

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних наук  
(повна назва кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка бази даних комп'ютерних комплектуючих на прикладі  
магазину «Hard Soft».

Виконав: студент IV курсу, групи СНС-41

спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

(підпис)

Курчак Ю.Р.

(прізвище та ініціали)

Керівник

(підпис)

Литвиненко Я.В.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Литвиненко Я.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Тотосько О.В.

(прізвище та ініціали)

Тернопіль  
2023

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних наук  
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.  
(підпис) (прізвище та ініціали)

« 21 » червня 2023 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр  
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки  
(шифр і назва спеціальності)

Студенту Курчак Юрій Романович  
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка бази даних комп'ютерних комплектуючих на прикладі магазину «Hard Soft».

Керівник роботи Литвиненко Ярослав Володимирович, д.т.н., про. професор кафедри КН  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 7 » лютого 2023 року № 4/7-133

2. Термін подання студентом завершеної роботи 22.06.2023р.

3. Вихідні дані до роботи Перелік необхідної інформації яка буде врахована в програмному забезпеченні

4. Зміст роботи (перелік питань, які потрібно розробити)

ВСТУП РОЗДІЛ 1. РОЗРОБКА БАЗИ ДАНИХ КОМП'ЮТЕРНИХ КОМПЛЕКТУЮЧИХ МАГАЗИНУ «HARD & SOFT». 1.1 Постановка та актуальність задачі. 1.2 Огляд технологій для проєктування БД. 1.3 Основні поняття бази даних. 1.4 Реляційна модель даних. 1.5 Види концептуальних моделей та розробка концептуальної моделі. 1.6 Проєктування бази даних програми. 1.7 Вибір мови програмування.

1.8. Висновок до першого розділу. РОЗДІЛ 2. РОЗРОБКА ІНТЕРФЕЙСУ ПРОГРАМИ ДЛЯ БАЗИ ДАНИХ. 2.1 Компоненти Delphi для роботи з базами даних. 2.2 Опис інтерфейсу програми та робота з програмою. 2.3 Висновок до другого розділу. РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ. 3.1 Зміст та завдання БЖД. 3.2 Оцінка ризиків БЖД у життєдіяльності людини. 3.3 Соціальні небезпеки для людини. 3.4. Висновок до третього розділу. Висновки. Список джерел. Додатки.

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

Слайди: Тема 1. Мета та задачі 2. Структурна схема бази даних 3. Вигляд розробленого інтерфейсу 4. Вигляд розробленого інтерфейсу 5. Вигляд розробленого інтерфейсу 6. Вигляд розробленого інтерфейсу 7. Блок схема алгоритму пошуку інформації по замовленнях 8. Висновки 9.



## АНОТАЦІЯ

Розробка бази даних комп'ютерних комплектуючих на прикладі магазину «Hard Soft»// Кваліфікаційна робота освітнього рівня «Бакалавр» // Курчак Юрій Романович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНс-41 // Тернопіль, 2023 // С. , рис. – , табл. – , кресл. – , додат. – , бібліогр. – .

**Ключові слова:** Програмне забезпечення, база даних, звіт, замовлення.

Кваліфікаційна робота присвячена розробці бази даних комп'ютерних комплектуючих для ведення замовлень для магазину «Hard & Soft», програмне забезпечення реалізує зберігання інформації в базі даних та її перегляд, формування різних звітів. У роботі приведено основні алгоритми, використані при розробці програми та інструкцію з використання програмного забезпечення.

У першому розділі роботи проведено огляд технологій та розроблена структура бази даних комп'ютерних комплектуючих для ведення замовлень для магазину «Hard & Soft».

У другому розділі роботи описана практична реалізація бази даних комп'ютерних комплектуючих для ведення замовлень для магазину «Hard & Soft», наведена розробка основних елементів інтерфейсу.

В третьому розділі кваліфікаційної роботи описано основні питання, які стосуються змісту та завдань БЖД. Також розписані питання оцінка ризиків БЖД у життєдіяльності людини та основні соціальні небезпеки для людини.

