

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет водного господарства та природокористування
Навчально-науковий інститут автоматичної, кібернетики та
обчислювальної техніки
Кафедра комп'ютерних технологій та економічної кібернетики

Допущено до захисту:

Завідувач кафедри

_____ д. е. н., проф. П. М. Грицюк

« _____ » _____ 2021 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня «бакалавр»

за освітньо-професійною програмою «Інформаційні системи і технології»
спеціальності 126 «Інформаційні системи та технології»

на тему: **«Інформаційна система обліку робіт з обслуговування
автотранспорту на станції технічного огляду»**

Виконав:

здобувач вищої освіти 2 курсу, групи ІСТ-21інт
Козачок Олександр Віталійович

Керівник:

канд. техн. наук, доцент Гладка О. М.

Рецензент:

д-р екон. наук, професор Грицюк П. М.

Рівне – 2021

Національний університет водного господарства та природокористування
ННІ автоматики, кібернетики та обчислювальної техніки
Кафедра комп'ютерних технологій та економічної кібернетики

Освітньо-кваліфікаційний рівень – **бакалавр**
за освітньо-професійною програмою **«Інформаційні системи і технології»**
спеціальність **126 «Інформаційні системи та технології»**

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ д. е. н., проф. П. М. Грицюк

« _____ » _____ 2021 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту _____ Козачку Олександр Віталійовичу
(прізвище, ім'я, по-батькові)

1. Тема роботи Інформаційна система обліку робіт з обслуговування автотранспорту на станції технічного огляду

керівник роботи: Гладка Олена Миколаївна, канд. техн. наук, доцент

(прізвище, ім'я, по-батькові, науковий ступінь, вчене звання)

затверджена наказом по університету від " 02 " квітня 2021 р. С № 263

2. Термін здачі студентом закінченої роботи " 11 " червня 2021 р.

3. Вихідні дані до роботи спроектувати та розробити інформаційну систему для обліку робіт з обслуговування автотранспорту на станції технічного огляду із сучасним та зручним для використання функціоналом.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) вступ; аналіз предметної області; вибір та обґрунтування засобів розробки програмного продукту; розробка і опис створеної інформаційної системи.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання “ 12 ” жовтня 2020 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Термін виконання етапів роботи	Примітка
1.	Аналіз об'єкта дослідження, виявлення існуючих проблем	12.10.20 – 30.10.20	
2.	Аналіз існуючих інформаційних методів (технологій) вирішення проблеми	2.11.20 – 20.11.20	
3.	Вибір та обґрунтування засобів розробки інформаційної системи	23.11.20 – 11.12.20	
4.	Проектування, розробка та реалізація інформаційної системи	14.12.20 – 30.04.21	
5.	Підготовка пояснювальної записки	5.04.21 – 21.05.21	
6.	Підготовка презентації роботи	24.05.21 – 11.06.21	
7.	Відгук керівника, рецензування роботи, перевірка на плагіат	11.06.21 – 21.06.21	

Студент _____ **О. В. Козачок**
(підпис) (прізвище і ініціали)

Керівник кваліфікаційної роботи

_____ **О. М. Гладка**
(підпис) (прізвище і ініціали)

Анотація

УДК 004.451.87

Козачок О.В. «Інформаційна система обліку робіт з обслуговування автотранспорту на станції технічного огляду» / Кваліфікаційна бакалаврська робота / м. Рівне: НУВГП, 2021. 60 с. Українською мовою.

Формул 13, ілюстрацій 25, таблиць 2.

Розроблено інформаційну систему, що вирішує поставлені задачі за допомогою механізму реляційних баз даних. Створена автоматизована інформаційна система забезпечує ефективне збереження та цілісність даних, за допомогою форм, кожна з яких відповідає певній таблиці чи сукупності таблиць, дозволяє легке введення, редагування, пошук даних в базі, за допомогою механізму запитів дозволяє пошук та відбір даних в базі за певними критеріями і ці дані можна перенаправити у відповідний звіт, який можна віддрукувати.

Створення автоматизованої інформаційної системи для обліку робіт станції технічного огляду автотранспорту реалізовано для контролю виконаних робіт на станції СТО.

Annotation

UDC 004.451.87

Kozachok OV "Information system of accounting for motor vehicle maintenance at the technical inspection station" / Thesis / Rivne: NUVGP, 2021.60 p. In Ukrainian.

Formulas 13, illustrations 25, tables 2.

The value of this dissertation lies in the creation of an information system that solves the tasks with the help of the mechanism of relational databases. The created automated information system provides effective storage and integrity of data, by means of forms, each of which corresponds to a certain table or set of tables, allows easy input, editing, search of data in data. and this data can be redirected to an appropriate report that can be printed.

Creation of the automated information system for accounting of bits of station of technical inspection of motor transport is realized for the control of the executed bits at station STO.

The organization of the car technical inspection station consists first of all in the development of the coding system to facilitate the work at this station. The finished object is considered to be a completed structural device with all the equipment belonging to it. It should be noted an important point: the information system used in this thesis really automates the work of the company, but automates in the sense that it eliminates the need to keep paper documents, records and documents. However, the main goal of this dissertation has been achieved, namely the information system that supports the model of relational databases, ensuring the safety, reliability and integrity of the entered data.

ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1. ОПИС ТА АНАЛІЗ ОБ’ЄКТУ ДОСЛІДЖЕННЯ.....	11
1.1 Аналіз предметної області	11
1.2 Загальносистемні вимоги до продукту, що розробляється	12
1.3 Інформаційно-математична суть проблеми	13
1.4 Вхідні дані.....	13
1.5 Вихідні результати.....	14
РОЗДІЛ 2. ОПИС МОДЕЛІ ТА ТЕХНОЛОГІЙ І МЕТОДІВ ЇЇ РЕАЛІЗАЦІЇ	
2.1 Бази даних та системи управління ними	15
2.2 Класифікація баз даних	16
2.3 Інфологічна модель бази даних, її види, проектування.....	18
2.4 Принципи проектування ІС	20
2.5 Побудова моделі даних	21
2.6 Побудова автоматизованої інформаційної системи.....	22
2.7 Відкрита інформаційна система	23
РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОЕКТУ	25
3.1 Призначення розробки	25
3.2 Середовище розробки.....	25
3.3 Вікно коду програми.	30
3.4 Головна форма програми	32
3.4 Інструкція системного програміста	38
3.5 Інструкція користувача	39
3.6 Економічне обґрунтування проекту.....	39
Висновки	42

Джерела літератури	44
ДОДАТКИ.....	46

ВСТУП

Ефективність діяльності будь-якого підприємства багато в чому залежить від обсягу реалізованої нею продукції, робіт і послуг. При здійсненні комерційної діяльності на підприємстві проводиться обробка та зберігання великої кількості інформації, пов'язаної з наданням послуг, продажем чи замовленнями реалізованої продукції тощо, суть яких полягає у своєчасному і правильному оформленні документів та контролі за кожною операцією, що відбувається на підприємстві. У зв'язку з цим для надійного функціонування системи надання послуг необхідно вести систематичний і безперервний облік за виконуваними роботами, що і є метою цього проекту. Основне завдання будь-якого успішного проекту полягає в тому, щоб на момент запуску системи і протягом усього часу її експлуатації можна було забезпечити:

- необхідну функціональність системи та ступінь адаптації до мінливих умов її функціонування;
- необхідну пропускну здатність системи;
- потрібний час реакції системи на запит;
- безвідмовну роботу системи в потрібному режимі, тобто – готовність і доступність системи для обробки запитів користувачів;
- простоту експлуатації і підтримки системи.

Головним чинником, що визначає ефективність системи, є продуктивність, а гарне проектне рішення служить основою високопродуктивної системи.

Метою розробки автоматизованої інформаційної системи для обліку робіт на станції технічного огляду автотранспорту є спрощення процедури обліку послуг, що надаються, автомобілів і клієнтів, скорочення часу, який витрачається при виборі відповідного в усіх відношеннях варіанту.

Проектування інформаційної системи охоплює такі основні області:

- проектування об'єктів даних, які будуть реалізовані в базі даних;

- проектування програм, екранних форм, які будуть забезпечувати виконання запитів до даних;

- облік конкретного середовища або технології, а саме: топології мережі, конфігурації апаратних засобів, використовуваної архітектури (файл-сервер або клієнт-сервер), паралельної обробки, розподіленої обробки даних і т.д.

У реальних умовах, проектування – це пошук способу, який задовольняє вимогам функціональності системи засобами наявних технологій з урахуванням заданих обмежень. До будь-якого проекту пред'являється ряд абсолютних вимог, наприклад максимальний час розробки проекту, максимальні грошові вкладення в проект і т.д. Проектування саме по собі є дуже складним. Процес розробки проекту включає багато стадій, завдань, які потрібно виконувати чітко і послідовно.

XXI століття – це століття комп'ютерних технологій, в якому все більше значення приділяється розробці автоматизованих багатофункціональних систем управління базами даних. Це підтверджує актуальність даної роботи, що полягає в автоматизації обліку проведених робіт з технічного обслуговування автомобілів.

РОЗДІЛ 1. ОПИС ТА АНАЛІЗ ОБ'ЄКТУ ДОСЛІДЖЕННЯ

1.1 Аналіз предметної області

Метою дипломної роботи є розробка та створення програмного продукту «Автоматизованої інформаційної системи для обліку робіт на станції технічного огляду автомобілів». Це програмне забезпечення призначене для створення системи обліку замовлень і контролю за якістю виконаної роботи на станції технічного огляду (СТО), а також обліку заробітної плати згідно виконаних обсягів робіт. Розроблений програмний продукт забезпечує необхідний набір функцій, даючи можливість доступу до інтерфейсних об'єктів користувацького подання програмного забезпечення за допомогою миші та клавіатури.

Робота проектувальників бази даних в значній мірі залежить від якості інформаційної моделі. Інформаційна модель не повинна містити ніяких незрозумілих конструкцій, які не можна реалізувати в рамках вибраної СУБД. Слід зазначити, що інформаційна модель створюється для того, щоб на її основі можна було побудувати модель даних, тобто повинна враховувати особливості реалізації вибраної СУБД.

Побудова логічної і фізичної моделей даних є основною частиною проектування бази даних. Отримана в процесі аналізу інформаційна модель спочатку перетворюється в логічну, а потім у фізичну модель даних. Після цього була створена пробна база даних, з якою почалася робота над створенням коду інформаційної системи. В ідеалі до моменту початку розробки модель даних повинна бути стійка. Проектування бази даних не може бути відірвано від проектування модулів та програм, оскільки бізнес-правила можуть створювати об'єкти в базі даних, наприклад серверні обмеження (constraints), а також збережені процедури і тригери, – у цьому випадку часто кажуть, що частина бізнес-логіки переноситься до бази даних. Проектування моделі даних для кожної СУБД містить свої особливості: проектні рішення, які дають гарний результат для однієї СУБД, можуть виявитися зовсім неприйнятними для іншої.

Після проведення обстеження СТО і збору інформації встановлено, що на СТО можна автоматизувати такі процеси:

- створення бази замовлень, що дасть можливість проаналізувати кількість прийнятих замовлень строки виконання робіт та постійність обслуговування клієнтів;
- створення прас-листа, який автоматично формулює ціни виконаних робіт;
- створення обліку виконаних робіт, який показує загальну суму, на яку виконані роботи, які роботи проведені і яким працівником, що дає можливість контролювати якість виконаної роботи;
- створення бази даних працівників(особові справи), місячний оклад;
- створення автоматичного підрахунку сум виконаних робіт за період та облік оплати клієнтів, який показує загальний об'єм робіт проведений на даному СТО і вчасність розрахунку клієнтів.

1.2 Загальносистемні вимоги до продукту, що розробляється

Основна мета проектування бази даних – це скорочення надмірності збережених даних, а отже, економія обсягу використаної пам'яті, зменшення витрат на багаторазові операції оновлення надлишкових копій та усунення можливості виникнення суперечностей через зберігання в різних місцях відомостей про один і той же об'єкт.

Автоматизовану систему інформації можна розглядати з декількох точок зору: як мережа інформаційних служб (головний інформаційно-обчислювальний центр, локальні обчислювальні центри виробництв, майстерень та інших підрозділів, автоматизовані робочі місця й інші складові) з розміщеними в ній масивами:

- зберігання інформації, документами, технічними засобами реєстрації;
- зберігання, передачі, обробки, представлення інформації, програмним забезпеченням, методичним забезпеченням (інструкціями для користувачів, положеннями про підрозділи тощо).

На основі загальносистемних вимог до продукту було розроблено програмний продукт технічним завданням якого, є автоматизація роботи відділів станції технічного огляду та збору інформації про об'єми виконаних робіт на СТО і прибутковість станції.

1.3 Інформаційно-математична суть проблеми

В сучасному світі при використанні нанотехнологій, які поліпшують високотехнічність механізмів і розширюють сферу їх застосування виникає необхідність в застосуванні в усіх сферах діяльності людини автоматизованих інформаційних систем. Вони дають можливість керувати процесами на відстані, застосовувати аналітичний і економічний аналіз роботи підприємства і досягати якості в роботі і підвищувати його прибутковість. Саме тому в рамках дипломної роботи пропонується автоматизувати роботу СТО для якісного обслуговування клієнтів та полегшення роботи працівникам, і аналізу роботи підприємства.

1.4 Вхідні дані

Аналізуючи інформаційні потоки, що циркулюють на станції технічного огляду автомобілів і забезпечують її роботу, було створено бази даних, які мають всі необхідні вхідні відомості (рис. 1.1).

