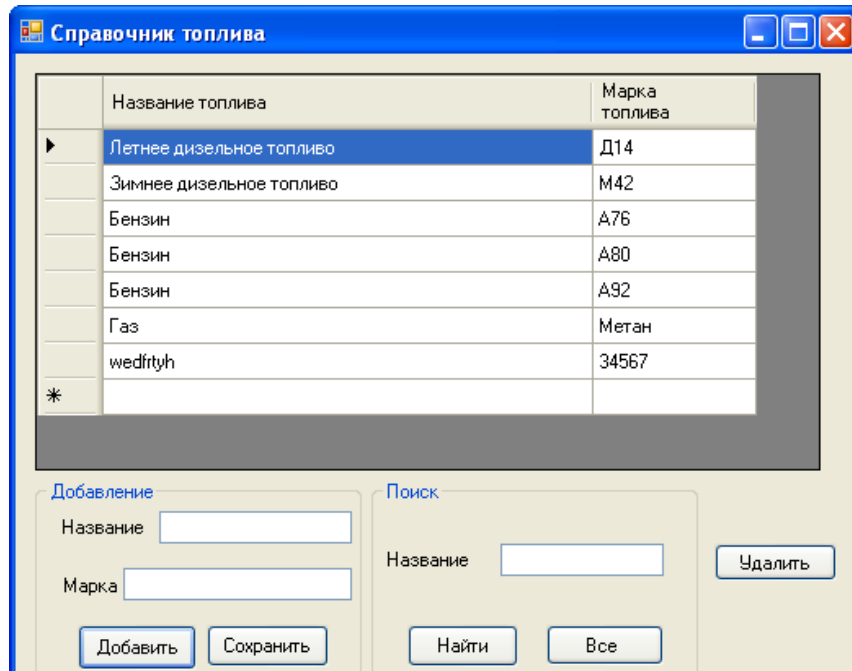


Рисунок 4.5 – Екранна форма довідника „Паливо”

- для редагування запису, що міститься в базі даних необхідно внести зміни безпосередньо у табличну частину форми та натиснути кнопку „Зберегти”;

- для пошуку запису необхідно задати відповідний параметр запиту до бази даних та натиснути кнопку „Пошук”, в результаті в тій самій таблиці буде відображено результат пошуку. Для повернення до всіх записів таблиці довідника необхідно натиснути кнопку „Всі”.

Екранні форми вхідної інформації системи представлено на рис. 4.6 – 4.10. Екранні форми вхідної інформації системи створено з урахуванням вимог уніфікації та стандартизації. Діалог між користувачем та системою має аналогічну схеми в рамках кожної екранної форми. Діалог користувача з системою в рамках форми перегляду документів „Подорожній лист” (рис. 4.6) наступний:

- для видалення подорожнього листа, необхідно у відповідній табличній частині екранної форми обрати подорожній лист, що необхідно видалити та натиснути кнопку „Удалить”, після чого користувачеві видасться повідомлення підтвердження видалення (рис. 4.7);

- для здійснення пошуку подорожнього листа із всіх збережених в базі даних користувачеві необхідно внести номер подорожнього листа у відповідне текстове поле та натиснути кнопку „Знайти”. Для повернення до всіх записів бази даних необхідно натиснути кнопку „Всі”.

Путевые листы в базе данных

Номер	Дата	Тип техники	Марка	Модель	ФИО водителя	Название топлива	Топ.	Пас. рас.	Показа	Расто	Выде/ топли	Остат на выезде	Остат по приез
1	09.05.2013	Грузовые автомобили до ...	TGA 26	MAN	Дроботян Игорь Втчеславович	Летнее ди...	Д14	20	12340	100	20	0	0
3	10.05.2013	Грузовые автомобили до ...	3456	КАМАЗ	Васюкин Виктор Федорович	Летнее ди...	Д14	20	21234	100	20	1	1
21	08.05.2013	Грузовые автомобили 5...	TGA	MAN	Содоров Василий Дмитриевич	Бензин	A76	25	10021	100	30	5	10
12	15.05.201...	Грузовые автомобили 5...	212-M	ЗИЛ	Петров Иван Петрович	Летнее ди...	Д14	32	13245	100	32	5	5
13	15.05.201...	Грузовые автомобили до ...	TGA 26	MAN	Иванов Петр Иванович	Летнее ди...	Д14	20	12334	20	6	2	2
14	15.05.201...	Грузовые автомобили 5...	213-A	МАЗ	Содоров Василий Дмитриевич	Бензин	A76	30	65432	100	30	5	5
3	10.05.201...	Грузовые автомобили 5...	TGA	MAN	Иванов Петр Иванович	Летнее ди...	Д14	25	23452	120	35	5	5
4	22.05.201...	Грузовые автомобили 5...	TGA	MAN	Иванов Петр Иванович	Летнее ди...	Д14	20	43212	100	25	5	8
*													

Создать Удалить Введите номер для поиска: Найти все Сохранить

Рисунок 4.6 – Экранна форма перегляду документів „Подорожній лист”

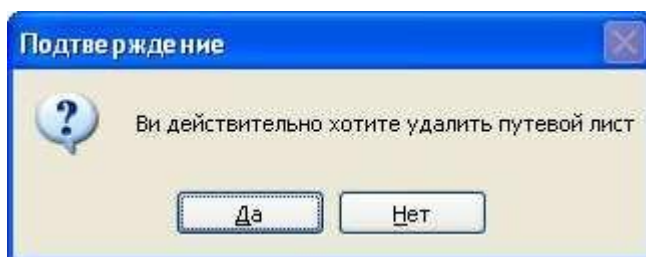


Рисунок 4.7 – Системне повідомлення для підтвердження видалення подорожнього листа

- для додавання нового подорожнього листа необхідно натиснути кнопку „Створити”, після чого відкриється екранна форма (рис. 4.8);
- при поверненні до гаражу в створений подорожній лист необхідно внести значення кількості залишку палива та натиснути кнопку „Зберегти”.

Діалог користувача з системою в рамках форми створення документів „Подорожній лист”:

- для додавання нового подорожнього листа необхідно внести номер, обрати дату створення документу, обрати з довідникової інформації дані стосовно виду техніки, техніки, палива та водія і натиснути кнопку „Створити”. Якщо номер подорожнього листа не введено, то користувачеві видається повідомлення (рис. 4.9).

Якщо всі поля подорожнього листа занесені, то користувачу видається системне повідомлення для підтвердження наміру збереження документу у базі даних (рис. 4.10).