## ANNOTATION

Development of a database of computer components on the example of the store "Hard Soft" // Qualification work of the educational level "Bachelor" // Kurchak Yuriy Romanovych // Ivan Pulyuy Ternopil National Technical University, Faculty of Computer Information Systems and Software Engineering, Faculty . of computer sciences, SNs-41 group // Ternopil, 2023 // C. , fig. - , tab. - , chair. - , add. – , bibliography –

**Keywords:** Software, database, report, order.

The qualification work is devoted to the development of a database of computer components for keeping orders for the "Hard & Soft" store, the software implements the storage of information in the database and its viewing, the generation of various reports. The work presents the main algorithms used in the development of the program and instructions for using the software.

In the first part of the work, an overview of technologies was carried out and the structure of the database of computer components for keeping orders for the "Hard & Soft" store was developed.

The second part of the work describes the practical implementation of the database of computer components for keeping orders for the "Hard & Soft" store, and the development of the main interface elements.

The third section of the qualification paper describes the main issues related to the content and tasks of BZD. There are also issues of assessment of the risks of BZD in human life and the main social dangers for humans.

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

БЖД – Безпека життєдіяльності.

ОС – Операційної системи.

ПЕОМ – Персональна електронна обчислювальна машина.

СУБД – Система управління базами даних.

## ЗМІСТ

ВСТУП .....	6
РОЗДІЛ 1. РОЗРОБКА БАЗИ ДАНИХ КОМП'ЮТЕРНИХ КОМПЛЕКТУЮЧИХ МАГАЗИНУ «HARD & SOFT» .....	8
1.1 Постановка та актуальність задачі.....	8
1.2 Огляд технологій для проєктування БД.....	9
1.3 Основні поняття бази даних .....	11
1.4 Реляційна модель даних.....	12
1.5 Види концептуальних моделей та розробка концептуальної моделі.	13
1.6 Проєктування бази даних програми .....	15
1.7 Вибір мови програмування.....	18
1.8. Висновок до першого розділу .....	20
РОЗДІЛ 2. РОЗРОБКА ІНТЕРФЕЙСУ ПРОГРМИ ДЛЯ БАЗИ ДАНИХ.....	21
2.1 Компоненти Delphi для роботи з базами даних .....	21
2.2 Опис інтерфейсу програми та робота з програмою.....	22
2.3 Висновок до другого розділу .....	27
РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	28
3.1 Зміст та завдання БЖД.....	28
3.2 Оцінка ризиків БЖД у життєдіяльності людини .....	31
3.3 Соціальні небезпеки для людини.....	34
3.4. Висновок до третього розділу .....	36
ВИСНОВКИ.....	37
ПЕРЕЛІК ДЖЕРЕЛ .....	38
ДОДАТКИ	

## ВСТУП

**Актуальність теми.** В наш час існує багато програм які зберігають значну інформацію про різні продукти товари та інше. Цими програмами є базами даних. Основне призначення бази даних полягає в накопиченні важливої інформації для підприємства, фірми чи установи для надійного зберігання. І одночасно для швидкого доступу чи пошуку необхідної інформації в ній.

Область використання баз даних на даний час дуже широка. Вони використовуються в банках, організаціях, підприємствах, що ведуть контроль всіх коштів, а також є бази даних які ведуть облік наприклад працівників невеликих фірм. Бази даних в нашій країні набули надзвичайно широкого застосування. Малі підприємства дуже часто користуються простими базами даних без клієнт-серверної частини, наприклад, для контролю за фінансами в офісі. Підприємства, що використовують розгалужену мережу офісів вже побачили перспективу в використанні клієнт-серверних підходах контролю над базами даних.

Використання саме електронних баз даних доступ до яких здійснюється за допомогою комп'ютера є дуже важливим на сьогоднішній день тому, що використання «паперових» баз даних, які ще існують на багатьох підприємствах не є зручним. Так наприклад, пошук потрібної інформації в такій базі даних займає значну кількість часу, тому актуальним є створення електронних баз даних.

**Мета і задачі дослідження.** Метою даної кваліфікаційної роботи освітнього рівня «Бакалавр» було створення бази даних, яка дозволяє проводити облік замовлень в магазині «Hard & Soft». Для досягнення поставленої мети потрібно виконати ряд завдань, зокрема:

- Розробити структуру бази даних;
- Розробити інтерфейс програми;
- Провести тестування програми.



### **Практичне значення одержаних результатів.**

Для ведення обліку товарів та замовлень в магазині «Hard & Soft» було створено базу даних, яка дозволяє автоматизувати ці процеси, що є важливим для практики.