	Прізвище, ім'я	Стать	Член профспіл	Має страхування?	Дата народження	Домашня адреса	Домашній телефон	Адреса електронної пошти	Фотокарт	Дата влаштування
▶	Некомський А	чол	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	10.12.1991	вул. Данила Галицького 9	(0362)548-56-35	172.0.0.1@i.ua	Рисунок	01.10.2001
+	Сенюк Микола	чол	<input type="checkbox"/>	<input type="checkbox"/>	05.07.1992	вул. С. Бандери 8	(5454)645-64-656	kolya@mail.ru		02.04.1998
+	Мартинюк Мих	чол	<input type="checkbox"/>	<input type="checkbox"/>	15.08.1991	вул. Корольова 2	(2543)685-46-546	19misha91@mail.ru		01.01.1999
+	Целюк Стас	чол	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	20.07.1992	вул. кн.Острозького 4/6	(4488)546-24-	s1ck.92@bigmir.net		20.07.1996
+	Кривульський С	чол	<input type="checkbox"/>	<input type="checkbox"/>	20.12.1991	вул. Кн.Острозького 8а		exec@mail.ru		11.02.1996
+	Кротов Олексій	чол	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	04.03.1955					02.03.1986
+	Крилова Інна	О	<input checked="" type="checkbox"/>	<input type="checkbox"/>	09.01.1958					02.03.2001
+	Воронова Гали	О	<input type="checkbox"/>	<input checked="" type="checkbox"/>	19.09.1958	вул. Корольова 2	(6546)456-54-654			02.03.1999
+	Андрийчук Вас	чол	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	02.07.1963					02.03.1997
+	Ясенева Алла	О	<input type="checkbox"/>	<input checked="" type="checkbox"/>	02.07.1969		(6546)456-56-5			02.03.1976
+	Белова Іванна	О	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	08.12.1968		(6546)546-54-65			02.03.1998
+	Бабкіна Галина	О	<input type="checkbox"/>	<input checked="" type="checkbox"/>	30.08.1943	вул. Корольова 87	(5464)54 - -			02.03.2003
+	Новіков Артем	чол	<input checked="" type="checkbox"/>	<input type="checkbox"/>	19.02.1952					02.03.2004
+	Кравец Олександр	чол	<input checked="" type="checkbox"/>	<input type="checkbox"/>	29.05.1960		(6456)546 - -			02.03.2008
+	Петренко Тарас	чол	<input checked="" type="checkbox"/>	<input type="checkbox"/>	17.08.1951					25.03.2005
*			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			() - -			22.06.2010

Рис. 1.1 Приклад вхідних даних

Ця таблиця являється прикладом вхідних даних. Також у формах Microsoft Access було створено в головному меню автоматизований спосіб

введення даних тобто за допомогою натиску декількох кнопок можна за декілька секунд сформулювати замовлення клієнтів.

1.5 Вихідні результати

При створенні бази даних було розроблено в програмі Microsoft Access таблиці даних пов'язані між собою та головна форма програми. Головна форма програми представляє собою таблицю з назвою дипломної роботи та меню програми. Меню програми складається з:

- головної частини, яка своєю чергу складається з двох підрозділів: замовлення, облік роботи;

- довідники мають структуру: відділи, прайс-лист, облік роботи, клієнти, співробітники;

- запити, які складаються: жінки відділу, підрахунок сум, автомобіль за роками, кількість співробітників, обслуговування клієнтів, оплачені ремонти, стаж співробітників, клієнти за період;

- звіти, які складаються: з відділів, прайс-листа, замовлення, засобів зв'язку, окладу, стажу співробітників, авто за роками випуску, оплата клієнтів і клієнти за період;

- автор.

За результатами виконаної роботи всі вищеперечислені таблиці зв'язані між собою і працюють як єдине ціле, що показано в дипломній роботі.

.

РОЗДІЛ 2. ОПИС МОДЕЛІ ТА ТЕХНОЛОГІЙ І МЕТОДІВ ЇЇ РЕАЛІЗАЦІЇ

2.1 Бази даних та системи управління ними

Бази даних є одним з основних компонентів сучасних інформаційних систем. Інформаційна система – це взаємопов'язана сукупність засобів, методів і персоналу, використовуваних для зберігання, обробки та видачі інформації.

Мета будь-якої інформаційної системи – обробка інформації конкретної предметної області.

Під предметною областю розуміється сукупність пов'язаних між собою функцій, завдань управління в деякій галузі діяльності підприємства, за допомогою яких досягається виконання поставленої мети.

Система управління базами даних – це додаток, що забезпечує створення, зберігання, оновлення і пошук інформації в базі даних, а також управління безпекою і цілісністю даних. У найпростішому випадку – це програма, яка забезпечує роботу з інформацією. Інформація в базі може бути будь-яка, при цьому, як правило, робота з інформацією має на увазі наступні дії:

- додавання нової інформації;
- пошук інформації;
- зміна інформації;
- видалення інформації з бази даних;

Особливостями такої сукупності даних є:

1. Досить великі обсяги інформації;
2. Максимально можлива компактність зберігання даних;
3. Можливість вилучення з бази даних різноманітної інформації у визначеній предметній області;
4. Зручні для користувача вигляд і форма витягнутої інформації;
5. Висока швидкість доступу до даних;
6. Надійність зберігання інформації і можливість надання санкціонованого доступу до даних шляхом окремих користувачів;

7. Зручність і простота конструювання користувачем запитів, форма звітів для вибірки даних. Створення бази даних, її підтримка та забезпечення Доступу користувачів до неї здійснюється за допомогою спеціального програмного інструменту - системи управління базами даних.

Світ програмних систем, що дозволяють використовувати бази даних, досить різноманітний. В даний час існує досить велика кількість програмних систем, що дозволяють створювати і використовувати локальні і віддалені бази даних. Серед найбільш відомих можна відзначити Paradox, dBase, FoxPro, MS Access, InterBase, Oracle, Infomix, MS SQL Server та інші.

До складу середовища Borland Delphi входять компоненти, що дозволяють створювати програми роботи з файлами даних, створених різними системами: від dBase до Infomix і Oracle. Середовище розробки дозволяє програмісту за допомогою утиліти Database Explorer створювати файли баз даних в різних форматах.

2.2 Класифікація баз даних

Бази даних можуть бути локальні і видалені. Відмінності в типах систем полягає в розташуванні програми, що використовує дані, самих даних, а також способу розподілу даних між декількома користувачами.

Локальна база даних. Дані локальної бази даних (файли даних) знаходяться на одному (локальному) пристрої, в якості якого може виступати диск комп'ютера або диск іншого комп'ютера, що працює в мережі.

Для забезпечення розмежування доступу до даних між кількома користувачами в локальних базах даних застосовується метод, що отримав назву блокування файлів. Метод блокування полягає в тому, що поки дані використовуються одним користувачем, інший користувач не може працювати з цими даними. Як приклад локальних баз даних можна навести програмні системи Paradox, dBase, FoxPro та MS Access.

Віддалена база даних. Дані (файли) віддаленої бази даних знаходяться на віддаленому комп'ютері. При цьому не слід плутати мережеві диски комп'ютерів, що працюють у мережі, і каталоги віддаленого комп'ютера.

Програма роботи з віддаленою базою даних складається з двох частин: клієнтської і серверної (така технологія отримала назву «клієнтсервер»). Клієнтська частина програми, яка працює на комп'ютері користувача, забезпечує взаємодію з серверною програмою за допомогою запитів, що передаються на віддалений комп'ютер.

Серверна частина програми, що працює на віддаленому комп'ютері, приймає запити, виконує їх і пересилає дані клієнтської програмі. Запити представляють собою команди на спеціальній мові структурованих запитів (Structured Query Language, SQL).

Технологія клієнт-сервер означає такий спосіб взаємодії програмних компонентів, при якому вони утворюють єдину систему. Як видно з самої назви, існує якийсь клієнтський процес, що вимагає певних ресурсів, а також серверний процес, який ці ресурси надає. Зовсім необов'язково, щоб вони знаходилися на одному комп'ютері. Зазвичай прийнято розміщувати сервер на одному вузлі локальної мережі, а клієнтів - на інших вузлах.

У контексті бази даних клієнт управляє для користувача інтерфейсом і логікою програми, діючи, як робоча станція, на якій виконуються програми баз даних. Клієнт приймає від користувача запит, перевіряє синтаксис і генерує запит до бази даних на мові SQL або іншою мовою бази даних, що відповідає логіці програми. Потім передає повідомлення серверу, очікує надходження відповіді і форматує отримані дані для представлення їх користувачеві. Сервер приймає й обробляє запити до бази даних, після чого відправляє отримані результати назад клієнтові. Така обробка включає перевірку повноважень клієнта, забезпечення вимог цілісності, а також виконання запиту і поновлення даних. Крім цього підтримується управління паралельною і відновленням.

Архітектура клієнт-сервер має низку переваг:

- забезпечується більш широкий доступ до існуючих баз даних;
- підвищується загальна продуктивність системи: оскільки клієнти та сервери знаходяться на різних комп'ютерах, їх процесори здатні виконувати додатки паралельно. Налаштування продуктивності комп'ютера з сервером спрощується, якщо на ньому виконується тільки робота з базою даних;
- знижується вартість апаратного забезпечення; досить потужний комп'ютер з великим пристроєм зберігання потрібен тільки серверу - для зберігання і управління базою даних;
- скорочуються комунікаційні витрати. Програми виконують частину операцій на комп'ютерах клієнтів і посилають через мережу тільки запити до баз даних, що дозволяє значно скоротити обсяг пересилаються по мережі даних;
- підвищується рівень несуперечності даних. Сервер може самостійно управляти перевіркою цілісності даних, оскільки лише на ньому визначаються і перевіряються всі обмеження. При цьому кожному з додатком не доведеться виконувати власну перевірку;

Архітектура клієнт-сервер природно відображається на архітектуру відкритих систем.

2.3 Інфологічна модель бази даних, її види, проектування

Основою бази даних є модель даних. Інформаційно-логічна (міфологічна) модель предметної області відбиває предметну область у вигляді сукупності інформаційних об'єктів та їх структурних зв'язків.

Інформаційний об'єкт – це опис деякої сутності (явища, реального об'єкта, процесу) у вигляді сукупності логічно пов'язаних реквізитів. Інформаційний об'єкт має безліч реалізацій - примірників.

Бази даних також можуть бути ієрархічними, мережними і реляційними.

Ієрархічна (деревоподібна) модель даних представляє собою ієрархію елементів, які називаються вузлами. Вузол – це сукупність атрибутів даних, що описують інформаційний об'єкт. На самому верхньому рівні є тільки один вузол – корінь. Кожен вузол крім кореня пов'язаний лише з одним вузлом на більш високому рівні, що зветься вихідним вузлом для даного вузла. Кожен вузол може бути пов'язаний з одним або декількома вузлами більш низького рівня, що називаються породженими (підлеглими). Вузли, що не мають породжених вузлів, називаються листям. Кількість дерев у базі даних визначається числом кореневих записів. До кожного запису бази даних існує тільки один шлях.

Мережева модель також ґрунтується на поняттях вузол, рівень, зв'язок. Мережева модель даних – це модель, у якій породжений вузол може мати більше одного вихідного вузла. У мережевій структурі будь-який елемент будь-якого рівня може бути пов'язаний з будь-яким іншим елементом.

Реляційна модель бази даних складається з одного або декількох файлів, кожен з яких відповідає одній таблиці.

Основна мета проектування бази даних – це скорочення надмірності збережених даних, а отже, економія обсяг використаної пам'яті, зменшення витрат на багаторазові операції оновлення надлишкових копій та усунення можливості виникнення суперечностей через зберігання в різних місцях відомостей про одному і тому ж об'єкті. У разі створення бази даних слід дотримуватися методології нормалізації відносин.

Процес проектування інформаційних систем є досить складним завданням. Він починається з побудови інфологічної моделі даних, тобто ідентифікації сутностей. Потім необхідно виконати наступні кроки процедури проектування даталогічної моделі, тобто міфологічна модель повинна бути: відображена в комп'ютеро-орієнтовану дата логічну модель, «зрозумілу» СУБД.

Уявити предметну область у вигляді сукупності окремих незалежних один від одного об'єктів, кожен з яких буде описуватися своєю таблицею.

Для кожної таблиці визначити ключові поля; встановити зв'язки між таблицями; для кожного зв'язку визначити тип.

Розробити структуру кожної таблиці: перелік полів, їх типи та властивості.

Заповнити таблиці даними.

Розробити необхідні запити до БД, вхідні і вихідні форми та звіти.

Передбачити можливість автоматизації часто виконуваних дій шляхом створення макросів і програмних модулів

2.4 Принципи проектування ІС

Під проектуванням автоматизованих інформаційних систем розуміється процес розробки технічної документації, пов'язаний з організацією системи одержання і перетворення вихідної інформації в результативну, тобто з організацією автоматизованої інформаційної системи. Документ, отриманий в процесі проектування, носить назву проект. Метою проектування є підбір технічного та формування інформаційного, математичного, програмного та організаційно-правового забезпечення.

Успішна робота ІС в першу чергу визначається якістю проектування, саме при проектуванні створюється система, здатна функціонувати при постійному її вдосконаленні.

Проектування та функціонування систем ґрунтується на системотехнічних принципах, що відображають найважливіші положення загальної теорії систем, системного проектування та інших наук, що забезпечують надійність експлуатації та економічність, як при проектуванні, так і при використанні систем.

Принцип системності або системний підхід. Суть в тому, що кожне явище розглядається у взаємозв'язку з іншими. Системний підхід зосереджує увагу на об'єкті як на єдине ціле, а не на його частинах, як би зовсім вони не виконували свої функції. Системний підхід пов'язаний із загальною активністю системи для досягнення мети. Основні етапи формування системи:

- визначення мети;
- визначення вимог до системи (визначення меж об'єкту);
- визначення функціональних підсистем, їх структури та завдань у загальній системі управління;
- виявлення та аналіз зв'язків між підсистемами;
- встановлення порядку функціонування та розвитку всієї системи в цілому.

Безперервний розвиток інформаційних систем (ІС) – передбачає, при створенні інформаційних технологій повинна бути закладена можливість швидкого і без великих витрат на перебудову зміни і нарощування ІТ при зміні та розвитку об'єкту.

Працюючи передбачає можливість взаємодії ІС різних рівнів і видів у процесі їх спільного функціонування.

Стандартизація та уніфікація – припускає використання типових, уніфікованих і стандартних рішень при створенні і розвитку ІС (типових програмних продуктів, уніфікованої документації, техніки).

Принцип ефективності – раціональне співвідношення між витратами на створення та експлуатацію і ефектом від функціонування створюваної системи.

Інтеграція – це об'єднання в єдиний технологічний процес процедур збору, передачі, накопичення, зберігання інформації, і процедур формування управлінських рішень.

Автоматизації інформаційних потоків та документообігу, що досягається шляхом використання технічних засобів збору, реєстрації, обробки даних, створення первинних і результативних документів, а також засобів передачі даних на будь-які відстані.

2.5 Побудова моделі даних

Робота проектувальників бази даних в значній мірі залежить від якості інформаційної моделі. Інформаційна модель не повинна містити ніяких незрозумілих конструкцій, які не можна реалізувати в рамках вибраної СУБД.

Слід зазначити, що інформаційна модель створюється для того, щоб на її основі можна було побудувати модель даних, тобто повинна враховувати особливості реалізації вибраної СУБД. Якщо ті чи інші особливості СУБД не дозволяють відобразити в моделі даних те, що описує інформаційна модель, виходить, треба змінювати інформаційну модель.

Побудова логічної і фізичної моделей даних є основною частиною проектування бази даних. Отримана в процесі аналізу інформаційна модель спочатку перетворюється в логічний, а потім у фізичну модель даних. Після цього для розробників інформаційної системи створюється пробна база даних. З нею починають працювати розробники коду. В ідеалі до моменту початку розробки модель даних повинна бути стійка. Проектування бази даних не може бути відірвано від проектування модулів та програм, оскільки бізнес-правила можуть створювати об'єкти в базі даних, наприклад серверні обмеження (constraints), а також збережені процедури і тригери, - у цьому випадку часто кажуть, що частина бізнес-логіки переноситься до бази даних. Проектування моделі даних для кожної СУБД містить свої особливості, проектні рішення, які дають гарний результат для однієї СУБД, але можуть виявитися зовсім неприйнятними для іншої.