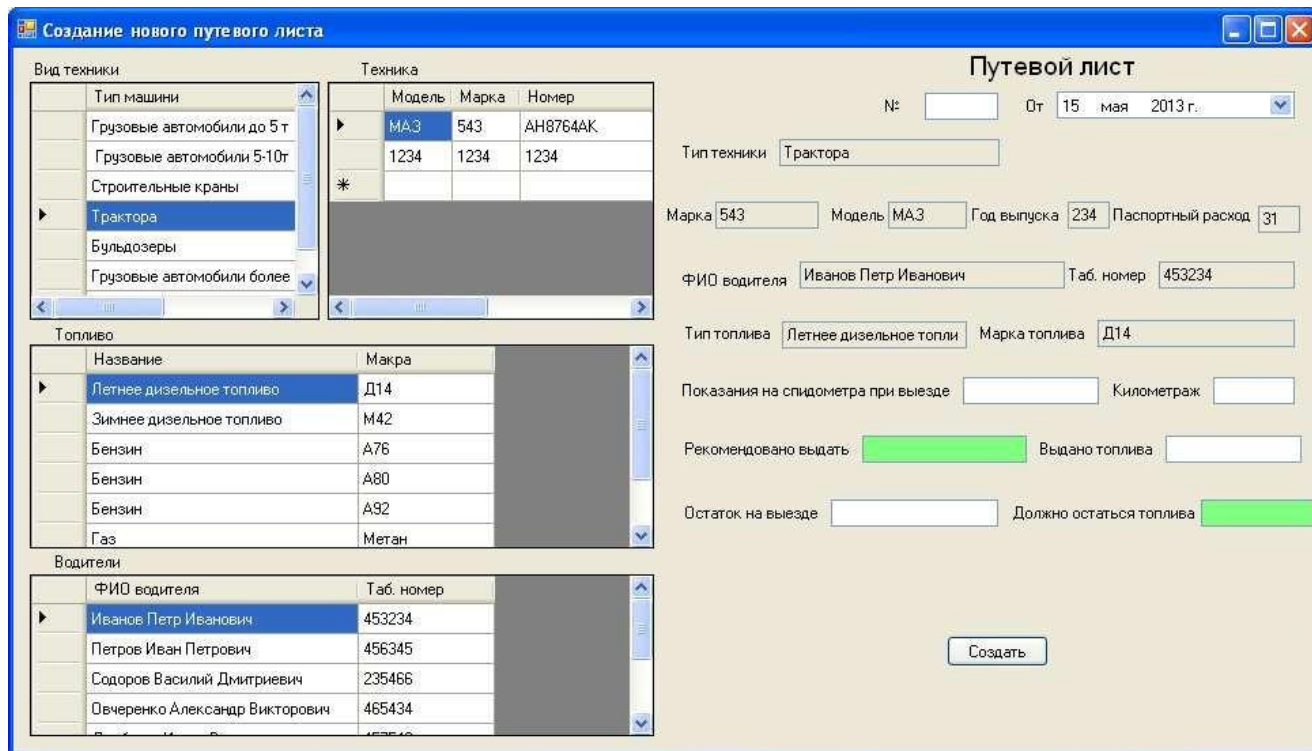


Рисунок 4.8 – Экранна створення документів „Подорожній лист”

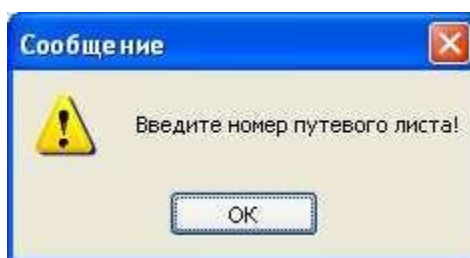


Рисунок 4.9 – Системне повідомлення при спробі зберегти документ без даних

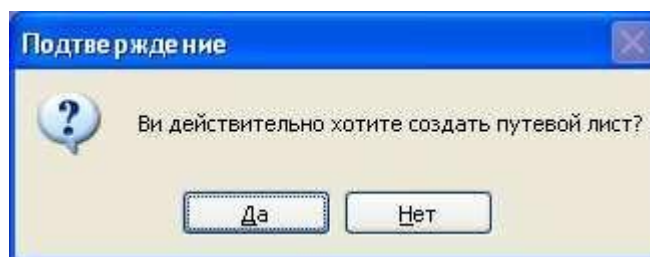


Рисунок 4.10 – Системне повідомлення для підтвердження наміру зберегти створених подорожній лист

Вихідна інформація системи генерується автоматизовано в залежності від уведених параметрів користувача.

В якості параметрів запиту в рамках задачі „Облік витрат палива по водіям” користувач має змогу задати:

- часовий період за допомогою компонента DataTimePicker (рис. 4.11);