## **РОЗДІЛ 1. РОЗРОБКА БАЗИ ДАНИХ КОМП'ЮТЕРНИХ КОМПЛЕКТУЮЧИХ МАГАЗИНУ «HARD & SOFT»**

### **1.1 Постановка та актуальність задачі**

Задача забезпечення обліку товарів на будь-якій фірмі є актуальною і потребує затрат часу та праці, а створення програмного забезпечення суттєво дозволить скоротити затрати праці та час на ці кроки. Крім цього полегшить і спростить роботу працівника, який її виконує.

Постановка задачі розробки бази даних виникає у багатьох сценаріях, коли потрібно ефективно та структуровано зберігати та управляти великим обсягом даних. Наприклад, найпоширеніші ситуації, де задача розробки бази даних є актуальною:

**Розробка веб-додатків:** Веб-додатки часто використовують бази даних для зберігання та обробки користувальницької інформації, наприклад, даних про користувачів, замовлень, платежів тощо. Розробка ефективної бази даних допомагає забезпечити надійну та швидку роботу веб-додатку.

**Управління проектами:** Великі проекти, будь то інженерні, інформаційні чи інші, можуть потребувати складних систем управління даними. База даних дозволяє зберігати дані про ресурси, задачі, зв'язки між ними, прогрес виконання проектів та іншу важливу інформацію.

**Електронна комерція:** В інтернет-магазинах та електронних платформах бази даних використовуються для зберігання товарів, інформації про замовлення, клієнтів, оплату тощо. Розробка добре організованої бази даних допомагає в управлінні товарами, запасами, виконанням замовлень та аналітиці діяльності компанії.

**Системи управління відносинами з клієнтами (CRM):** Бази даних допомагають зберігати дані про клієнтів, їх контактну інформацію, історію взаємодії, обслуговування тощо. Це дає змогу підтримувати ефективні системи CRM та забезпечувати персоналізоване обслуговування клієнтів.

Аналітика та прийняття рішень: Бази даних забезпечують зберігання великого обсягу даних, які можуть бути використані для аналітики та прийняття рішень. Вони дозволяють здійснювати запити, агрегувати дані, виконувати аналіз та знаходити інсайти, що підтримують прийняття стратегічних рішень.

Задача розробки бази даних залишається актуальною, оскільки висока ефективність та організованість управління даними є ключовими чинниками успіху для багатьох організацій та проектів. Вона дозволяє покращити доступність, швидкість, безпеку та аналітику даних, спрощує роботу з інформацією та допомагає забезпечити ефективну діяльність у різних галузях.

## **1.2 Огляд технологій для проектування БД**

Для створення БД можуть бути застосовані різні інструменти. Наприклад, ERwin є популярним програмним продуктом, який використовується для проектування баз даних і моделювання схем даних. На сьогоднішній день ERwin пропонує два основних програмних рішення: ERwin Data Modeler та ERwin Data Intelligence.

ERwin Data Modeler: ERwin Data Modeler - це програмний продукт, що дозволяє проектувати, моделювати та управляти схемами баз даних. Він має потужні інструменти для створення концептуальних, логічних та фізичних моделей даних, а також можливості для генерації скриптів створення бази даних для різних СУБД.

ERwin Data Intelligence: ERwin Data Intelligence - це програмний продукт, спрямований на управління метаданими та розуміння даних в організації. Він надає можливості для збору, управління та використання метаданих, а також для виявлення зв'язків та розуміння структури даних.

Обидва ці продукти мають свої унікальні функції та можливості, і кращий вибір залежить від конкретних потреб організації і проектів. Тому перед

вибором рекомендується ретельно оцінити вимоги, функціональність та можливість програмного продукту перед прийняттям рішення..

ERwin, як і багато інших сучасних інструментів для моделювання баз даних, має свої переваги та можливості. Наведемо декілька загальних переваг ERwin та сучасних інструментів для моделювання баз даних:

Професійна розробка моделей даних: ERwin надає потужні інструменти для професійної розробки моделей даних, включаючи концептуальні, логічні та фізичні моделі. Вони допомагають управляти складністю баз даних і забезпечують ясність та структурованість у процесі проектування.

Візуалізація і редагування: ERwin та сучасні інструменти для моделювання баз даних надають візуальні засоби для створення та редагування моделей даних. Це дозволяє легко сприймати та зрозуміти структуру даних, а також здійснювати зміни та внесення коректив до моделей.