2.6 Побудова автоматизованої інформаційної системи

Автоматизовану систему інформації можна розглядати з двох точок зору:

1. Як мережа інформаційних служб (головний інформаційно - обчислювальний центр, локальні обчислювальні центри виробництв, цехів та інших підрозділів, автоматизовані робочі місця й інші складові) з розміщеними в ній масивами зберігання інформації, документами, технічними засобами реєстрації, зберігання, передачі, обробки, представлення інформації, програмним забезпеченням, методичним забезпеченням (інструкціями для користувачів, положеннями про підрозділи) та іншими видами забезпечення;
2. Як потоки різної інформації, що циркулюють всередині організації, які необхідно структурувати, відсортувати і оптимізувати.

Якщо говорити про систему інформації, як про мережу інформаційних служб, то при її формуванні виділяють два основних етапи:

1. Формування моделі, що відображає можливі варіанти проходження інформації
2. Оцінка моделі і вибір найкращого варіанту шляху проходження інформації.

При цьому використовують методику системного аналізу. Ці два етапи містять певні внутрішні етапи.

Формування моделі, що відображає можливі варіанти проходження інформації в автоматизованій інформаційній системі. Відмежування системи від середовища ("перерахування" елементів системи).

Під етап може виконуватися із застосуванням методу "мозкової атаки", а в реальних умовах - методів типу комісій, семінарів та інших форм колективного обговорення, в результаті якого визначається певний перелік елементів майбутньої системи. До складу таких комісій повинні входити розробники і майбутні користувачі АІС.

2.7 Відкрита інформаційна система

Вибір технологій і стандартів при розробці системи рішення, що визначає успіх застосування системи, можливості реінжинірингу, термін служби, згодом перехід на нові більш прогресивні технології.

Термін «реінжиніринг» був введений провідними американськими фахівцями в області консалтингу Майклом Хаммером (Michael Hammer) і Джеймсом Чампі (James Champy).

Реінжиніринг – це перебудова (перепроєктування) ділових процесів для досягнення радикального, стрибкоподібного поліпшення діяльності фірми. Це - комплексна зміна діючих бізнес процедур і перепланування операцій, спрямована на кардинальне зниження витрат, підвищення якості продукції та більш повне задоволення потреб споживачів.

Кваліфіковані розробники ІС орієнтуються на стандарти відкритих систем. Відкритими називаються системи, які можуть розвиватися за рахунок

технічних чи програмних засобів, створених на основі технології, що задовольняють вимогам міжнародних стандартів. Головна перевага відкритих систем у тому, що вони забезпечують інтероперабельність технічних і програмних засобів різних виробників, тобто спільне їх використання в єдиному проекті. Дотримання стандартів дозволяє забезпечити інтероперабельність як системних компонентів, так і різних взаємодіючих систем, міжплатформову сумісність програмного забезпечення, додатків і даних в технологіях баз даних і т.д.

РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОЕКТУ

3.1 Призначення розробки

Метою дипломної роботи є розробка та створення програмного продукту «Автоматизованої інформаційної системи для обліку робіт на станції технічного огляду автотранспорту». Це програмне забезпечення призначене для створення обліку замовлень і контролю за якістю виконаної роботи на СТО і обліку заробітної плати згідно виконаних обсягів робіт. Розроблений програмний продукт повинен забезпечувати необхідний набір функцій, даючи можливість доступу до інтерфейсних об'єктів користувацького подання програмного забезпечення за допомогою миші та клавіатури.

3.2 Середовище розробки

Вбудована в Microsoft Access підтримка роботи з базами даних (БД) є ключовою можливістю цього програмного середовища. Дуже багато програмістів більшу частину свого часу витрачають на створення коду, в середовищі розробки Delphi, але за допомогою Microsoft Access можна полегшити цю процедуру.

Нижченаведені розділи — це короткі описи частин бази даних, створеної в MS Access.

- 1.Таблиці;
- 2.Форми;
- 3.Звіти;
- 4.Запити;
- 5.Макроси;
- 6.Модулі;
- 7.Таблиці.

Таблиці

Таблиця бази даних схожа на електронну таблицю, в якій дані зберігаються в рядках і стовпцях. В результаті зазвичай досить легко імпортувати електронну таблицю до таблиці бази даних. Головна відмінність між збереженням даних в електронній таблиці та базі даних — це спосіб упорядкування даних.

Щоб забезпечити максимальну гнучкість бази даних, дані необхідно впорядкувати в таблицях, щоб позбутися зайвих елементів. Наприклад, якщо потрібно зберігати дані про працівників, відомості про кожного працівника необхідно один раз ввести в таблиці, яка настроєна лише для розміщення даних про працівників. Цей процес називається оптимізацією.

Кожний рядок у таблиці називається записом. Записи — це місце розташування окремих елементів даних. кожен запис складається з одного або кількох полів. Поля відповідають стовпцям у таблиці. Наприклад, можна створити таблицю «Працівники», де кожен запис (рядок) зберігає відомості про окремого працівника, а кожне поле (стовпець) містить власний тип даних, наприклад ім'я, прізвище, адресу тощо. Поля мають містити певний тип даних: текст, дату або час, число або інший тип.

Ще один спосіб опису записів і полів: уявіть старий картковий каталог у бібліотеці. Кожна картка у ящику відповідає запису бази даних. кожен елемент даних на окремій картці (автор, назва тощо) відповідає полю бази даних.

Форми

Форми іноді називаються «екранами вводу даних». Це інтерфейси, які використовуються під час роботи з даними, тому вони часто містять кнопки для виконання різних команд. Можна створити базу даних без використання форм, просто редагуючи дані в таблицях даних. Проте більшість користувачів баз даних використовують форми для перегляду, введення та редагування даних у таблицях.

Форми пропонують простий у використанні формат роботи з даними, крім того, до них можна також додавати функціональні елементи, наприклад кнопки. Ці кнопки можна налаштувати для визначення даних, що відобразатимуться у формі, відкриття інших форм або звітів та для виконання низки інших завдань. Наприклад, є форма «Форма клієнта», у якій виконується робота з даними клієнта. Форма клієнта може містити кнопку, яка відкриває форму замовлення, де можна ввести нове замовлення цього клієнта.

Форми також дають змогу керувати способом взаємодії інших користувачів із даними бази даних. Наприклад, можна створити форму, яка відображає лише певні поля та дозволяє виконувати лише певні операції. Це допомагає захистити дані та забезпечує належне введення даних.

Звіти

Звіти використовуються для зведення та представлення даних у таблицях. Звіт зазвичай відповідає на певне питання, наприклад «Яку суму було отримано від кожного клієнта цього року?» або «У яких містах розташовані наші клієнти?». кожний звіт можна відформатувати таким чином, щоб він представляв дані найбільш зрозумілим способом.

Звіт можна запустити будь-коли, і він завжди відобразатиме поточні дані в базі даних. Звіти зазвичай мають формат для друку, але їх також можна переглядати на екрані, експортувати до іншої програми або надсилати електронною поштою.

Запити

Запити — це справжні робочі коники бази даних, які можуть виконувати багато різних функцій. Їх найпоширеніша функція — отримання певних даних із таблиць. Дані, які потрібно переглянути, зазвичай розташовані в кількох таблицях, і запити дають змогу переглянути їх в одній таблиці даних. Також, оскільки зазвичай не потрібно бачити всі записи одночасно, запити дозволяють додавати критерії для «фільтрування» даних, щоб переглядати лише потрібні записи. Запити часто виконують роль джерела записів для форм і звітів.

Певні запити є «оновлюваними», тобто дані в базових таблицях можна редагувати за допомогою таблиці даних запиту. Якщо дії виконуються з оновлюваним запитом, слід пам'ятати, що зміни насправді виконуються в таблицях, а не лише в таблиці даних запиту.

Запити поділяються на дві основні групи: запити на вибірку і запити на дію. Запит на вибірку просто отримує дані й робить їх доступними для використання. Результати запиту можна переглянути на екрані, роздрукувати або скопіювати до буфера обміну. Або можна використати результат запиту як джерело записів для форми чи звіту.

Запит на змінення, згідно з назвою, виконує з даними певне завдання. Запити на змінення можна використовувати для створення нових таблиць, додавання даних до наявних таблиць, оновлення або видалення даних.

Макроси

Макроси в програмі Access можна вважати спрощеною мовою програмування, яку можна використовувати для додавання функціональності до бази даних. Наприклад, можна вкласти макрос до кнопки форми, щоб запускати макрос у разі натискання цієї кнопки. Макроси містять дії, які виконують завдання, наприклад відкривають звіт, виконують запит або закривають базу даних. Більшість операцій із базою даних, які виконуються вручну, можна зробити автоматичними за допомогою макросів, тому вони можуть бути корисним засобом економії часу.

Модулі

Модулі, як і макроси, — це об'єкти, які можна використовувати для додавання функціональності до бази даних. Проте, якщо макроси Access створюються за допомогою вибору зі списку дій макросу, модулі пишуться мовою програмування Visual Basic для застосунків (VBA). Модуль — це збірка декларацій, інструкцій і процедур, які зберігаються разом. Модуль може бути модулем класу або стандартним модулем. Модулі класу додаються до форм або

звітів і зазвичай містять процедури, характерні для форми чи звіту, до яких вони додаються. Стандартні модулі містять загальні процедури, не пов'язані з жодним іншим об'єктом. Стандартні модулі відображаються в області переходів у розділі Модулі, проте модулі класу там не відображаються.

3.3 Вікно коду програми.

Вікно коду призначене для створення і редагування тексту програми. Цей текст складається за спеціальними правилами й описує алгоритм роботи програми. Сукупність правил запису тексту називається мовою програмування. У системі Access використовується мова програмування Visual Basic, що замінила мову програмування Access Basic, яка використовувалась в версіях 1 і 2 Microsoft Access. В останні роки Visual Basic стала спільною мовою програмування для пакету Microsoft Office. Ця мова має розвинене середовище, прекрасні інструменти від лагодження, які переконують, що розроблена програма функціонує належним чином.

Всі програми Visual Basic зберігаються в модулях. База даних може містити два види модулів: стандартні модулі, які є об'єктами бази даних, і модулі форм або звітів, які є частиною їх об'єктів.

Побачити стандартні модулі відкритої бази даних можна, якщо у вікні бази даних клацнути на кнопці *Модулі*.

Для полегшення створення процедур обробки подій, які виникають у формі або звіті, Access представляє модуль класу, який зв'язаний з кожною формою або звітом. В модулі класу форми або звіту містяться спеціальним чином іменовані процедури обробки подій, особисті процедури, доступні тільки в модулі класу, і загальні процедури, які можна викликати як методи класу. Відредагувати модуль форми або звіту можна, відкривши відповідну форму або звіт в режимі конструктора і клацнувши на кнопці Програма на панелі інструментів.

Вікно програми не може закріплюватись, і воно завжди відображається в тій частині робочої області, яка не зайнята закріпленими вікнами.

У верхній частині вікна програми під рядком заголовку міститься два списки, які можуть розкриватись.

Object (об'єкт). При редагуванні модуля форми чи звіту можна вибрати в цьому списку форму або звіт, їх розділ або елемент керування, який здатний

генерувати подію.

Procedure (процедура). Процедура, яка вибрана в цьому списку, виводиться у вікні програми. При відкритті стандартного модуля цей список містить всі включені в нього процедури, які відображаються в алфавітному порядку.

Разом з вікном коду звичайно активізується) браузера Code Explorer, що полегшує пошук потрібних елементів у випадку, коли у вікні набрано багато рядків коду .

Одне з вікон програмного коду нашої інформаційної системи має наступний вигляд.(Рис. 3.1)

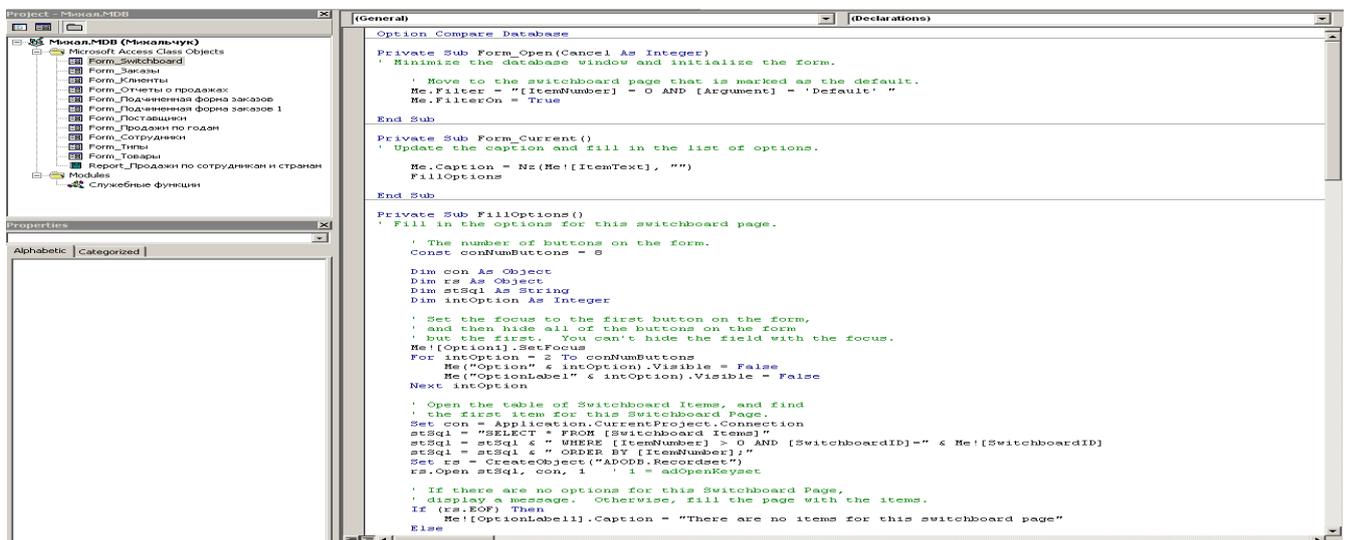


Рис 3.1 Вікно коду з розташованим ліворуч браузером Code Explorer

Подвійний щиклик мишею на елементі у вікні браузера приводить до позиціонування текстового покажчика у вікні коду на опис відповідного об'єкта чи його перше згадування. При розробці нескладних навчальних програм можна закрити вікно браузера маленькою кнопкою в його верхньому правому куті. Відновити закрите вікно можна після щиклика правою кнопкою миші по вікну коду і вибору опції view Explorer. Вікно браузера розташовується на допоміжній панелі і має керуючу смугу у верхній частині рамки. “Схопивши” її мишею, можна перетягнути вікно на будь-яке вільне місце чи екрана “причалити” його до вікна Інспектора об'єктів.