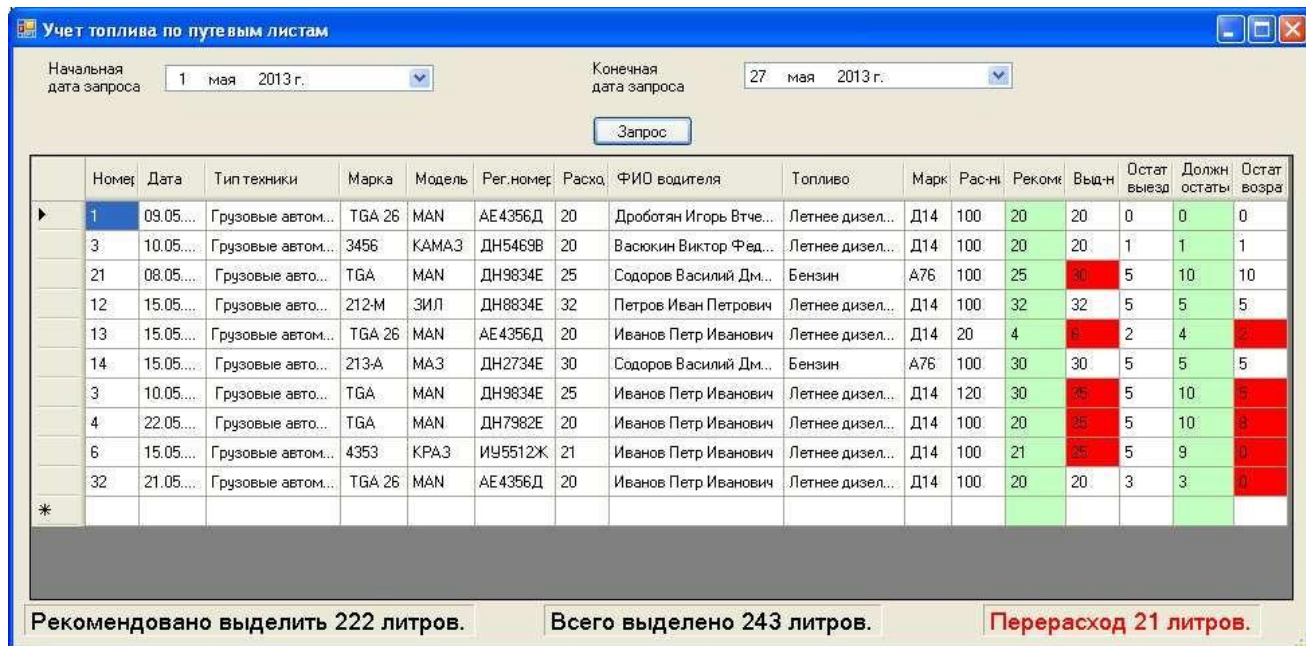


Рисунок 4.15 – Экранна форма „Облік витрат палива по подорожнім листам”

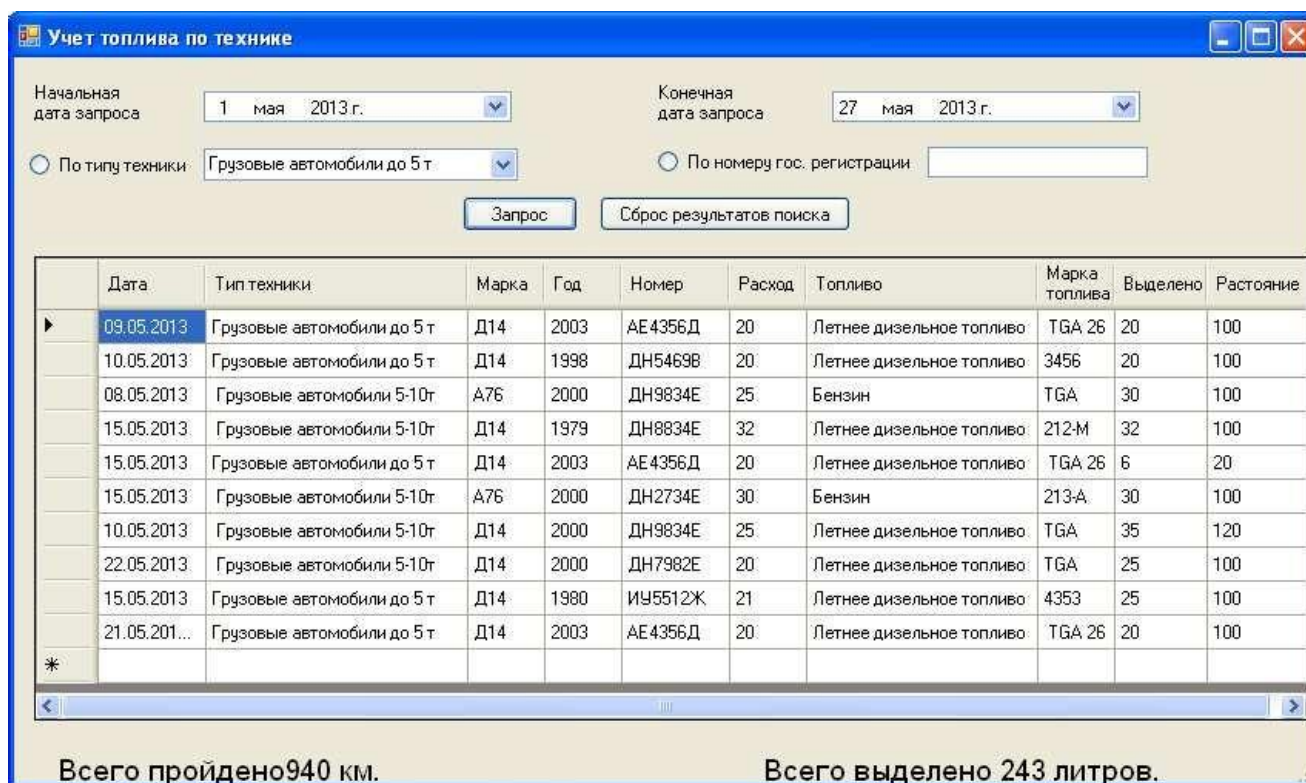


Рисунок 4.16 – Экранна форма „Облік витрат палива по техніці”