Автоматизована генерація скриптів: ERwin та інші інструменти можуть автоматизувати генерацію скриптів створення баз даних для різних СУБД. Це полегшує процес розгортання та реалізації моделей даних у фактичні бази даних.

Сумісність з різними СУБД: ERwin та сучасні інструменти підтримують різні СУБД, такі як Oracle, SQL Server, MySQL, PostgreSQL та інші. Це дає можливість працювати з різними типами баз даних і забезпечує гнучкість у виборі технологій.

Колаборація та спільна робота: Багато сучасних інструментів мають функціональність для колаборації та спільної роботи над моделями даних. Це дозволяє розробникам, архітекторам та іншим учасникам команди працювати разом, обмінюючись інформацією та змінами у реальному часі.

Відмітимо, що конкретні переваги можуть варіюватися в залежності від вибраного інструменту для моделювання баз даних. Необхідно ознайомитися з функціональністю та можливостями кількох інструментів, порівняти їх і врахувати специфічні потреби нашого проекту, щоб зробити правильний вибір.

### 1.3 Основні поняття бази даних

База даних (БД) - це структурована колекція даних, яка зберігається та управляється з використанням спеціального програмного забезпечення, відомого як система управління базами даних (СУБД). База даних використовується для зберігання і організації великої кількості даних, до яких можна звертатися та які можна обробляти за допомогою запитів і операцій.

Основні поняття, пов'язані з базами даних:

**Таблиці:** Таблиці представляють схему даних і складаються з рядків і стовпців. Кожен стовпець таблиці відповідає певному типу даних, а кожен рядок містить конкретні значення для кожного стовпця.

**Схема бази даних:** Схема бази даних визначає структуру та організацію даних в базі даних. Вона включає таблиці, відношення між ними, обмеження цілісності та інші атрибути.

**Запити:** Запити використовуються для вибору, вставки, оновлення або видалення даних з бази даних. Запити дозволяють отримувати необхідну інформацію з бази даних за певними критеріями.

**Ключі:** Ключі використовуються для ідентифікації та зв'язку даних між різними таблицями. Головний ключ є унікальним ідентифікатором запису в таблиці, а зовнішній ключ встановлює зв'язок між двома таблицями.

**Транзакції:** Транзакції використовуються для забезпечення цілісності та безпеки даних в базі даних. Транзакція - це одинична операція або група операцій, які виконуються як єдине ціле.

**Резервне копіювання та відновлення:** Резервне копіювання бази даних важливо для забезпечення безпеки даних. Воно полягає в створенні резервних копій даних, щоб у разі втрати або пошкодження вихідних даних їх можна було відновити.

Існує багато різних СУБД, таких як MySQL, Oracle, SQL Server, PostgreSQL, MongoDB і багато інших, кожна з яких має свої унікальні особливості та відмінності у функціональності. Вибір СУБД залежить від

конкретних потреб проекту, масштабу системи, вимог до продуктивності та інших факторів.

#### **1.4 Реляційна модель даних**

Реляційна модель даних є однією з основних моделей даних, яка використовується в системах управління базами даних (СУБД). Вона була запропонована Едгаром Коддом в 1970 році і є популярною та широко використовуваною моделлю.

У реляційній моделі дані організовані у вигляді таблиць, відомих як "реляції". Кожна реляція складається з множини рядків, відомих як "кортежі", і множини стовпців, відомих як "атрибути". Кожен атрибут визначає тип даних, який може бути цілочисельним, рядком, датою, булевим значенням тощо.

Основні поняття реляційної моделі даних включають:

**Ключі:** Кожна реляція має один або декілька атрибутів, які можна використовувати як ключі. Головний ключ є унікальним ідентифікатором кожного кортежу у реляції. Зовнішній ключ встановлює зв'язок між різними реляціями.

**Відношення:** В реляційній моделі дані можуть мати відношення між собою. Наприклад, відношення "студенти" може мати відношення з реляцією "курси" через спільний атрибут "ID курсу".

**Цілісність даних:** Реляційна модель даних дозволяє встановлювати правила цілісності для забезпечення правильності та непорушення даних. Наприклад, обмеження унікальності може гарантувати, що значення в головному ключі є унікальними.

**Мови запитів:** Для роботи з реляційною моделлю даних використовуються мови запитів, такі як SQL (Structured Query Language). Ці мови дозволяють виконувати операції вибірки, вставки, оновлення та видалення даних з реляційних таблиць.