3.4 Головна форма програми

На головній формі програми представлено меню, яке складається з вкладок: Головна, Довідники, Запити, Звіти, Автор (Рис 3.2).

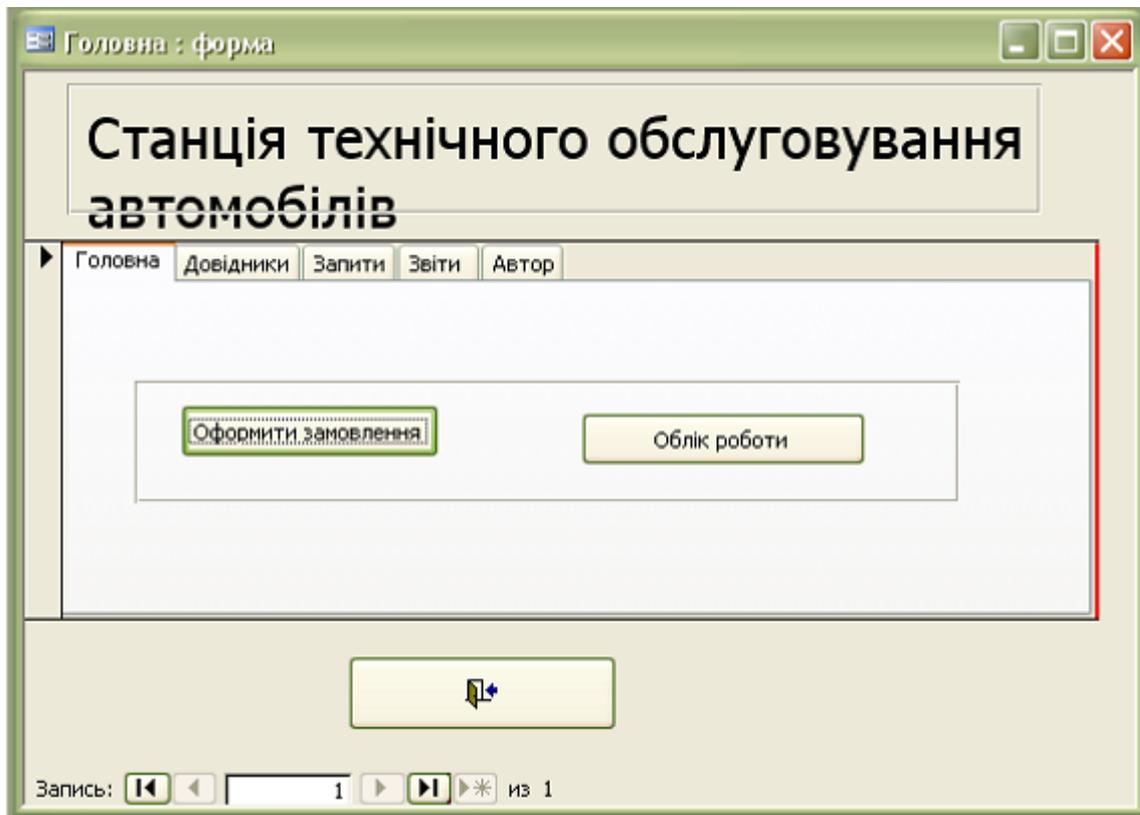


Рис 3.2 Головна сторінка

Головна сторінка складається з кнопок: оформлення звіту (Рис 1.1), облік роботи (Рис. 3.2)

На формі Оформлення замовлення реалізовано доступ до таблиці для оформлення замовлення, кнопка Клієнти виводить прізвища всіх клієнтів та дані про автомобілі за конкретний період. Кнопка Заява формує заяву на замовлення (Рис3.3).

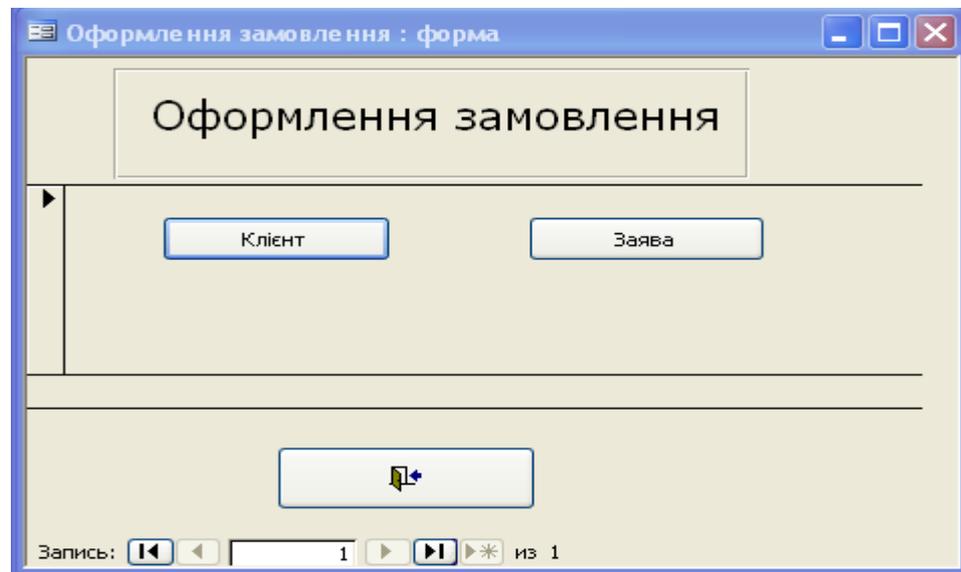


Рис 3.3 форма для оформлення замовлення

Дана форма відображає облік проведених робіт та замовника цих робіт. Тобто, прізвище співробітника, прізвище клієнта, марку автомобіля та всі виконані роботи (Рис 1.2).

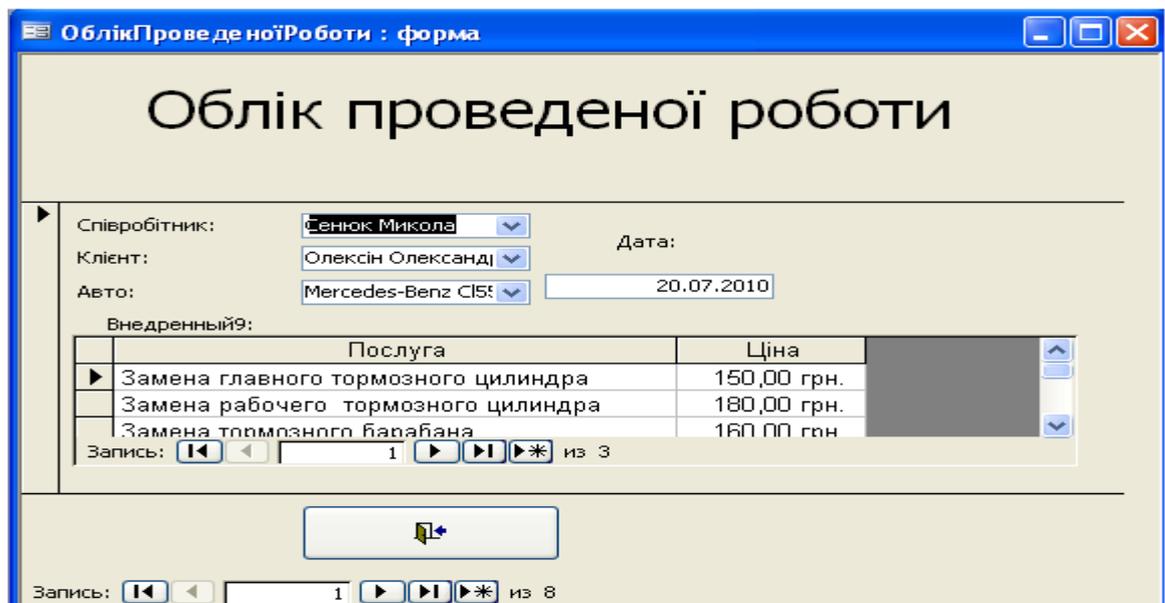


Рис 3.4 Облік проведених робіт

Вкладка Довідники (Рис 3.1) складається з таких кнопок: Відділи (Рис 3.2), Прайс-лист (Рис 3.3), Проведена робота (Рис 3.4), Клієнти (Рис 3.5) та Співробітники (Рис 3.5).

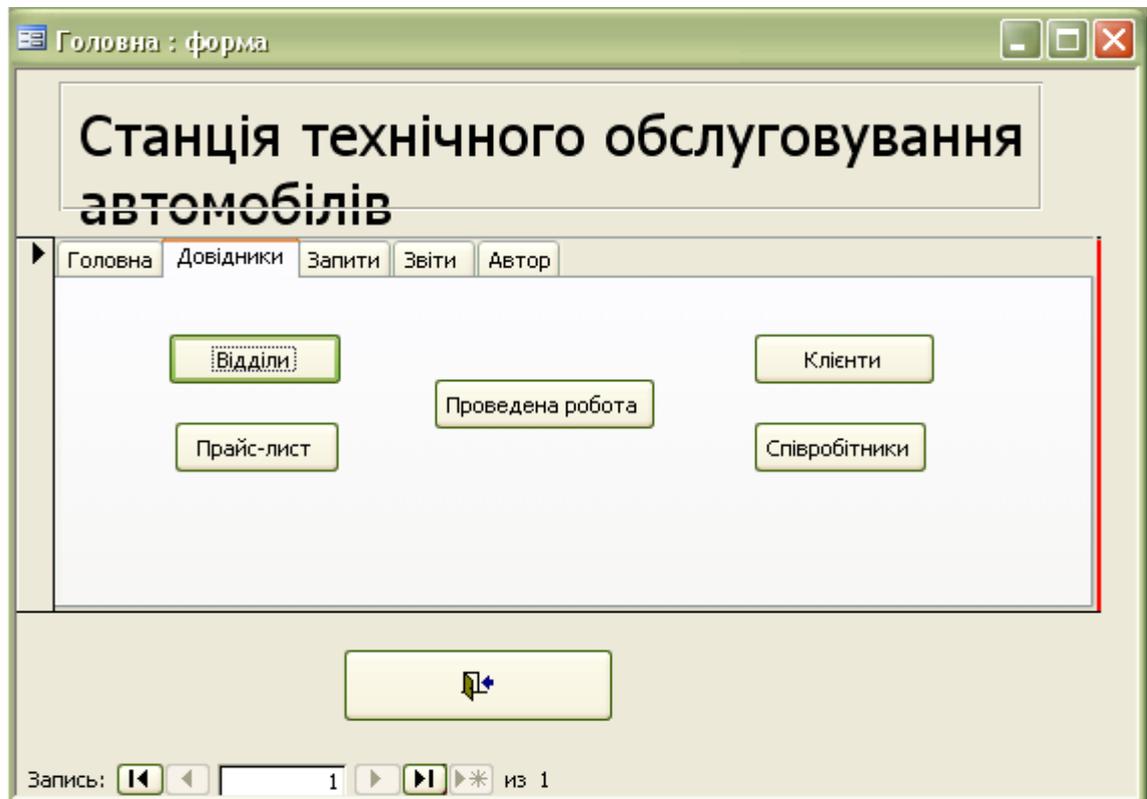


Рис 3.5

На даній формі відображається назви відділів та їх розташування (Рис 3.6).

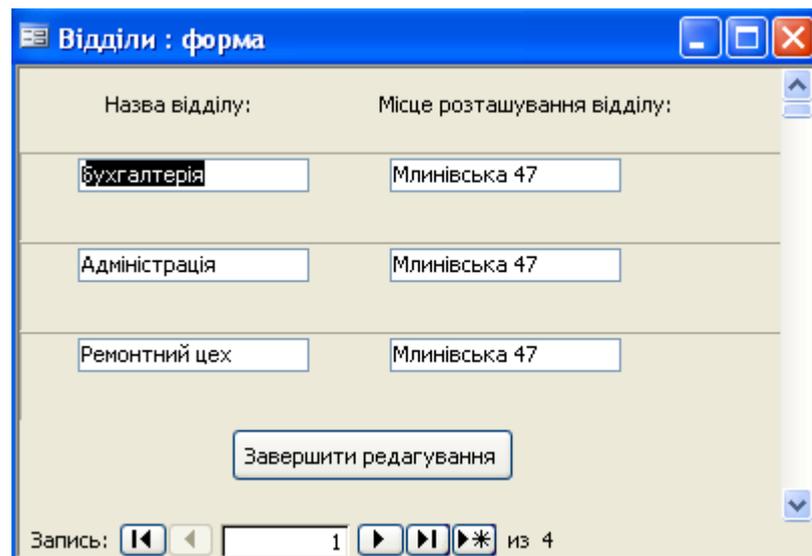


Рис 3.6

На формі Клієнти зображено таблицю, яка виводить ініціали клієнта, дані про його автомобіль (марка та рік випуску)(Рис 3.9).

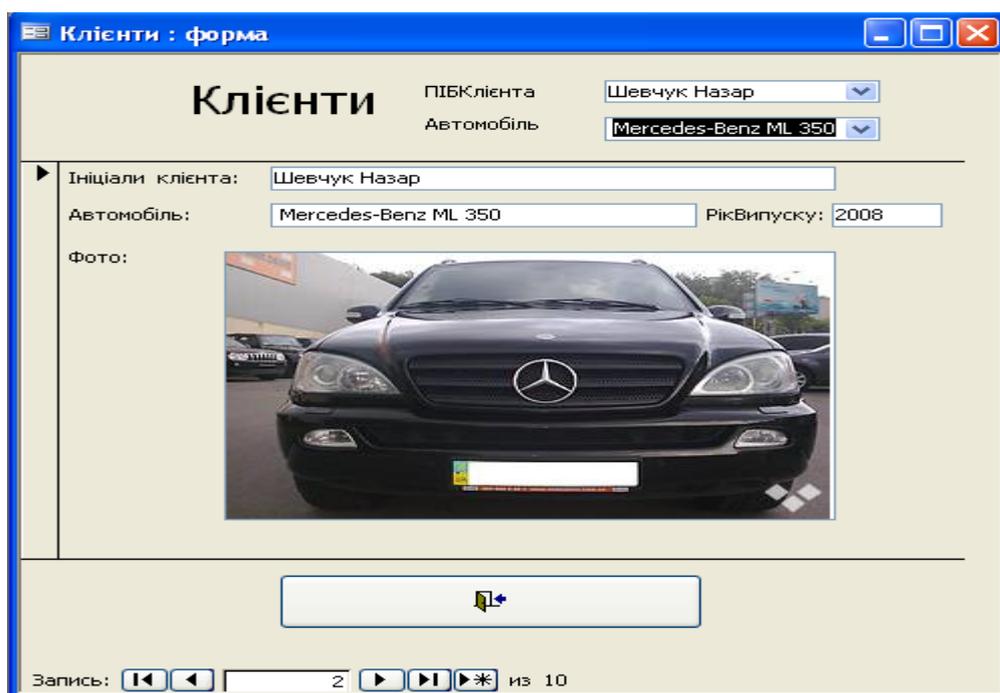


Рис 3.9

Дана форма відображає особисті дані співробітника, зокрема ініціали співробітника, дату народження, адресу, домашній телефон, адресу електронної пошти, стать, оклад працівника та дату прийняття його на роботу (Рис 3.10).