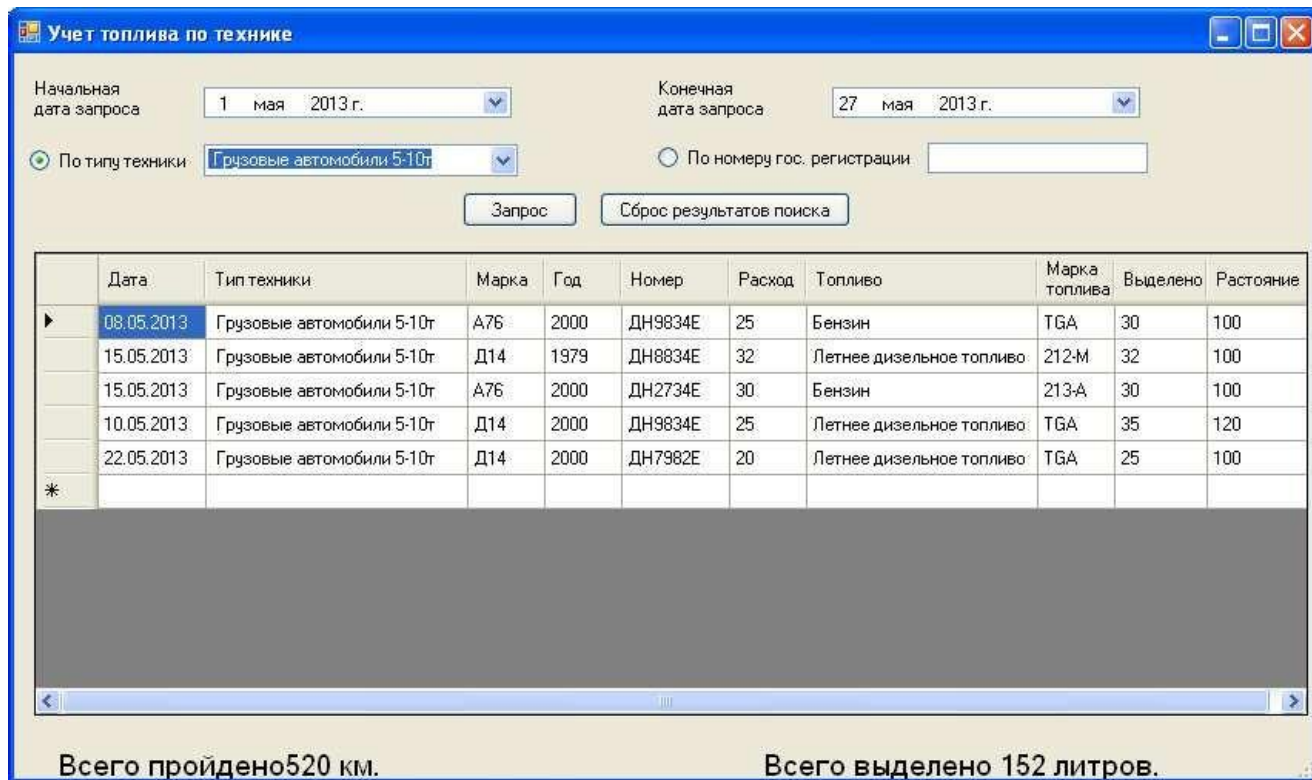


Рисунок 4.17 – Экранна форма „Облік витрат палива по техніці” із заданим входним параметром „Тип техніки”

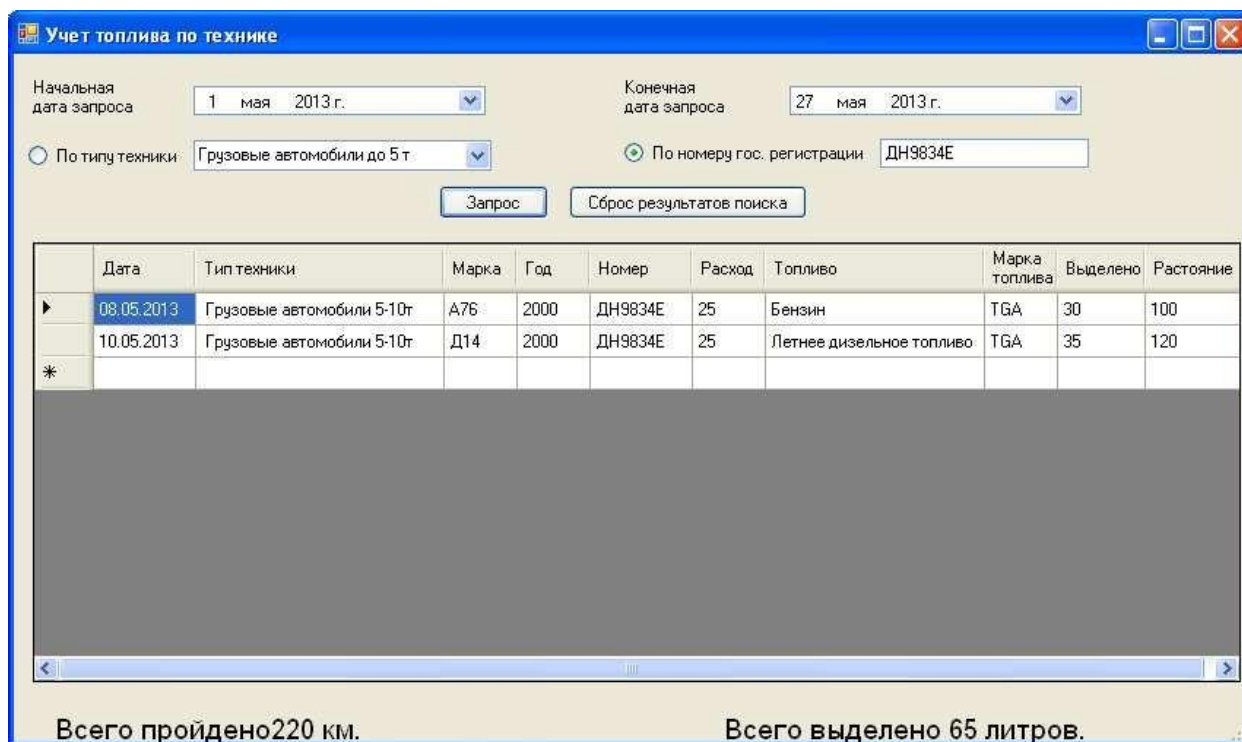


Рисунок 4.18 – Экранна форма „Облік витрат палива по техніці” із заданим входним параметром „Номер держреєстрації”

Лістинг створення пакету прикладних програм наведено у додатку Б.

В якості мови програмування обрано мову C# всередині інтегрованого програмного середовища Microsoft Visual Studio. Net. Для реалізації механізму доступу до даних БД використовується технологія ADO.NET – технологія обробки даних у відриві від їх джерела [5]. Додатки підключаються до бази даних тільки на невеликий проміжок часу. З'єднання встановлюється тільки тоді, коли клієнт з віддаленого комп'ютера запитує на централізованому сервері дані.

При програмній реалізації механізмів переглядання, додавання, видалення та пошуку даних із довідникової інформації застосовується SqlDataAdapter та його відповідні властивості і методи. Об'єкт SqlDataAdapter [6] створюється для кожної таблиці, з якою працює форма. Створення SqlDataAdapter здійснювалось через конструктор SqlDataAdapter із обов'язковим створенням з'єднання до БД і створенням SqlConnection [7]. На основі описаних даних в SqlDataAdapter здійснюється генерування об'єкта DataSet.

При програмній реалізації механізмів переглядання, додавання, видалення та пошуку даних із документа вхідної інформації використовується об'єкт DataSet. При цьому створюється об'єкт BindingSource та відповідний tableAdapter.

При програмній реалізації механізмів генерування вихідної інформації використовувалися представлення бази даних та запити SqlDataAdapter.

Введення інформації у текстові поля супроводжуються валідацією даних із застосування методів IsDigital, IsUpper и т.п.