Переваги реляційної моделі даних включають:

**Структурованість:** Реляційна модель даних дозволяє структурувати дані в логічних таблицях з чітко визначеними відношеннями між ними.

**Простота:** Реляційна модель має просту та зрозумілу структуру, що спрощує процес розробки, запитів та адміністрування баз даних.

**Гнучкість:** Реляційна модель дозволяє ефективно маніпулювати та організовувати дані, виконувати різноманітні операції та аналітику.

**Незалежність від фізичної реалізації:** Запити до реляційних баз даних можуть бути виконані незалежно від того, як саме дані фізично зберігаються та організовані.

Однак, реляційна модель також має свої обмеження. Вона може бути неефективною для деяких видів даних, які мають складну структуру або потребують швидкого доступу до даних у великому масштабі. У таких випадках існують інші моделі даних, наприклад, ієрархічна, мережева або об'єктно-орієнтована модель, які можуть бути більш кращими.

## **1.5 Види концептуальних моделей та розробка концептуальної моделі**

Види концептуальних моделей даних включають:

**Ієрархічна модель:** Ця модель використовує ієрархічну структуру даних, де дані організовані у вигляді дерева з батьківськими та дочірніми вузлами. Кожен вузол може мати лише одного батька, але може мати багато дочірніх вузлів. Ієрархічна модель часто використовується в старіших системах управління базами даних.

**Мережева модель:** У мережевій моделі дані організовані у вигляді графа, де кожен вузол може мати багато батьківських і багато дочірніх вузлів. Ця модель дозволяє встановлювати складні зв'язки між об'єктами. Мережева модель була популярною до появи реляційної моделі, але зараз використовується значно менше.

**Реляційна модель:** Реляційна модель даних була описана раніше. Вона базується на ідеї реляцій, де дані організовані у вигляді таблиць зі стовпцями і

рядками. Реляційна модель є популярною та широко використовуваною моделлю у сучасних системах управління базами даних.

Об'єктно-орієнтована модель: Ця модель даних базується на концепції об'єктно-орієнтованого програмування, де дані моделюються як об'єкти з властивостями та методами. Об'єктно-орієнтована модель даних дозволяє більш гнучко описувати складні зв'язки та взаємодію об'єктів.

Розробка концептуальної моделі бази даних (БД) є важливим етапом у процесі проектування БД. Концептуальна модель описує високорівневу структуру та взаємозв'язки між даними, незалежно від конкретних деталей реалізації.

Процес розробки концептуальної моделі БД включає наступні етапи:

Визначення вимог: Спілкування зі зацікавленими сторонами для зрозуміння бізнес-вимог та потреб користувачів. Вимоги включають типи даних, взаємозв'язки між об'єктами, обмеження та функціональні вимоги.

Створення сутностей та атрибутів: Ідентифікування основних сутностей, які включають об'єкти, що потребують збереження даних. Для кожної сутності визначаються атрибути, які відображають властивості цих сутностей.

Визначення взаємозв'язків: Встановлення зв'язків між сутностями для відображення залежностей між ними. Взаємозв'язки можуть бути один-до-одного, один-до-багатьох або багато-до-багатьох.

Нормалізація: Перевірка концептуальної моделі на відповідність нормальним формам, що допомагають уникнути аномалій та забезпечують ефективну організацію даних.

Документування: Детальне документування концептуальної моделі, включаючи опис сутностей, атрибутів, взаємозв'язків та обмежень.

Розробка концептуальної моделі БД відбувається на етапі аналізу та проектування БД і є важливим кроком для успішної реалізації бази даних згідно з бізнес-потребами.



## 1.6 Проектування бази даних програми

Розроблена програма проста у використанні і не потребуватиме попереднього навчання, але ознайомлення з нею є обов'язковим. При написанні програми розроблено базу даних, яка складається з наступних таблиць:

- «таблиця банківські данні»,
- «таблиця моделі товару»,
- «таблиця категорій»,
- «таблиця списку складових товарів та послуг»,
- «таблиця специфікація товару»,
- «таблиця клієнтів»
- «таблиця список послуг»,
- «таблиця груп товарів»,
- «таблиця замовлень»,
- «таблиця констант операцій».

Для розробки таблиць бази даних використано такі форму документів такі як замовлення на певні товари.