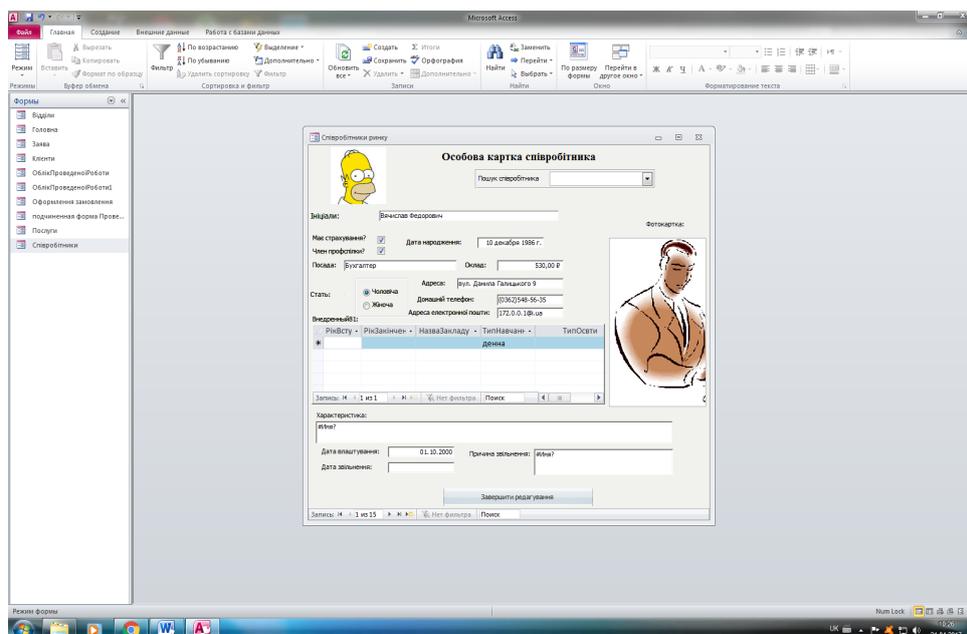


Рис 3.11

Вкладка запити складається з таких кнопок: Жінки відділів (див додатки 1), Авто за роком (див додатки 2), Кількість співробітників (див додатки 3), обслуговування клієнтів (див додатки 4), Оплачені ремонти (див додатки 5), Стаж (див додатки 6) та Клієнти за період (див додатки 7)

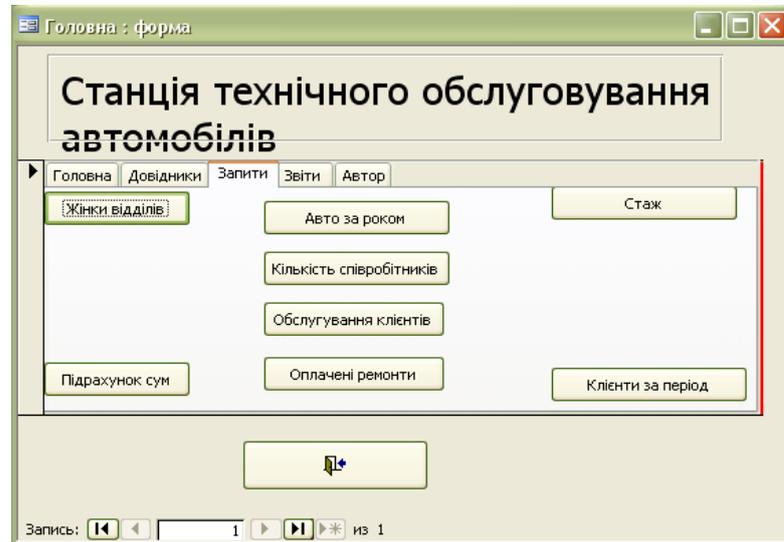


Рис 3.12

Вкладка звіти складається з таких кнопок: Відділи (див додатки 8), Прайс-лист (див додатки 9), Замовлення (див додатки 10), Засоби зв'язку (див додатки 11), Співробітники (див додатки 12), Оклади (див додатки 13), Стаж співробітників (див додатки 14), Авто за роком випуску (див додатки 15), Оплати клієнтів(див додатки 16).

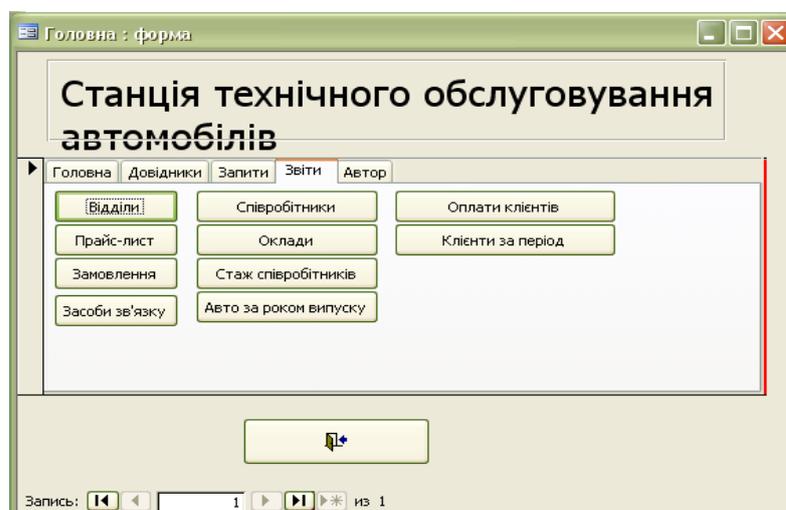


Рис. 3.13

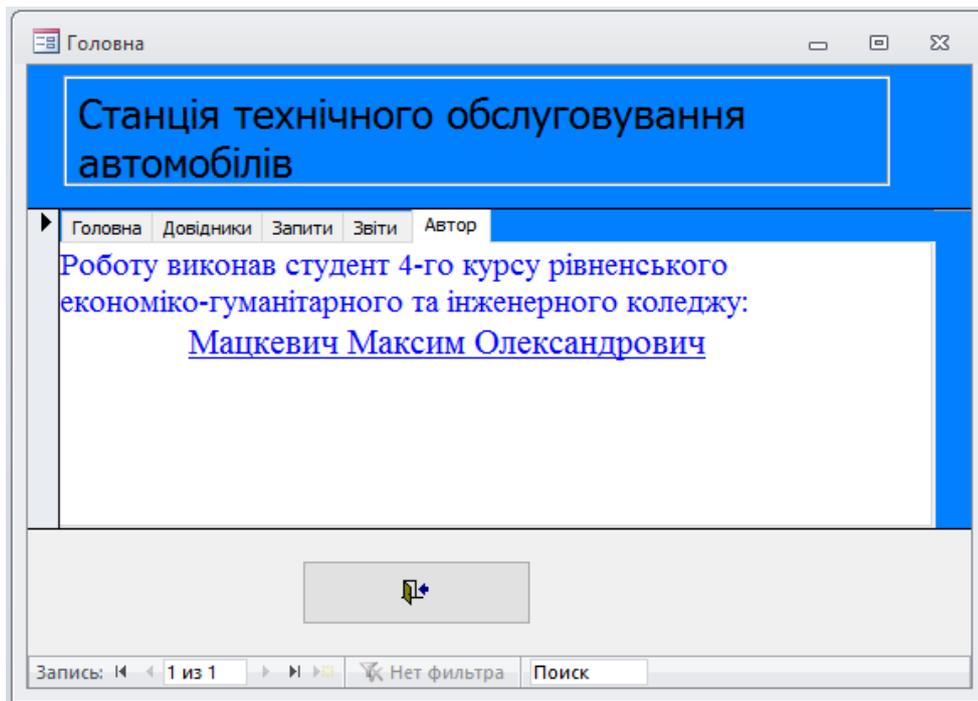


Рис 3.14

Вкладка Автор містить відомості про розробника програми.

3.4 Інструкція системного програміста

Системний адміністратор повинен слідкувати за правильною роботою програмного забезпечення, зокрема за програмою MS Access, всіх її додатків, та модулів.

Для виконання завдання використовувалась наступна версія програми.

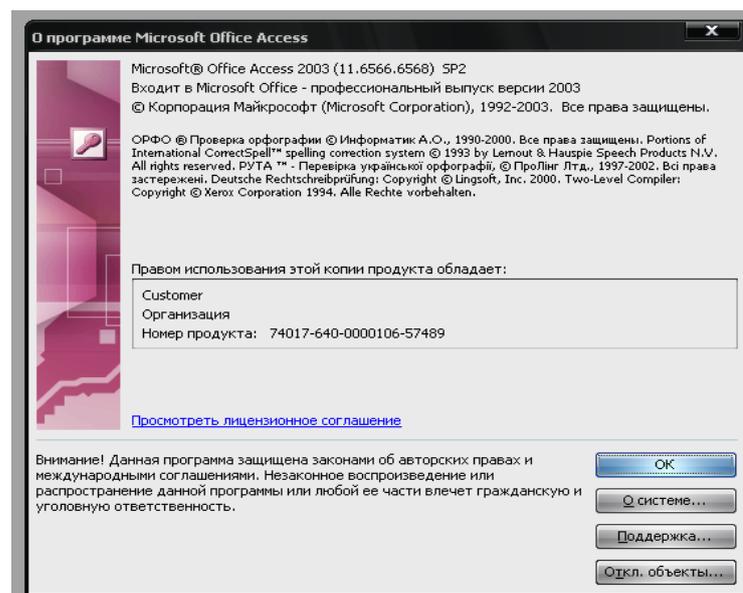


Рис. 3.15

3.5 Інструкція користувача

Користувач працює з наступними формами:

- Головна: оформити замовлення, облік роботи;
- Довідники: відділи, прайс-лист, проведена робота, клієнти, співробітники;
- Запити: журнал відділів, авто за роком, кількість співробітників, обслуговування клієнтів, оплачені ремонти, стаж, клієнти за період;
- Звіти: відділи, прайс-лист, замовлення, засоби зв'язку співробітники, оклад, стаж співробітників, авто за роком випуску, оплата клієнтів;
- Автор

Запуск програми виконується активізацією файлу «db2.mln» на жорсткому диску комп'ютера.

3.6 Економічне обґрунтування проекту

Створення конкурентоспроможної продукції передбачає проведення техніко-економічних розрахунків, з метою визначення оптимальних шляхів вирішення технічних питань під час допроектної підготовки, проектування, впровадження, експлуатації, обслуговування та оновлення інформаційної системи. Тому невід'ємною частиною інженерної освіти є формування знань і набуття практичних навичок в вирішенні таких питань: техніко-економічне порівняння варіантів проектних рішень; розрахунок собівартості програмного забезпечення; розрахунок кошторису витрат на науково-дослідницькі роботи; розрахунки економічного ефекту внаслідок впровадження і експлуатації інформаційної системи, проведення науково-дослідних робіт.

Метою виконання економічної частини дипломної роботи бакалавра є доведення економічної доцільності розробленого програмного продукту. Для цього проведено техніко-економічне порівняння проектних рішень, яке складається з таких етапів:

1. Розрахунок капіталовкладень для варіантів, що порівнюються.

2. Розрахунок щорічних експлуатаційних витрат для варіантів, що порівнюються.
3. Розрахунок звідних економічних показників, на підставі якого робиться висновок про доцільність впровадження об'єкту дипломного проектування.

1) Розрахунок витрат на видаткові матеріали наведений в таблиці 4.1.

Таблиця 4.1–Витрати на видаткові матеріали (М)

Матеріал	Сума, грн.
Канцелярські прилади	120
Програмне	15
Інші матеріали	320
Разом	455

2) Заробітна плата – матеріальна винагорода, що отримується працівником відповідно до його роботи.

Заробітна плата рахується за формулою: $Z = (\text{Max}Z / \text{MaxTimes}) * \text{spentTimes}$, де $\text{Max}Z$ –бюджет проекту, MaxTimes - максимальна кількість годин в місяць, spentTimes – кількість витрачених годин на розробку проекту.

- Бюджет проекту 10 000 тис грн
- Максимальна кількість годин в місяць 160 годин
- Кількість витрачених годин на розробку проекту 72 години

Заробітна плата (Z) програміста складає – 4500 грн.

3) Накладні витрати (N_v) – це додаткові витрати до основних витрат для забезпечення процесів виробництва. До накладних витрат відносяться: відрахування на соціальне страхування і інші обов'язкові платежі; заробітну плату адміністративно-управлінського персоналу; витрати, пов'язані з втратами від браку і простоїв, і інше.

$$N_v = Z * 20\% = (4500 * 20) / 100 = 900$$

4) Послуги сторонніх організацій виразилися в наданні фірмою «Kyivstar» доступу до глобальної інформаційної мережі Internet. Помісячна оплата цієї послуги складає 75 грн (P).

5) Собівартість (S) — всі витрати, затрачені на виробництво і реалізацію продукції або послуги.

$$S = Z + N_v + M + P$$

$$S = 4500 + 900 + 455 + 75 = 5930$$

6) Податок на додану вартість (ПДВ) – непрямий податок, форма вилучення до бюджету частини доданої вартості, яка створюється на всіх стадіях процесу виробництва товарів, робіт і послуг і вноситься до бюджету по мірі реалізації.

$$\text{ПДВ} = (5930 * 20) / 100 = 1186 \text{ грн}$$

Кошторис витрат наведений у таблиці 4.2

Таблиця 4.2 – Кошторис витрат

Статті затрат	Вартість
Зарплата	4500
Матеріали	555
Накладні	900
Послуги сторонніх організацій	75
Собівартість	5930
Всього	5930
ПВД	1186
Всього	7116

Випускна вартість складає – 7116 грн.

Висновки

Цінність даної дипломної роботи полягає у створенні інформаційної системи, що вирішує поставлені задачі за допомогою механізму реляційних баз даних. Створена автоматизована інформаційна система забезпечує ефективне збереження та цілісність даних, за допомогою форм, кожна з яких відповідає певній таблиці чи сукупності таблиць, дозволяє легке введення, редагування, пошук даних в базі, за допомогою механізму запитів дозволяє пошук та відбір даних в базі за певними критеріями і ці дані можна перенаправити у відповідний звіт, який можна виддрукувати.

Створення автоматизованої інформаційної системи для обліку робіт станції технічного огляду автотранспорту реалізовано для контролю виконаних робіт на станції СТО. Концепція системи передбачає модель програми, що має виконувати наступні функції.