Операції з даними, які можуть призвести до руйнування проекту супроводжуються реалізацією виключних ситуацій, а саме: видалення даних з порожньої таблиці, спроба обробки даних при невстановленому з'єднанні з базою даних, вихід параметрів запитів до БД за межі допустимих значень, спроба дублювання первинного ключа и т.п.

Для забезпечення цілісності даних БД реалізовано програмний механізм автоінкременту первинного ключа та механізму каскадних змін даних.

ВИСНОВКИ

В результаті виконання курсової роботи було здійснено:

- представлено результати аналізу предметної області у вигляді функціональної та об'єктної моделі предметної області.
- розроблено структуру меню АІС із визначенням складових частин нормативно-довідникової, вхідної та вихідної інформації:
 - нормативно-довідникову інформацію:
 - техніка ділянки;
 - паливо;
 - водії.
 - вхідну інформацію:
 - подорожній лист.
 - вихідну інформацію:
 - обліку витрат палива по водіям;
 - обліку витрат палива по подорожньому листу;
 - обліку витрат палива по техніці.
- представлено результати моделювання БД;
- наведено всі екранні форми АІС із визначення прав доступу до них користувачів з різними правами доступу

Розроблена архітектура АІС складається з наступних рівнів:

- Рівень надання інформації – це набір екранних форм користувача для роботи з системою, розроблені на основі стандарту графічного інтерфейсу WIMP. За допомогою екранних форм користувачі вводять, редагують, видаляють, здійснюють пошук та отримують необхідну вихідну інформацію. Всі форми виконані в єдиному стилі з урахуванням вимог стандартизації та уніфікації. Екранні форми та механізми оновлення бази даних через них реалізовані в програмного середовищі Microsoft Visual Studio 2010, на мові програмування C#.

- Рівень бізнес-логіки в системі реалізується як в вигляді збережених процедур та представлень СУБД – MSSQLServer 2008, так і на рівні програмного додатку для розрахунку суми грошових коштів, отриманих від продажу квитків.

- Рівень збереження інформації реалізовано на СУБД MSSQLServer 2008. БД забезпечує контроль цілісності інформації.

Передбачається, що розроблена АІС „Облік витрат палива технікою на ділянки” буде функціонувати в локальній мережі будь-якого виробничого підприємства. Внесення даних до БД відбувається операторами в залежності від ситуації. Вся електронна інформація, що створюється, зберігається в єдиній реляційній базі даних, створеної в рамках СУБД – MSSQLServer 2008. Компоненти системи працюють з одними і тими ж даними. Обмін інформацією між компонентами виконується тільки через базу даних. Таким чином, будь-які зміни в даних відразу ж стають доступними всім модулям системи.

ПЕРЕЛІК ПОСИЛАНЬ

1. Батюк А.Є. та ін. Інформаційні системи в менеджменті: Навч. посіб/ А.Є. Батюк, З.П. Двудіт, К.М. Обельовська, І.М. Огородник, Л.П. Фабрі. – Львів: Національний університет „Львівська політехніка”, 2004. – 520с.

Додаток А. Лістинг створення бази даних

```
USE [master]
GO
/***** Об'єкт: Database [LuckyTop]  Дата сценарія: 05/15/2013 23:26:21 *****/
CREATE DATABASE [LuckyTop] ON PRIMARY
( NAME = N'LuckyTop', FILENAME = N'D:\ Base\LuckyTop.mdf' , SIZE = 3072KB , MAXSIZE =
UNLIMITED, FILEGROWTH = 1024KB )
LOG ON
( NAME = N'LuckyTop_log', FILENAME = N'D:\ Base\LuckyTop_log.ldf' , SIZE = 1024KB ,
MAXSIZE = 2048GB , FILEGROWTH = 10%)
COLLATE Cyrillic_General_CI_AS
GO
EXEC dbo.sp_dbcmptlevel @dbname=N'LuckyTop', @new_cmptlevel=90
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [LuckyTop].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
ALTER DATABASE [LuckyTop] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [LuckyTop] SET ANSI_NULLS OFF
GO
ALTER DATABASE [LuckyTop] SET ANSI_PADDING OFF
GO
ALTER DATABASE [LuckyTop] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [LuckyTop] SET ARITHABORT OFF
GO
ALTER DATABASE [LuckyTop] SET AUTO_CLOSE ON
GO
ALTER DATABASE [LuckyTop] SET AUTO_CREATE_STATISTICS ON
GO
ALTER DATABASE [LuckyTop] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [LuckyTop] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [LuckyTop] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [LuckyTop] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [LuckyTop] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [LuckyTop] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [LuckyTop] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [LuckyTop] SET RECURSIVE_TRIGGERS OFF
GO
```

```

ALTER DATABASE [LuckyTop] SET DISABLE_BROKER
GO
ALTER DATABASE [LuckyTop] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [LuckyTop] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [LuckyTop] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [LuckyTop] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [LuckyTop] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [LuckyTop] SET READ_WRITE
GO
ALTER DATABASE [LuckyTop] SET RECOVERY SIMPLE
GO
ALTER DATABASE [LuckyTop] SET MULTI_USER
GO
ALTER DATABASE [LuckyTop] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [LuckyTop] SET DB_CHAINING OFF
USE [LuckyTop]
GO
/***** Íáúâçò: Table [dbo].[Mashunu]  Äàòà ñöáíàðèÿ: 05/15/2013 23:26:00 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Mashunu](
    [ID_MASH] [int] IDENTITY(1,1) NOT NULL,
    [ID_TipM] [int] NOT NULL,
    [Nomer] [varchar](50) COLLATE Cyrillic_General_CI_AS NULL,
    [Rashod] [varchar](50) COLLATE Cyrillic_General_CI_AS NULL,
    [Model] [varchar](50) COLLATE Cyrillic_General_CI_AS NULL,
    [Marka] [varchar](50) COLLATE Cyrillic_General_CI_AS NULL,
    [God] [int] NULL,
    CONSTRAINT [PK_Mashunu] PRIMARY KEY CLUSTERED
(
    [ID_MASH] ASC
)WITH (PAD_INDEX = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
ALTER TABLE [dbo].[Mashunu] WITH CHECK ADD CONSTRAINT
[FK_Mashunu_TipMashun1] FOREIGN KEY([ID_TipM])
REFERENCES [dbo].[TipMashun] ([ID_TIPM])
ON DELETE CASCADE