В загальному форма документів які використовуються на практиці не нормалізована, оскільки виконання правил нормалізації призведе до зниження продуктивності даної розробки. Після нормалізації зросте кількість відношень, тому при виконанні запитів до такої бази даних потрібно багато об'єднань, що споживають ресурси системи. З ціллю підвищення продуктивності системи при переході на фізичний рівень здійснюється свідома відмова від нормальної форми.

Організація БД повинна задовільня ти таким основним властивостям:

Цілісність даних: БД повинна забезпечувати цілісність даних, що означає, що дані в ній мають бути коректні та сумісні з встановленими правилами цілісності. Це включає унікальність даних, обмеження цілісності для запобігання введенню некоректних даних та забезпечення внутрішньої консистентності даних.

**Надійність:** БД повинна бути надійною і забезпечувати захист від втрати даних внаслідок відмови апаратного чи програмного забезпечення. Це може бути досягнуто за допомогою резервного копіювання даних, реплікації, контролю цілісності та механізмів відновлення.

**Ефективність:** БД повинна бути ефективною в роботі з великими обсягами даних та виконанні запитів. Це може включати оптимізацію запитів, індексування даних, розподілення даних для розподіленої обробки та використання оптимальних алгоритмів обробки даних.

**Безпека:** БД повинна забезпечувати захист даних від несанкціонованого доступу, зміни або видалення. Це може бути досягнуто за допомогою автентифікації та авторизації користувачів, шифрування даних та механізмів аудиту дій користувачів.

**Розширюваність:** БД повинна бути гнучкою та здатною розширюватися для забезпечення обробки нових типів даних, зростання обсягів даних та забезпечення швидкого доступу до них. Це може включати механізми розділення даних, масштабування апаратного забезпечення та оптимізацію структури БД.

Ці властивості допомагають забезпечити ефективне та надійне функціонування БД, збереження цілісності та безпеки даних, а також забезпечити гнучкість та розширюваність для відповіді на змінні потреби користувачів та бізнесу.

Інтерфейс програмного забезпечення повинен бути орієнтований на кінцевого користувача і враховувати можливість того, що користувач не має необхідної бази знань по теорії баз даних.

Для організації зберігання даних отриманих з електролічильників створюємо локальну базу даних. Програми написані на Delphi здійснюють доступ до БД через BDE (Borland Database Engine процесор баз даних фірми Borland). Так, Borland Database Engine (BDE) був популярним програмним забезпеченням, розробленим компанією Borland, для доступу до реляційних баз даних з програм, написаних на платформі Delphi та C++Builder. BDE надавав



В таблиці ім'я поля повинно складатися з букв (допускається кирилиця) та цифр і починається з букви. В імені можна використовувати такі символи, як пробіл, “#”, “\$” та деякі інші. Не рекомендується використовувати символи “.”, “!” , “|”.

Для створення таблиці БД та додавання нових записів використовуємо розроблені процедури `NewBaza(n:integer;name:namstr;dir:string)` та

`AddBaza(n:integer;name:namstr;znach1:znach);`

Процедура `NewBaza(n:integer;name:namstr;dir:string)` створює таблицю БД. Кількість полів та їх назви задаються користувачем. Процедура має такі вхідні параметри:

`n` – кількість полів бази даних, які містять дані отримані з лічильників;

`name` – масив з назвами цих полів (`namstr=array [1..32] of string[10]`);

`dir` – каталог в якому розміщена таблиця.

Додавання записів в таблицю БД здійснюється процедурою `AddBaza(n:integer;name:namstr;znach1:znach)`. Вхідні параметри процедури:

Таблиці (див. рис.1.1) пов'язані між собою за допомогою полів.

Зв'язок між базами даних один-до-багатьох, тобто одному запису основної таблиці відповідає кілька записів підпорядкованої. В нашому випадку основною базою даних є «клієнти», а підпорядкованою є база даних «послуги» та «склад».

База даних «клієнти» пов'язана з базою даних «послуги» по полю прізвище, ім'я, назва фірми. При цьому між таблицями існує зв'язок один-до-одного, головною є таблиця «група товарів», а підпорядкованою «ціна товарів».

## 1.7 Вибір мови програмування

У нашому випадку мову програмування обрав замовник і цією мовою програмування є Delphi, програма і бази даних створювалися у середовищі програмування Delphi.