- ведення даних про СТО;
- ведення даних про виконані роботи;
- пошук даних за певними критеріями;
- перегляд звітності та можливість їх роздруку.

Організація станції технічного огляду автомобілів полягає передусім у розробці системи кодування для полегшення роботи на цій станції. Готовим об'єктом вважається закінчений конструктивний пристрій з усім належним до нього устаткуванням. Слід відмітити важливу річ: розроблена в даній дипломній роботі інформаційна система справді автоматизує роботу фірми, але автоматизує в тому плані, що позбавляє необхідності ведення паперової документації та роботи з нею, полегшує ведення обліку комплектуючих, полегшує пошук конкретного. Але, головна мета даної дипломної роботи досягнута, а саме розроблено інформаційну систему, яка підтримує модель реляційних баз даних, забезпечуючи збереження, надійність та цілісність введених даних.

Отже, під час проведених досліджень було автоматизовано і полегшено ведення обліку роботи СТО. Це дає можливість швидко працювати з клієнтами, вести облік всіх виконаних робіт, перевіряти кількість звернень клієнтів до фірми, переглядати виконані роботи співробітників та вести облік оплачених ремонтів. Тобто, завдання поставлені при написанні дипломної роботи виконані в повному обсязі.

Джерела літератури

1. Андон П.І., Лаврищева Е.М. «Основи програмної інженерії», Знання, К., 2013.
2. Бобровський С. Acctss, навчальний курс, Пітер, 2015.
3. Бойко В.В., Савінков В.М. "Проектування інформаційної бази автоматизованої системи на основі СУБД" М.: Фінанси і статистика, 2009.
4. Вендров А.М. «Проектування програмного забезпечення», Фінанси й Статистика, М., 2014.
5. Л.В. Кокорева, О.Л. Перевозчикова "Діалогові системи і представлення знань", М., 2012.
6. Марка Д.А., Мак-Гоуен К. «Методологія структурного аналізу і проектування», Мета Технологія, М., 2013.
7. Марко Кент «Access для професіоналів», Пітер, С-П., 2005.
8. Ріхтер Джеффри "Windows для професіоналів", Російська редакція, С-П., 2013.
9. Фарафонов В.В. Access. Програмування високого рівня: підручник для вузів. 2011
10. Фокс Д. «Програмне забезпечення та його розробка», Мир, М., 2012.
11. Хомоненко А. Access, програмування, ВНУ, 2009.
12. Еліенс А. «Принципи об'єктно-орієнтованої розробки програм», Вільямс, М., 2012.
13. Microsoft Access 10 SR-2 Корпорація Microsoft 1983-2015.
14. Вейскас Дж.Эффективная работа с Microsoft Access 2000. – СПб.: Питер, 2000. –1040 с.
15. Праг Кері Н., Ірвін Майкл Р. Access 2002. Біблія користувача.: пер. з англ. – М.: вид „Вільямс”, 2003.
16. Пушкар О.І. Інформатика. Комп'ютерна техніка. Комп'ютерні технології.- Київ: Академія, 2001.

17. Шпортько О.В. Практикум з інформатики та комп'ютерної техніки . Частина II. Прикладна інформатика: Навч.-метод. Посібник. – Рівне: РДГУ, 2003.
18. Шпортько О.В. Тези доповідей та лекцій факультету документальних комунікацій та менеджменту.
19. Дейт К. Введение в системы баз данных, - Москва: 2000, -845 с.
20. Руденко В.Д., Макачук О.М., Патланжоглу М.О. Практичний курс інформатики, - К: 1997,- 300 с.
21. Послед Борис Access 2000. Базы данных и приложения. Лекции и упражнения. – К: ДиаСофт. – 2000, -512 с.
22. Збірник нормативних документів з безпеки життєдіяльності. Витяг з державних санітарних правил і норм.- Під редак. проф.Сачкова Л.С.,- Київ: 2000,- с.738-739.
23. Манов Г.М. Человек и дисплей. – Ленинград: 2010.
24. Сибаров Ю.Г. Охрана труда в вычислительных центрах. – Москва : 2002.
25. Дж. Тельман, "Основы систем баз даних", Москва, Фінанси і статистика', 2004г.
26. Дейт До., "Введення в системи баз даних", Москва, 'Наука', 2009 р.
27. Когловський м.Р., "Технологія баз даних на персональних ЕОМ", Москва, 'Фінанси і статистика', 2006 р.
28. Шумаков П. В. "Delphi 3.0 і створення баз даних". Москва 1997г.
29. Джон Матчо, Девід Р.Фолкнер. «Delphi» — пер. з англ. — М.:Бином, 2009
А.М.Епанешников., "Програмування в середовищі Delphi 2.0"
Дж. Мартін., "Організація баз даних в обчислювальних системах" М: Мир 2009.
30. С.М.Діго "Проектування і використання баз даних". Москва: Фінанси і статистика 2013.
31. Горев А., Ахаян Р., Макашаріпов С. «Ефективна робота з СУБД».СПб.:Питер, 2013—704 с., мул.

ДОДАТКИ

Додаток А

	Назва відділу	Прізвище, ім'я, по батькові
	Ремонтний цех	Воронова Галина
	Ремонтний цех	Крилова Інна
▶	Ремонтний цех	Ясенєва Алла
	Цех по фарбуванню	Бабкіна Галина
	Цех по фарбуванню	Белова Іванна
*		

Запись: 3 из 5

Додаток В

	Ініціали клієнта	Автомобіль	РікВипуску
▶	Смірнов Іван	Mitsubishi Pajero Wagon	2007
	Кукуруза Юрій	Toyota Camry	2007
	Альохін Дмитро	Mazda CX-7	2007
	Гарячий Олексій	Mercedes-Benz C1550	2007
*			

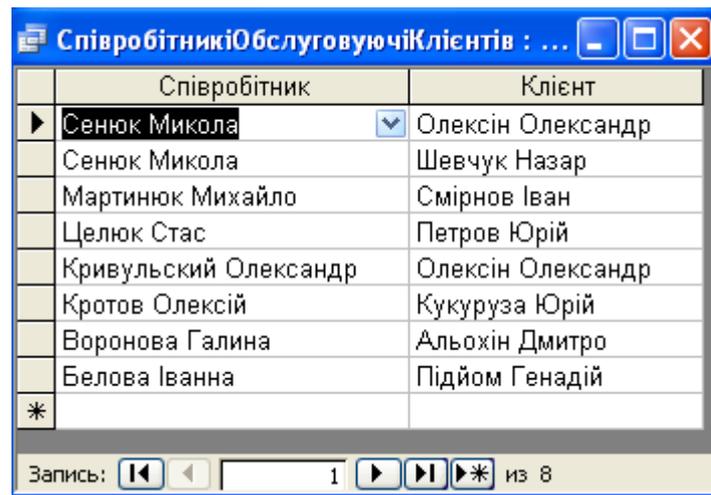
Запись: 1 из 4

Додаток С

	Назва відділу	Кількість співробітників
▶	Адміністрація	3
	Бухгалтерія	2
	Ремонтний цех	7
	Цех по фарбуванню	3

Запись: 1 из 4

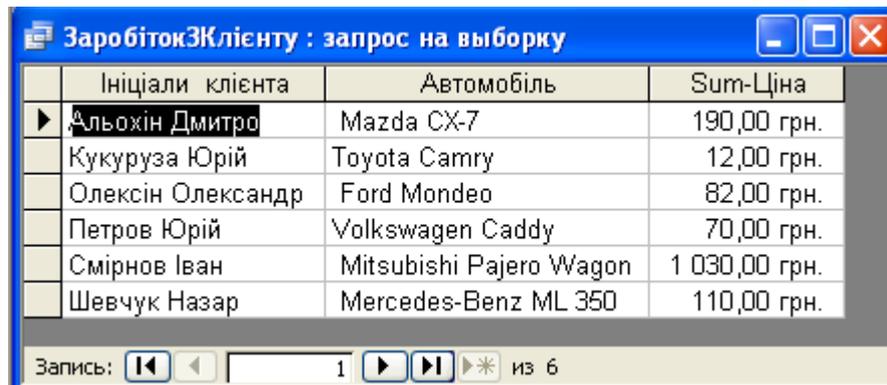
Додаток D



Співробітник	Клієнт
Сенюк Микола	Олексін Олександр
Сенюк Микола	Шевчук Назар
Мартинюк Михайло	Смірнов Іван
Целюк Стас	Петров Юрій
Кривульський Олександр	Олексін Олександр
Кротов Олексій	Кукуруза Юрій
Воронова Галина	Альохін Дмитро
Белова Іванна	Підйом Генадій
*	

Запись: 1 из 8

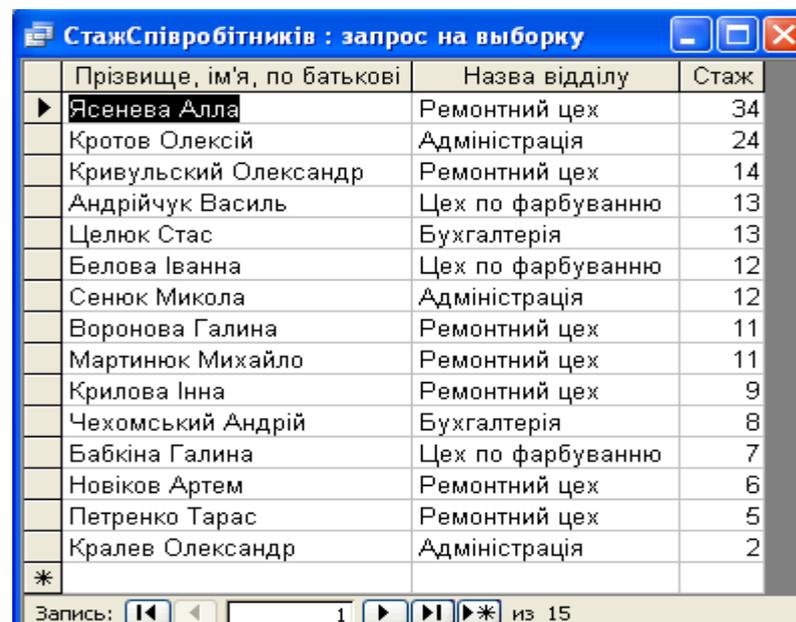
Додаток E



Ініціали клієнта	Автомобіль	Sum-Ціна
Альохін Дмитро	Mazda CX-7	190,00 грн.
Кукуруза Юрій	Toyota Camry	12,00 грн.
Олексін Олександр	Ford Mondeo	82,00 грн.
Петров Юрій	Volkswagen Caddy	70,00 грн.
Смірнов Іван	Mitsubishi Pajero Wagon	1 030,00 грн.
Шевчук Назар	Mercedes-Benz ML 350	110,00 грн.

Запись: 1 из 6

Додаток F



Прізвище, ім'я, по батькові	Назва відділу	Стаж
Ясенева Алла	Ремонтний цех	34
Кротов Олексій	Адміністрація	24
Кривульський Олександр	Ремонтний цех	14
Андрійчук Василь	Цех по фарбуванню	13
Целюк Стас	Бухгалтерія	13
Белова Іванна	Цех по фарбуванню	12
Сенюк Микола	Адміністрація	12
Воронова Галина	Ремонтний цех	11
Мартинюк Михайло	Ремонтний цех	11
Крилова Інна	Ремонтний цех	9
Чехомський Андрій	Бухгалтерія	8
Бабкіна Галина	Цех по фарбуванню	7
Новіков Артем	Ремонтний цех	6
Петренко Тарас	Ремонтний цех	5
Кралеv Олександр	Адміністрація	2
*		

Запись: 1 из 15

Додаток G

КлієнтиЗаПеріод : запрос на выборку

	Ініціали клієнта	Автомобіль	ДатаРеєстрації
▶	Олексін Олександр	Ford Mondeo	28.12.2009
	Петров Юрій	Volkswagen Caddy	15.06.2007
	Смірнов Іван	Mitsubishi Pajero Wagon	07.06.2009
*			21.06.2010

Запись: 1 из 3

Додаток H

АлфавітнийСписокВідділів

Алфавітний список відділів

<i>Назва відділу</i>	<i>Місце розташування відділу</i>
Адміністрація	Млинівська 47
Бухгалтерія	Млинівська 47
Ремонтний цех	Млинівська 47
Цех по фарбуванню	Млинівська 47

Страница: 1

Додаток I

Послуги

Послуги

Послуга	Ціна
Вставка профілактик пох амортиз.	40€
Вставка профітровацок резинозамс	70€
Діагностика	30€
Діагностика	30€
Діагностика	30€
Зам.Башмака колс.	130€
Зам.главного сцепления	75€
Зам.главного сцепления	70€
Зам.гофра	100€
Зам.датчика давления масла.	30€
Зам.заклейк ——— 1 шт.	70€
Зам.заклейк амортизат.	65€
Зам.капота статора	80€
Зам.компл. сцепления	380€
Зам.кулоск.	80€
Зам.масла змтп насос	55€
Зам.масла фрезера	45€
Зам.малшоса	70€

Страница: 1

Додаток J

Клієнт	Авто	Дата
Шевчук Назар	Mercedes-Benz ML 350	15.06.2009
Олександр Олександр	Mercedes-Benz CL550	20.07.2010
Смірнов Іван	Mitsubishi Pajero Wagon	17.06.2009
Петров Юрій	Volkswagen Caddy	21.07.2010
Олександр Олександр	Ford Mondeo	12.01.2010
Куцуруза Юрій	Toyota Camry	14.10.2006
Александр Дмитро	Mazda CX-7	11.06.2010
Підійом Геннадій	Mercedes-Benz C-Class	07.05.2010

Додаток K

ПІБ	Домашня адреса	Домашній телефон	Адреса електронної пошти
Андрійчук Василь			
Бабкіна Галина	в ул. Корольова 87	(5464)54 - -	
Белова Іванна		(6546)546-54-65	
Воронова Галина	в ул. Корольова 2	(6546)456-54-654	
Кравец Олександр		(6456)546 - -	
Кривульський Олександр	в ул. Кн.Острозького 8а		esec@mail.ru
Крилова Інна			
Кротов Олександр			
Мартинюк Михайло	в ул. Корольова 2	(2543)685-46-546	19misha91@mail.ru
Новіков Артем			

Додаток L

Прізвище, ім'я, по батькові	Дата народження	Вік
Андрійчук Василь	02.07.1963	46
Бабкіна Галина	30.08.1943	66
Белова Іванна	08.12.1968	41
Воронова Галина	19.09.1958	51
Кравец Олександр	29.05.1960	50
Кривульський Олександр	20.12.1991	18
Крилова Інна	09.01.1958	52
Кротов Олександр	04.03.1955	55
Мартинюк Михайло	15.08.1991	18
Новіков Артем	19.02.1952	58
Петренко Тарас	17.08.1951	58
Сенюк Микола	05.07.1992	17
Ценюк Стас	20.07.1992	17
Чех омський Андрій	10.12.1991	18
Яценев Алла	02.07.1969	40