```

```

GO
ALTER TABLE [dbo].[Mashunu] CHECK CONSTRAINT [FK_Mashunu_TipMashun1]
USE [LuckyTop]
GO
/***** Íáúâè: Table [dbo].[TipMashun]  Äàòà ñöáíàðèÿ: 05/15/2013 23:26:55 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[TipMashun](
    [ID_TIPM] [int] IDENTITY(1,1) NOT NULL,
    [Name_TMash] [varchar](50) COLLATE Cyrillic_General_CI_AS NULL,
    CONSTRAINT [PK_TipMashun] PRIMARY KEY CLUSTERED
(
    [ID_TIPM] ASC
)WITH (PAD_INDEX = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
USE [LuckyTop]
GO
/***** Íáúâè: Table [dbo].[toplivo]  Äàòà ñöáíàðèÿ: 05/15/2013 23:27:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[toplivo](
    [ID_toplivo] [int] IDENTITY(1,1) NOT NULL,
    [nazv] [varchar](50) COLLATE Cyrillic_General_CI_AS NULL,
    [marka] [varchar](50) COLLATE Cyrillic_General_CI_AS NULL,
    CONSTRAINT [PK_toplivo] PRIMARY KEY CLUSTERED
(
    [ID_toplivo] ASC
)WITH (PAD_INDEX = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
USE [LuckyTop]
GO
/***** Íáúâè: Table [dbo].[Voditeli]  Äàòà ñöáíàðèÿ: 05/15/2013 23:27:15 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON

```



```

GO
CREATE TABLE [dbo].[Voditeli](
    [ID_VOD] [int] IDENTITY(1,1) NOT NULL,
    [Name_VOD] [varchar](50) COLLATE Cyrillic_General_CI_AS NULL,
    [Tab_nomer] [varchar](50) COLLATE Cyrillic_General_CI_AS NULL,
    CONSTRAINT [PK_Voditeli] PRIMARY KEY CLUSTERED
(
    [ID_VOD] ASC
)WITH (PAD_INDEX = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
USE [LuckyTop]
GO
/***** Íáúâè: Table [dbo].[pytevoy]  Äàòà ñöáíàðëÿ: 05/15/2013 23:27:27 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[pytevoy](
    [Kod_a] [int] IDENTITY(1,1) NOT NULL,
    [Nomer_a] [int] NULL,
    [Data_a] [datetime] NULL,
    [ID_tipm] [int] NULL,
    [ID_mash] [int] NULL,
    [Kod_v] [int] NULL,
    [Id_toplivo] [int] NULL,
    [pokaz] [int] NULL,
    [rastoyanie] [int] NULL,
    [vudelenotop] [int] NULL,
    [ostat_v] [int] NULL,
    [ostat_p] [int] NULL,
    CONSTRAINT [PK_pytevoy] PRIMARY KEY CLUSTERED
(
    [Kod_a] ASC
)WITH (PAD_INDEX = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

```

Додаток Б. Лістинг створення програмного забезпечення

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace RashodGSM
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void водителиToolStripMenuItem_Click(object sender, EventArgs e)
        {
            vod2 fr=new vod2();
            fr.Show();
        }
        private void мвшиныToolStripMenuItem_Click(object sender, EventArgs e)
        {
            tex frm = new tex();
            frm.ShowDialog();
        }
        private void направлениеToolStripMenuItem_Click(object sender, EventArgs e)
        {
            Napravlenia frm = new Napravlenia();
            frm.ShowDialog();
        }
        private void расходТопливаToolStripMenuItem_Click(object sender, EventArgs e)
        {
            rez1 frm = new rez1();
            frm.ShowDialog();
        }
        private void выходToolStripMenuItem_Click(object sender, EventArgs e)
        {
            Close();
        }
        private void путевойЛистToolStripMenuItem_Click(object sender, EventArgs e)
        {
            pyt_gl frm=new pyt_gl();
            frm.ShowDialog();
        }
        private void hELPMYToolStripMenuItem_Click(object sender, EventArgs e)
    }
}
```

```

        {
            HELP frm = new HELP();
            frm.ShowDialog();
        }
private void обзорПутЛистовToolStripMenuItem_Click(object sender, EventArgs e)
{
    rez2 frm = new rez2();
    frm.ShowDialog();
}
private void машиныToolStripMenuItem_Click(object sender, EventArgs e)
{
    rez3 frm = new rez3();
    frm.ShowDialog();
}
private void топливоToolStripMenuItem_Click(object sender, EventArgs e)
{
    topl fr = new topl();
    fr.Show();
}
}
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace RashodGSM
{
    public partial class rez1 : Form
    {
        int allsum,rast;
        public rez1()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            if (radioButton1.Checked == true)
            {
                sqlDataAdapter1.SelectCommand.Parameters["@Param1"].Value = dateTimePicker1.Value;
                sqlDataAdapter1.SelectCommand.Parameters["@Param2"].Value = dateTimePicker2.Value;
                sqlDataAdapter1.SelectCommand.Parameters["@Param4"].Value = comboBox1.Text;
                luckyTopDataSet7.Clear();
                sqlDataAdapter1.Fill(luckyTopDataSet7);
            }
        }
    }
}

```

```

        rast = 0;
        allsum = 0;
        for (int i = 0; i < dataGridView1.RowCount - 1; i++)
        {
            allsum +=
Convert.ToInt32(dataGridView1.Rows[i].Cells["vudelenotop"].Value.ToString());
            rast +=
Convert.ToInt32(dataGridView1.Rows[i].Cells["rastoyanie"].Value.ToString());
        }
        label3.Text = "Всего выделено " + allsum.ToString() + " литров.";
        label4.Text = "Всего пройдено " + rast.ToString() + " км.";
    }
    else
    {
        sqlDataAdapter1.SelectCommand.Parameters["@Param1"].Value = dateTimePicker1.Value;
        sqlDataAdapter1.SelectCommand.Parameters["@Param2"].Value = dateTimePicker2.Value;
        sqlDataAdapter1.Fill(luckyTopDataSet7);
        this.voditeliTableAdapter.Fill(this.luckyTopDataSet7.Voditeli);
        rast = 0;
        allsum = 0;
        for (int i = 0; i < dataGridView1.RowCount - 1; i++)
        {
            allsum +=
Convert.ToInt32(dataGridView1.Rows[i].Cells["vudelenotop"].Value.ToString());
            rast +=
Convert.ToInt32(dataGridView1.Rows[i].Cells["rastoyanie"].Value.ToString());
        }
        label3.Text = "Всего выделено " + allsum.ToString() + " литров.";
        label4.Text = "Всего пройдено " + rast.ToString() + " км.";
        if (radioButton2.Checked == true)
        {
            sqlDataAdapter1.SelectCommand.Parameters["@Param1"].Value = dateTimePicker1.Value;
            sqlDataAdapter1.SelectCommand.Parameters["@Param2"].Value = dateTimePicker2.Value;
            sqlDataAdapter1.SelectCommand.Parameters["@Param3"].Value = textBox1.Text;
            sqlDataAdapter1.SelectCommand.Parameters["@Param4"].Value = "% %";
            luckyTopDataSet7.Clear();
            sqlDataAdapter1.Fill(luckyTopDataSet7);
            rast = 0;
            allsum = 0;
            for (int i = 0; i < dataGridView1.RowCount - 1; i++)
            {
                allsum +=
Convert.ToInt32(dataGridView1.Rows[i].Cells["vudelenotop"].Value.ToString());
                rast +=
Convert.ToInt32(dataGridView1.Rows[i].Cells["rastoyanie"].Value.ToString());
            }
            label3.Text = "Всего выделено " + allsum.ToString() + " литров.";
        }
    }
}

```