Додаток М

Оклади співробітників

Прізвище, ім'я, по батькові	Посадовий оклад
Андрійчук Василь	140,00
Бабіка Галина	140,00
Белова Іванна	140,00
Воронова Галина	140,00
Кравець Олександр	140,00
Кравчукський Олександр	350,00
Кривоноз Ірина	140,00
Кривоноз Олександр	140,00
Мартинюк Михайло	780,00
Новіков Артем	140,00
Петренко Тарас	140,00
Сенюк Микола	555,00
Цепко Стас	1.5 000,00
Чесомський Андрій	530,00
Ясенева Алла	140,00

Страница: 1

Додаток N

Співробітники

Прізвище, ім'я, по батькові	Назва відділу	Стаж
Андрійчук Василь	Цех по фарбуванню	13
Бабіка Галина	Цех по фарбуванню	7
Белова Іванна	Цех по фарбуванню	12
Воронова Галина	Ремонтний цех	11
Кравець Олександр	Адміністрація	2
Кравчукський Олександр	Ремонтний цех	14
Кривоноз Ірина	Ремонтний цех	9
Кривоноз Олександр	Адміністрація	24
Мартинюк Михайло	Ремонтний цех	11
Новіков Артем	Ремонтний цех	6
Петренко Тарас	Ремонтний цех	5
Сенюк Микола	Адміністрація	12
Цепко Стас	Бухгалтерія	13
Чесомський Андрій	Бухгалтерія	8
Ясенева Алла	Ремонтний цех	34

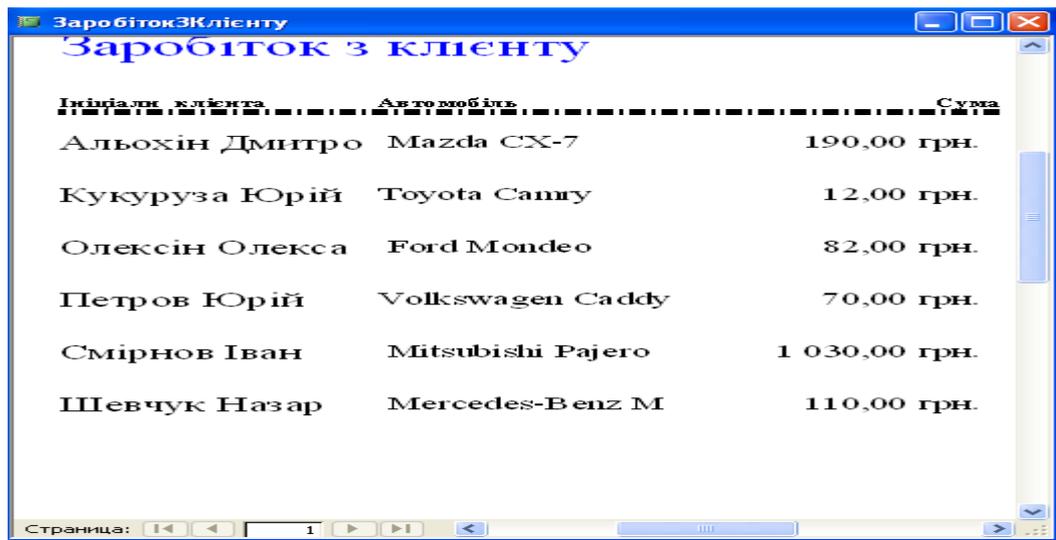
Страница: 1

Додаток O

АвтоЗаРокомВипуску

ПІБ клієнта	Автомобіль	Рік випуску
Альохін Дмитро	Mazda CX-7	2007
Гарячий Олександр	Mercedes-Benz C1550	2007
Кукурюз а Юрій	Toyota Camry	2007
Смірнов Іван	Mitsubishi Pajero Wagon	2007

Страница: 1



The screenshot shows a software window with a blue title bar containing the text 'Заробіток з клієнту'. The main content area displays a table with three columns: 'Ініціали клієнта', 'Автомобіль', and 'Сума'. The table lists six entries with their respective client names, car models, and earnings in Ukrainian hryvnia (грн.). At the bottom of the window, there is a status bar with the text 'Страница: 1' and navigation icons.

Ініціали клієнта	Автомобіль	Сума
Альохін Дмитро	Mazda CX-7	190,00 грн.
Кукуруза Юрій	Toyota Camry	12,00 грн.
Олексін Олекса	Ford Mondeo	82,00 грн.
Петров Юрій	Volkswagen Caddy	70,00 грн.
Смірнов Іван	Mitsubishi Pajero	1 030,00 грн.
Шевчук Назар	Mercedes-Benz M	110,00 грн.

```

Option Compare Database
Private Sub Кнопка5_Click()
On Error GoTo Err_Кнопка5_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = ChrW(1042) & ChrW(1110) & ChrW(1076) & ChrW(1076) & ChrW(1110) &
ChrW(1083) & ChrW(1080)
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Кнопка5_Click:
    Exit Sub
Err_Кнопка5_Click:
    MsgBox Err.Description
    Resume Exit_Кнопка5_Click
End Sub
Private Sub Кнопка6_Click()
On Error GoTo Err_Кнопка6_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = ChrW(1057) & ChrW(1087) & ChrW(1110) & ChrW(1074) & ChrW(1088) &
ChrW(1086) & ChrW(1073) & ChrW(1110) & ChrW(1090) & ChrW(1085) & ChrW(1080) &
ChrW(1082) & ChrW(1080)
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Кнопка6_Click:
    Exit Sub
Err_Кнопка6_Click:
    MsgBox Err.Description
    Resume Exit_Кнопка6_Click
End Sub
Private Sub Кнопка7_Click()
On Error GoTo Err_Кнопка7_Click
    Screen.PreviousControl.SetFocus
    DoCmd.DoMenuItem acFormBar, acEditMenu, 10, , acMenuVer70
Exit_Кнопка7_Click:
    Exit Sub
Err_Кнопка7_Click:
    MsgBox Err.Description
    Resume Exit_Кнопка7_Click
End Sub
Private Sub Кнопка8_Click()
On Error GoTo Err_Кнопка8_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = ChrW(1042) & ChrW(1110) & ChrW(1076) & ChrW(1076) & ChrW(1110) &
ChrW(1083) & ChrW(1080)
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Кнопка8_Click:
    Exit Sub
Err_Кнопка8_Click:
    MsgBox Err.Description

```

```

    Resume Exit_Кнопка8_Click
End Sub
Private Sub Кнопка9_Click()
On Error GoTo Err_Кнопка9_Click
    DoCmd.Close
Exit_Кнопка9_Click:
    Exit Sub
Err_Кнопка9_Click:
    MsgBox Err.Description
    Resume Exit_Кнопка9_Click
End Sub
Private Sub Кнопка10_Click()
On Error GoTo Err_Кнопка10_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = ChrW(1054) & ChrW(1092) & ChrW(1086) & ChrW(1088) & ChrW(1084) &
ChrW(1083) & ChrW(1077) & ChrW(1085) & ChrW(1085) & ChrW(1103) & ChrW(32) &
ChrW(1079) & ChrW(1072) & ChrW(1084) & ChrW(1086) & ChrW(1074) & ChrW(1083) &
ChrW(1077) & ChrW(1085) & ChrW(1085) & ChrW(1103)
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Кнопка10_Click:
    Exit Sub
Err_Кнопка10_Click:
    MsgBox Err.Description
    Resume Exit_Кнопка10_Click
End Sub
Private Sub Кнопка11_Click()
On Error GoTo Err_Кнопка11_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = ChrW(1054) & ChrW(1073) & ChrW(1083) & ChrW(1110) & ChrW(1082) &
ChrW(1055) & ChrW(1088) & ChrW(1086) & ChrW(1074) & ChrW(1077) & ChrW(1076) &
ChrW(1077) & ChrW(1085) & ChrW(1086) & ChrW(1111) & ChrW(1056) & ChrW(1086) &
ChrW(1073) & ChrW(1086) & ChrW(1090) & ChrW(1080)
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Кнопка11_Click:
    Exit Sub
Err_Кнопка11_Click:
    MsgBox Err.Description
    Resume Exit_Кнопка11_Click
End Sub
Private Sub Кнопка13_Click()
On Error GoTo Err_Кнопка13_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = ChrW(1042) & ChrW(1110) & ChrW(1076) & ChrW(1076) & ChrW(1110) &
ChrW(1083) & ChrW(1080)
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Кнопка13_Click:
    Exit Sub
Err_Кнопка13_Click:
    MsgBox Err.Description

```

```

    Resume Exit_Кнопка13_Click
End Sub
Private Sub Кнопка14_Click()
On Error GoTo Err_Кнопка14_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = ChrW(1050) & ChrW(1083) & ChrW(1110) & ChrW(1108) & ChrW(1085) &
ChrW(1090) & ChrW(1080)
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Кнопка14_Click:
    Exit Sub
Err_Кнопка14_Click:
    MsgBox Err.Description
    Resume Exit_Кнопка14_Click
End Sub
Private Sub Кнопка15_Click()
On Error GoTo Err_Кнопка15_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = ChrW(1055) & ChrW(1086) & ChrW(1089) & ChrW(1083) & ChrW(1091) &
ChrW(1075) & ChrW(1080)
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Кнопка15_Click:
    Exit Sub
Err_Кнопка15_Click:
    MsgBox Err.Description
    Resume Exit_Кнопка15_Click
End Sub
Private Sub Кнопка16_Click()
On Error GoTo Err_Кнопка16_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = ChrW(1057) & ChrW(1087) & ChrW(1110) & ChrW(1074) & ChrW(1088) &
ChrW(1086) & ChrW(1073) & ChrW(1110) & ChrW(1090) & ChrW(1085) & ChrW(1080) &
ChrW(1082) & ChrW(1080)
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Кнопка16_Click:
    Exit Sub
Err_Кнопка16_Click:
    MsgBox Err.Description
    Resume Exit_Кнопка16_Click
End Sub
Private Sub Кнопка17_Click()
On Error GoTo Err_Кнопка17_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = ChrW(1054) & ChrW(1073) & ChrW(1083) & ChrW(1110) & ChrW(1082) &
ChrW(1055) & ChrW(1088) & ChrW(1086) & ChrW(1074) & ChrW(1077) & ChrW(1076) &
ChrW(1077) & ChrW(1085) & ChrW(1086) & ChrW(1111) & ChrW(1056) & ChrW(1086) &
ChrW(1073) & ChrW(1086) & ChrW(1090) & ChrW(1080)
    DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Кнопка17_Click:

```

```

Exit Sub
Err_Кнопка17_Click:
MsgBox Err.Description
Resume Exit_Кнопка17_Click
End Sub
Private Sub Кнопка19_Click()
On Error GoTo Err_Кнопка19_Click
Dim stDocName As String
Dim stLinkCriteria As String
stDocName = ChrW(1054) & ChrW(1073) & ChrW(1083) & ChrW(1110) & ChrW(1082) &
ChrW(1055) & ChrW(1088) & ChrW(1086) & ChrW(1074) & ChrW(1077) & ChrW(1076) &
ChrW(1077) & ChrW(1085) & ChrW(1086) & ChrW(1111) & ChrW(1056) & ChrW(1086) &
ChrW(1073) & ChrW(1086) & ChrW(1090) & ChrW(1080)
DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit_Кнопка19_Click:
Exit Sub
Err_Кнопка19_Click:
MsgBox Err.Description
Resume Exit_Кнопка19_Click
End Sub
Private Sub Кнопка23_Click()
On Error GoTo Err_Кнопка23_Click
Dim stDocName As String
stDocName = ChrW(1046) & ChrW(1110) & ChrW(1085) & ChrW(1082) & ChrW(1080) &
ChrW(1042) & ChrW(1110) & ChrW(1076) & ChrW(1076) & ChrW(1110) & ChrW(1083) &
ChrW(1110) & ChrW(1074)
DoCmd.OpenQuery stDocName, acNormal, acEdit
Exit_Кнопка23_Click:
Exit Sub
Err_Кнопка23_Click:
MsgBox Err.Description
Resume Exit_Кнопка23_Click
End Sub
Private Sub Кнопка24_Click()
On Error GoTo Err_Кнопка24_Click
Dim stDocName As String
stDocName = ChrW(1057) & ChrW(1090) & ChrW(1072) & ChrW(1078) & ChrW(1057) &
ChrW(1087) & ChrW(1110) & ChrW(1074) & ChrW(1088) & ChrW(1086) & ChrW(1073) &
ChrW(1110) & ChrW(1090) & ChrW(1085) & ChrW(1080) & ChrW(1082) & ChrW(1110) &
ChrW(1074)
DoCmd.OpenQuery stDocName, acNormal, acEdit
Exit_Кнопка24_Click:
Exit Sub
Err_Кнопка24_Click:
MsgBox Err.Description
Resume Exit_Кнопка24_Click
End Sub
Private Sub Кнопка25_Click()
On Error GoTo Err_Кнопка25_Click
Dim stDocName As String
stDocName = ChrW(1040) & ChrW(1074) & ChrW(1090) & ChrW(1086) & ChrW(1047) &
ChrW(1072) & ChrW(1056) & ChrW(1086) & ChrW(1082) & ChrW(1086) & ChrW(1084) &

```

```

ChrW(1042) & ChrW(1080) & ChrW(1087) & ChrW(1091) & ChrW(1089) & ChrW(1082) &
ChrW(1091)
    DoCmd.OpenQuery stDocName, acNormal, acEdit
Exit_Кнопка25_Click:
    Exit Sub
Err_Кнопка25_Click:
    MsgBox Err.Description
    Resume Exit_Кнопка25_Click
End Sub
Private Sub Кнопка27_Click()
On Error GoTo Err_Кнопка27_Click
    Dim stDocName As String
    stDocName = ChrW(1040) & ChrW(1083) & ChrW(1092) & ChrW(1072) & ChrW(1074) &
ChrW(1110) & ChrW(1090) & ChrW(1085) & ChrW(1080) & ChrW(1081) & ChrW(1057) &
ChrW(1087) & ChrW(1080) & ChrW(1089) & ChrW(1086) & ChrW(1082) & ChrW(1042) &
ChrW(1110) & ChrW(1076) & ChrW(1076) & ChrW(1110) & ChrW(1083) & ChrW(1110) &
ChrW(1074)
    DoCmd.OpenReport stDocName, acPreview
Exit_Кнопка27_Click:
    Exit Sub
Err_Кнопка27_Click:
    MsgBox Err.Description
    Resume Exit_Кнопка27_Click
End Sub
Private Sub Кнопка28_Click()
On Error GoTo Err_Кнопка28_Click
    Dim stDocName As String
    stDocName = ChrW(1040) & ChrW(1083) & ChrW(1092) & ChrW(1072) & ChrW(1074) &
ChrW(1110) & ChrW(1090) & ChrW(1085) & ChrW(1080) & ChrW(1081) & ChrW(1057) &
ChrW(1087) & ChrW(1080) & ChrW(1089) & ChrW(1086) & ChrW(1082) & ChrW(1057) &
ChrW(1087) & ChrW(1074) & ChrW(1110) & ChrW(1088) & ChrW(1086) & ChrW(1073) &
ChrW(1110) & ChrW(1090) & ChrW(1085) & ChrW(1080) & ChrW(1082) & ChrW(1110) &
ChrW(1074)
    DoCmd.OpenReport stDocName, acPreview
Exit_Кнопка28_Click:
    Exit Sub
Err_Кнопка28_Click:
    MsgBox Err.Description
    Resume Exit_Кнопка28_Click
End Sub
Private Sub Кнопка29_Click()
On Error GoTo Err_Кнопка29_Click
    Dim stDocName As String
    stDocName = ChrW(1047) & ChrW(1072) & ChrW(1084) & ChrW(1086) & ChrW(1074) &
ChrW(1083) & ChrW(1077) & ChrW(1085) & ChrW(1085) & ChrW(1103)
    DoCmd.OpenReport stDocName, acPreview
Exit_Кнопка29_Click:
    Exit Sub
Err_Кнопка29_Click:
    MsgBox Err.Description
    Resume Exit_Кнопка29_Click
End Sub

```

```

Private Sub Кнопка30_Click()
On Error GoTo Err_Кнопка30_Click
Dim stDocName As String
stDocName = ChrW(1047) & ChrW(1072) & ChrW(1089) & ChrW(1086) & ChrW(1073) &
ChrW(1080) & ChrW(1047) & ChrW(1074) & ChrW(1103) & ChrW(39) & ChrW(1079) &
ChrW(1082) & ChrW(1091)
DoCmd.OpenReport stDocName, acPreview
Exit_Кнопка30_Click:
Exit Sub
Err_Кнопка30_Click:
MsgBox Err.Description
Resume Exit_Кнопка30_Click
End Sub
Private Sub Кнопка31_Click()
On Error GoTo Err_Кнопка31_Click
Dim stDocName As String
stDocName = ChrW(1054) & ChrW(1082) & ChrW(1083) & ChrW(1072) & ChrW(1076) &
ChrW(1080) & ChrW(1057) & ChrW(1087) & ChrW(1110) & ChrW(1074) & ChrW(1088) &
ChrW(1086) & ChrW(1073) & ChrW(1110) & ChrW(1090) & ChrW(1085) & ChrW(1080) &
ChrW(1082) & ChrW(1110) & ChrW(1074)
DoCmd.OpenReport stDocName, acPreview
Exit_Кнопка31_Click:
Exit Sub
Err_Кнопка31_Click:
MsgBox Err.Description
Resume Exit_Кнопка31_Click
End Sub
Private Sub Кнопка32_Click()
On Error GoTo Err_Кнопка32_Click
Dim stDocName As String
stDocName = ChrW(1055) & ChrW(1086) & ChrW(1089) & ChrW(1083) & ChrW(1091) &
ChrW(1075) & ChrW(1080)
DoCmd.OpenReport stDocName, acPreview
Exit_Кнопка32_Click:
Exit Sub
Err_Кнопка32_Click:
MsgBox Err.Description
Resume Exit_Кнопка32_Click
End Sub
Private Sub Кнопка33_Click()
On Error GoTo Err_Кнопка33_Click
Dim stDocName As String
stDocName = ChrW(1055) & ChrW(1110) & ChrW(1076) & ChrW(1088) & ChrW(1072) &
ChrW(1093) & ChrW(1091) & ChrW(1085) & ChrW(1086) & ChrW(1082) & ChrW(1057) &
ChrW(1091) & ChrW(1084)
DoCmd.OpenQuery stDocName, acNormal, acEdit
Exit_Кнопка33_Click:
Exit Sub
Err_Кнопка33_Click:
MsgBox Err.Description
Resume Exit_Кнопка33_Click
End Sub

```

```

Private Sub Кнопка34_Click()
On Error GoTo Err_Кнопка34_Click
Dim stDocName As String
stDocName = ChrW(1047) & ChrW(1072) & ChrW(1088) & ChrW(1086) & ChrW(1073) &
ChrW(1110) & ChrW(1090) & ChrW(1086) & ChrW(1082) & ChrW(1047) & ChrW(1050) &
ChrW(1083) & ChrW(1110) & ChrW(1108) & ChrW(1085) & ChrW(1090) & ChrW(1091)
DoCmd.OpenQuery stDocName, acNormal, acEdit
Exit_Кнопка34_Click:
Exit Sub
Err_Кнопка34_Click:
MsgBox Err.Description
Resume Exit_Кнопка34_Click
End Sub
Private Sub Кнопка35_Click()
On Error GoTo Err_Кнопка35_Click
Dim stDocName As String
stDocName = ChrW(1057) & ChrW(1090) & ChrW(1074) & ChrW(1086) & ChrW(1088) &
ChrW(1077) & ChrW(1085) & ChrW(1085) & ChrW(1103) & ChrW(1055) & ChrW(1083) &
ChrW(1072) & ChrW(1090) & ChrW(1042) & ChrW(1110) & ChrW(1076) & ChrW(1086) &
ChrW(1084) & ChrW(1086) & ChrW(1089) & ChrW(1090) & ChrW(1110) & ChrW(1042) &
ChrW(1110) & ChrW(1076) & ChrW(1076) & ChrW(1110) & ChrW(1083) & ChrW(1091)
DoCmd.OpenQuery stDocName, acNormal, acEdit
Exit_Кнопка35_Click:
Exit Sub
Err_Кнопка35_Click:
MsgBox Err.Description
Resume Exit_Кнопка35_Click
End Sub
Private Sub Кнопка36_Click()
On Error GoTo Err_Кнопка36_Click
Dim stDocName As String
stDocName = ChrW(1057) & ChrW(1090) & ChrW(1074) & ChrW(1086) & ChrW(1088) &
ChrW(1077) & ChrW(1085) & ChrW(1085) & ChrW(1103) & ChrW(1055) & ChrW(1083) &
ChrW(1110) & ChrW(1090) & ChrW(1072) & ChrW(1078) & ChrW(1085) & ChrW(1086) &
ChrW(1111) & ChrW(1042) & ChrW(1110) & ChrW(1076) & ChrW(1086) & ChrW(1084) &
ChrW(1086) & ChrW(1089) & ChrW(1090) & ChrW(1110) & ChrW(1047) & ChrW(1072) &
ChrW(1057) & ChrW(1090) & ChrW(1072) & ChrW(1078)
DoCmd.OpenQuery stDocName, acNormal, acEdit
Exit_Кнопка36_Click:
Exit Sub
Err_кнопка 36_Click:
MsgBox Err.Description
Resume Exit_кнопка 36_Click
End Sub
Private Sub кнопка 37_Click()
On Error GoTo Err_кнопка 37_Click
Dim stDocName As String
stDocName = ChrW(1055) & ChrW(1110) & ChrW(1076) & ChrW(1074) & ChrW(1080) &
ChrW(1097) & ChrW(1077) & ChrW(1085) & ChrW(1085) & ChrW(1103) & ChrW(1054) &
ChrW(1082) & ChrW(1083) & ChrW(1072) & ChrW(1076) & ChrW(1110) & ChrW(1074) &
ChrW(1053) & ChrW(1072) & ChrW(49) & ChrW(48) & ChrW(1042) & ChrW(1110) &

```

```

ChrW(1076) & ChrW(1089) & ChrW(1090) & ChrW(1086) & ChrW(1090) & ChrW(1082) &
ChrW(1110) & ChrW(1074)
    DoCmd.OpenQuery stDocName, acNormal, acEdit
Exit_кнопка 37_Click:
    Exit Sub
Err_кнопка 37_Click:
    MsgBox Err.Description
    Resume Exit_кнопка 37_Click
End Sub
Private Sub кнопка 39_Click()
On Error GoTo Err_кнопка 39_Click
    Dim stDocName As String
    stDocName = ChrW(1050) & ChrW(1083) & ChrW(1110) & ChrW(1108) & ChrW(1085) &
ChrW(1090) & ChrW(1080) & ChrW(1047) & ChrW(1072) & ChrW(1055) & ChrW(1077) &
ChrW(1088) & ChrW(1110) & ChrW(1086) & ChrW(1076)
    DoCmd.OpenQuery stDocName, acNormal, acEdit
Exit_кнопка 39_Click:
    Exit Sub
Err_кнопка 39_Click:
    MsgBox Err.Description
    Resume Exit_кнопка 39_Click
End Sub
Private Sub кнопка 40_Click()
On Error GoTo Err_кнопка40_Click
    Dim stDocName As String
    stDocName = ChrW(1050) & ChrW(1110) & ChrW(1083) & ChrW(1100) & ChrW(1082) &
ChrW(1110) & ChrW(1089) & ChrW(1090) & ChrW(1100) & ChrW(1057) & ChrW(1087) &
ChrW(1110) & ChrW(1074) & ChrW(1088) & ChrW(1086) & ChrW(1073) & ChrW(1055) &
ChrW(1086) & ChrW(1042) & ChrW(1110) & ChrW(1076) & ChrW(1076) & ChrW(1110) &
ChrW(1083) & ChrW(1072) & ChrW(1084)
    DoCmd.OpenQuery stDocName, acNormal, acEdit
Exit_кнопка 40_Click:
    Exit Sub
Err_кнопка 40_Click:
    MsgBox Err.Description
    Resume Exit_кнопка 40_Click
End Sub
Private Sub кнопка 41_Click()
On Error GoTo Err_кнопка 41_Click
    Dim stDocName As String
    stDocName = ChrW(1057) & ChrW(1087) & ChrW(1110) & ChrW(1074) & ChrW(1088) &
ChrW(1086) & ChrW(1073) & ChrW(1110) & ChrW(1090) & ChrW(1085) & ChrW(1080) &
ChrW(1082) & ChrW(1110) & ChrW(1054) & ChrW(1073) & ChrW(1089) & ChrW(1083) &
ChrW(1091) & ChrW(1075) & ChrW(1086) & ChrW(1074) & ChrW(1091) & ChrW(1102) &
ChrW(1095) & ChrW(1110) & ChrW(1050) & ChrW(1083) & ChrW(1110) & ChrW(1108) &
ChrW(1085) & ChrW(1090) & ChrW(1110) & ChrW(1074)
    DoCmd.OpenQuery stDocName, acNormal, acEdit
Exit_кнопка 41_Click:
    Exit Sub
Err_кнопка 41_Click:
    MsgBox Err.Description
    Resume Exit_кнопка 41_Click

```

```

End Sub
Private Sub кнопка 42_Click()
On Error GoTo Err_кнопка 42_Click
Dim stDocName As String
stDocName = ChrW(1055) & ChrW(1088) & ChrW(1077) & ChrW(1084) & ChrW(1110) &
ChrW(1111)
DoCmd.OpenQuery stDocName, acNormal, acEdit
Exit_кнопка 42_Click:
Exit Sub
Err_кнопка 42_Click:
MsgBox Err.Description
Resume Exit_кнопка 42_Click
End Sub
Private Sub кнопка 44_Click()
On Error GoTo Err_кнопка 44_Click
Dim stDocName As String
stDocName = ChrW(1057) & ChrW(1090) & ChrW(1072) & ChrW(1078)
DoCmd.OpenReport stDocName, acPreview
Exit_кнопка 44_Click:
Exit Sub
Err_кнопка 44_Click:
MsgBox Err.Description
Resume Exit_кнопка 44_Click
End Sub
Private Sub кнопка 45_Click()
On Error GoTo Err_кнопка 45_Click
Dim stDocName As String
stDocName = ChrW(1040) & ChrW(1074) & ChrW(1090) & ChrW(1086) & ChrW(1047) &
ChrW(1072) & ChrW(1056) & ChrW(1086) & ChrW(1082) & ChrW(1086) & ChrW(1084) &
ChrW(1042) & ChrW(1080) & ChrW(1087) & ChrW(1091) & ChrW(1089) & ChrW(1082) &
ChrW(1091)
DoCmd.OpenReport stDocName, acPreview
Exit_кнопка 45_Click:
Exit Sub
Err_кнопка 45_Click:
MsgBox Err.Description
Resume Exit_кнопка 45_Click
End Sub
Private Sub кнопка 46_Click()
On Error GoTo Err_Кнопка46_Click
DoCmd.Quit
Exit_кнопка 46_Click:
Exit Sub
Err_кнопка 46_Click:
MsgBox Err.Description
Resume Exit_кнопка 46_Click
End Sub
Private Sub кнопка 47_Click()
On Error GoTo Err_Кнопка47_Click
Dim stDocName As String

```

```

    stDocName = ChrW(1047) & ChrW(1072) & ChrW(1088) & ChrW(1086) & ChrW(1073) &
ChrW(1110) & ChrW(1090) & ChrW(1086) & ChrW(1082) & ChrW(1047) & ChrW(1050) &
ChrW(1083) & ChrW(1110) & ChrW(1108) & ChrW(1085) & ChrW(1090) & ChrW(1091)
    DoCmd.OpenReport stDocName, acPreview
Exit_кнопка 47_Click:
    Exit Sub
Err_кнопка 47_Click:
    MsgBox Err.Description
    Resume Exit_кнопка 47_Click
End Sub
Private Sub кнопка 48_Click()
On Error GoTo Err_Кнопка48_Click
    Dim stDocName As String
    stDocName = ChrW(1050) & ChrW(1083) & ChrW(1110) & ChrW(1108) & ChrW(1085) &
ChrW(1090) & ChrW(1080) & ChrW(1047) & ChrW(1072) & ChrW(1055) & ChrW(1077) &
ChrW(1088) & ChrW(1110) & ChrW(1086) & ChrW(1076)
    DoCmd.OpenReport stDocName, acPreview
Exit_кнопка 48_Click:
    Exit Sub
Err_кнопка 48_Click:
    MsgBox Err.Description
    Resume Exit_кнопка 48_Click
End Sub
Private Sub кнопка 49_Click()
On Error GoTo Err_Кнопка49_Click
    Dim stDocName As String
    stDocName = ChrW(1040) & ChrW(1074) & ChrW(1090) & ChrW(1086) & ChrW(1047) &
ChrW(1072) & ChrW(1056) & ChrW(1086) & ChrW(1082) & ChrW(1086) & ChrW(1084) &
ChrW(1042) & ChrW(1080) & ChrW(1087) & ChrW(1091) & ChrW(1089) & ChrW(1082) &
ChrW(1091)
    DoCmd.OpenReport stDocName, acPreview
Exit_кнопка49_Click:
    Exit Sub
Err_Кнопка49_Click:
    MsgBox Err.Description
    Resume Exit_Кнопка49_Click
End Sub

```