```

        label4.Text = "Всего пройдено" + rast.ToString() + " км.";
    }
}
}
private void rez1_Load(object sender, EventArgs e)
{
    this.voditeliTableAdapter.Fill(this.luckyTopDataSet7.Voditeli);
    label3.Text = "";
    label4.Text = "";
    sqlDataAdapter1.SelectCommand.Parameters["@Param1"].Value = dateTimePicker1.Value;
    sqlDataAdapter1.SelectCommand.Parameters["@Param2"].Value = dateTimePicker2.Value;
    sqlDataAdapter1.SelectCommand.Parameters["@Param3"].Value = "% %";
    sqlDataAdapter1.SelectCommand.Parameters["@Param4"].Value = "% %";
    sqlDataAdapter1.Fill(luckyTopDataSet7);
}
private void button2_Click(object sender, EventArgs e)
{
    sqlDataAdapter1.SelectCommand.Parameters["@Param1"].Value =
dateTimePicker1.Value;
    sqlDataAdapter1.SelectCommand.Parameters["@Param2"].Value =
dateTimePicker2.Value;
    sqlDataAdapter1.SelectCommand.Parameters["@Param3"].Value = "% %";
    sqlDataAdapter1.SelectCommand.Parameters["@Param4"].Value = "% %";
    radioButton1.Checked = false;
    radioButton2.Checked = false;
    textBox1.Clear();
    luckyTopDataSet7.Clear();
    sqlDataAdapter1.Fill(luckyTopDataSet7);
    this.voditeliTableAdapter.Fill(this.luckyTopDataSet7.Voditeli);
    rast = 0;
    allsum = 0;
    for (int i = 0; i < dataGridView1.RowCount - 1; i++)
    {
        allsum +=
Convert.ToInt32(dataGridView1.Rows[i].Cells["vudelenotop"].Value.ToString());
        rast +=
Convert.ToInt32(dataGridView1.Rows[i].Cells["rastoyanie"].Value.ToString());
    }
    label3.Text = "Всего выделено" + allsum.ToString() + " литров.";
    label4.Text = "Всего пройдено" + rast.ToString() + " км.";
}
private void dateTimePicker2_ValueChanged(object sender, EventArgs e)
{
    if (dateTimePicker1.Value > dateTimePicker2.Value)
    {
        MessageBox.Show("Конечна дата запроса не должна быть меньше начальной",
"", MessageBoxButtons.OK, MessageBoxIcon.Hand);
    }
}

```

```

        dateTimePicker2.Value = dateTimePicker1.Value;
    }
}
}
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace RashodGSM
{
    public partial class rez2 : Form
    {
        int vud, dol;
        public rez2()
        {
            InitializeComponent();
        }
        private void rez2_Load(object sender, EventArgs e)
        {
            sqlDataAdapter1.SelectCommand.Parameters["@Param1"].Value = dateTimePicker1.Value;
            sqlDataAdapter1.SelectCommand.Parameters["@Param2"].Value = dateTimePicker2.Value;
            sqlDataAdapter1.Fill(luckyTopDataSet7);
            label3.Text = "";
            label4.Text = "";
        }
        private void button1_Click(object sender, EventArgs e)
        {
            sqlDataAdapter1.SelectCommand.Parameters["@Param1"].Value = dateTimePicker1.Value;
            sqlDataAdapter1.SelectCommand.Parameters["@Param2"].Value = dateTimePicker2.Value;
            luckyTopDataSet7.Clear();
            sqlDataAdapter1.Fill(luckyTopDataSet7);
            for (int i = 0; i < dataGridView1.RowCount - 1; i++)
            {
                if
                ((Convert.ToInt32(dataGridView1.Rows[i].Cells["vudelenotop"].Value.ToString()) >
                Convert.ToInt32(dataGridView1.Rows[i].Cells["rekom"].Value.ToString()))
                dataGridView1.Rows[i].Cells["vudelenotop"].Style.BackColor =
                System.Drawing.Color.Red;
            }
            for (int i = 0; i < dataGridView1.RowCount - 1; i++)
            {

```

```

        if (Convert.ToInt32(dataGridView1.Rows[i].Cells["ostat_p"].Value.ToString()) <
Convert.ToInt32(dataGridView1.Rows[i].Cells["doljno"].Value.ToString()))
            dataGridView1.Rows[i].Cells["ostat_p"].Style.BackColor =
System.Drawing.Color.Red;
    }
    vud = 0;
    dol = 0;
    for (int i = 0; i < dataGridView1.RowCount - 1; i++)
    {
        vud +=
Convert.ToInt32(dataGridView1.Rows[i].Cells["vudelenotop"].Value.ToString());
        dol += Convert.ToInt32(dataGridView1.Rows[i].Cells["rekom"].Value.ToString());
    }
    label3.Text = "Всего выделено " + vud.ToString() + " литров.";
    label4.Text = "Рекомендовано выделить " + dol.ToString() + " литров.";
    if(vud>dol)
        label5.Text="Перерасход " + (vud-dol).ToString() + " литров.";
    else
        label5.Text="Экономия " + (dol-vud).ToString() + " литров.";
    }
    private void button2_Click(object sender, EventArgs e)
    {
        sqlDataAdapter1.SelectCommand.Parameters["@Param1"].Value = dateTimePicker1.Value;
        sqlDataAdapter1.SelectCommand.Parameters["@Param2"].Value = dateTimePicker2.Value;
        luckyTopDataSet7.Clear();
        sqlDataAdapter1.Fill(luckyTopDataSet7);
    }
}
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace RashodGSM
{
    public partial class rez3 : Form
    {
        int rast, allsum;
        public rez3()
        {
            InitializeComponent();
        }
        private void rez3_Load(object sender, EventArgs e)

```

```

    {
        this.mashunuTableAdapter.Fill(this.luckyTopDataSet8.Mashunu);
        this.tipMashunTableAdapter.Fill(this.luckyTopDataSet8.TipMashun);
        label3.Text = "";
        label4.Text = "";
        sqlDataAdapter1.SelectCommand.Parameters["@Param1"].Value =
dateTimePicker1.Value;
        sqlDataAdapter1.SelectCommand.Parameters["@Param2"].Value =
dateTimePicker2.Value;
        sqlDataAdapter1.SelectCommand.Parameters["@Param3"].Value = "%%";
        sqlDataAdapter1.SelectCommand.Parameters["@Param4"].Value = "%%";
        sqlDataAdapter1.SelectCommand.Parameters["@Param5"].Value = "%%";
        sqlDataAdapter1.Fill(luckyTopDataSet8);
    }
    private void button1_Click(object sender, EventArgs e)
    {
        if (radioButton1.Checked == true)
        {
            sqlDataAdapter1.SelectCommand.Parameters["@Param1"].Value = dateTimePicker1.Value;
            sqlDataAdapter1.SelectCommand.Parameters["@Param2"].Value = dateTimePicker2.Value;
            sqlDataAdapter1.SelectCommand.Parameters["@Param3"].Value = comboBox1.Text;
            luckyTopDataSet8.Clear();
            sqlDataAdapter1.Fill(luckyTopDataSet8);
            this.tipMashunTableAdapter.Fill(this.luckyTopDataSet8.TipMashun);
            this.mashunuTableAdapter.Fill(this.luckyTopDataSet8.Mashunu);
            rast = 0;
            allsum = 0;
            for (int i = 0; i < dataGridView1.RowCount - 1; i++)
            {
                allsum +=
Convert.ToInt32(dataGridView1.Rows[i].Cells["expr3"].Value.ToString());
                rast +=
Convert.ToInt32(dataGridView1.Rows[i].Cells["expr2"].Value.ToString());
            }
            label3.Text = "Всего выделено " + allsum.ToString() + " литров.";
            label4.Text = "Всего пройдено " + rast.ToString() + " км.";
        }
        else
        {
            sqlDataAdapter1.SelectCommand.Parameters["@Param1"].Value =
dateTimePicker1.Value;
            sqlDataAdapter1.SelectCommand.Parameters["@Param2"].Value =
dateTimePicker2.Value;
            sqlDataAdapter1.Fill(luckyTopDataSet8);
            this.tipMashunTableAdapter.Fill(this.luckyTopDataSet8.TipMashun);
            this.mashunuTableAdapter.Fill(this.luckyTopDataSet8.Mashunu);
            rast = 0;
        }
    }
}

```



```

        allsum = 0;
        for (int i = 0; i < dataGridView1.RowCount - 1; i++)
        {
            allsum +=
Convert.ToInt32(dataGridView1.Rows[i].Cells["expr3"].Value.ToString());
            rast +=
Convert.ToInt32(dataGridView1.Rows[i].Cells["expr2"].Value.ToString());
        }
        label3.Text = "Всего выделено " + allsum.ToString() + " литров.";
        label4.Text = "Всего пройдено" + rast.ToString() + " км.";
        if (radioButton2.Checked == true)
        {
            sqlDataAdapter1.SelectCommand.Parameters["@Param1"].Value = dateTimePicker1.Value;
            sqlDataAdapter1.SelectCommand.Parameters["@Param2"].Value = dateTimePicker2.Value;
            sqlDataAdapter1.SelectCommand.Parameters["@Param5"].Value = textBox1.Text;
            luckyTopDataSet8.Clear();
            sqlDataAdapter1.Fill(luckyTopDataSet8);
            this.tipMashunTableAdapter.Fill(this.luckyTopDataSet8.TipMashun);
            this.mashunuTableAdapter.Fill(this.luckyTopDataSet8.Mashunu);
            rast = 0;
            allsum = 0;
            for (int i = 0; i < dataGridView1.RowCount - 1; i++)
            {
                allsum +=
Convert.ToInt32(dataGridView1.Rows[i].Cells["expr3"].Value.ToString());
                rast +=
Convert.ToInt32(dataGridView1.Rows[i].Cells["expr2"].Value.ToString());
            }
            label3.Text = "Всего выделено " + allsum.ToString() + " литров.";
            label4.Text = "Всего пройдено" + rast.ToString() + " км.";
        }
    }

private void button2_Click(object sender, EventArgs e)
{
    sqlDataAdapter1.SelectCommand.Parameters["@Param1"].Value =
dateTimePicker1.Value;
    sqlDataAdapter1.SelectCommand.Parameters["@Param2"].Value =
dateTimePicker2.Value;
    sqlDataAdapter1.SelectCommand.Parameters["@Param3"].Value = "%";
    sqlDataAdapter1.SelectCommand.Parameters["@Param4"].Value = "%";
    sqlDataAdapter1.SelectCommand.Parameters["@Param5"].Value = "%";
    radioButton1.Checked = false;
    radioButton2.Checked = false;
    textBox1.Clear();
    luckyTopDataSet8.Clear();
}

```

```

sqlDataAdapter1.Fill(luckyTopDataSet8);
this.tipMashunTableAdapter.Fill(this.luckyTopDataSet8.TipMashun);
this.mashunuTableAdapter.Fill(this.luckyTopDataSet8.Mashunu);
rast = 0;
allsum = 0;
for (int i = 0; i < dataGridView1.RowCount - 1; i++)
{
    allsum +=
Convert.ToInt32(dataGridView1.Rows[i].Cells["expr3"].Value.ToString());
    rast += Convert.ToInt32(dataGridView1.Rows[i].Cells["expr2"].Value.ToString());
}
label3.Text = "Всего выделено " + allsum.ToString() + " литров.";
label4.Text = "Всего пройдено " + rast.ToString() + " км.";
}
private void radioButton2_CheckedChanged(object sender, EventArgs e)
{
}
}
}

```