Delphi є об'єктно-орієнтованою мовою програмування, яка має свої переваги для розробки програмного забезпечення. Ось деякі переваги мови програмування Delphi:

**Легкість вивчення:** Delphi має простий та логічний синтаксис, що робить її легкою для вивчення, особливо для розробників з досвідом в інших мовах програмування, таких як C++ або Pascal.

**Широка підтримка платформи:** Delphi підтримує розробку програмного забезпечення для різних платформ, включаючи Windows, macOS, iOS, Android та Linux. Це дозволяє створювати кросплатформенні додатки з використанням єдиного коду.

**Радикальний збір сміття:** Delphi використовує радикальний збір сміття (garbage collection), що спрощує управління пам'яттю та дозволяє уникнути багатьох проблем, пов'язаних з утечками пам'яті.

**Багатофункціональність:** Delphi має багатий інструментарій для розробки, включаючи бібліотеки компонентів (VCL та FMX), які надають широкі можливості для створення графічного інтерфейсу користувача та обробки даних.

**Швидкодія:** Delphi є компільованою мовою програмування, що дозволяє створювати швидкодіючі додатки. Вона має добру оптимізацію та ефективну роботу з пам'яттю.

**Інтеграція з іншими мовами:** Delphi підтримує механізми взаємодії з іншими мовами програмування, такими як C++ і .NET. Це дозволяє використовувати функціонал і бібліотеки, написані на різних мовах.

**Спільнота та підтримка:** Delphi має активну спільноту розробників та підтримку від компанії Embarcadero. Це означає, що ви можете швидко знайти допомогу, ресурси та оновлення для розробки вашого проекту.

Загалом, Delphi є потужним інструментом для розробки додатків, який поєднує легкість вивчення, кросплатформенну підтримку, широкий функціонал та швидкодію.

## **1.8. Висновок до першого розділу**

В першому розділі кваліфікаційної роботи була здійснена постановка та актуальність задачі, проведений огляд технологій для проєктування БД, огляд основних понять БД, описано що таке реляційна модель та наведені види концептуальних моделей. Проведено проєктування бази даних та описані переваги мови програмування Delphi.

## РОЗДІЛ 2. РОЗРОБКА ІНТЕРФЕЙСУ ПРОГРАМИ ДЛЯ БАЗИ ДАНИХ

### 2.1 Компоненти Delphi для роботи з базами даних

Використовуючи Delphi, можна створити додатки, що працюють як з розрахованими на одного користувача базами даних, так і з серверними СУБД, такими як Oracle, Sybase, Informix, Interbase, MS SQL Server, DB2, а також з ODBC-джерелами.

Delphi має різноманітні компоненти та фреймворки для роботи з базами даних. Розглянемо декілька популярних компонентів Delphi для роботи з базами даних:

**TADOConnection і TADOTable:** Ці компоненти використовуються для роботи з базами даних за допомогою ADO (ActiveX Data Objects). Вони надають з'єднання з базою даних і можливість виконувати SQL-запити та отримувати результати.

**TFDConnection і TFDQuery:** Ці компоненти належать до FireDAC, потужного фреймворку для роботи з базами даних. Вони підтримують багато різних баз даних, включаючи MySQL, Oracle, SQL Server та багато інших.

**TSQLConnection і TSQLQuery:** Ці компоненти використовуються для роботи з базами даних через протокол ODBC (Open Database Connectivity). Вони забезпечують з'єднання з базою даних та можливість виконувати SQL-запити.

**TClientDataSet:** Цей компонент представляє локальний набір даних, який може працювати з різними джерелами даних, включаючи бази даних. Він надає функціональність кешування та маніпулювання даними.

**TIBDatabase і TIBQuery:** Ці компоненти використовуються для роботи з базами даних InterBase або Firebird. Вони надають з'єднання з базою даних та можливість виконувати SQL-запити.

**TDataSetProvider і TClientDataSet:** Ці компоненти використовуються для створення багаторівневих додатків з використанням технології DataSnap. Вони дозволяють передавати дані між клієнтською та серверною частинами додатка.

Це невелика кількість прикладів компонентів Delphi для роботи з базами даних. Delphi також підтримує використання ORM-фреймворків, таких як Entity Framework, для спрощення роботи з базами даних.

## 2.2 Опис інтерфейсу програми та робота з програмою

Перейдемо до опису розробленого інтерфейсу програми.

Після запуску виконавчого файлу з'являється вікно, що містить меню, як це показано на рисунку 2.1.

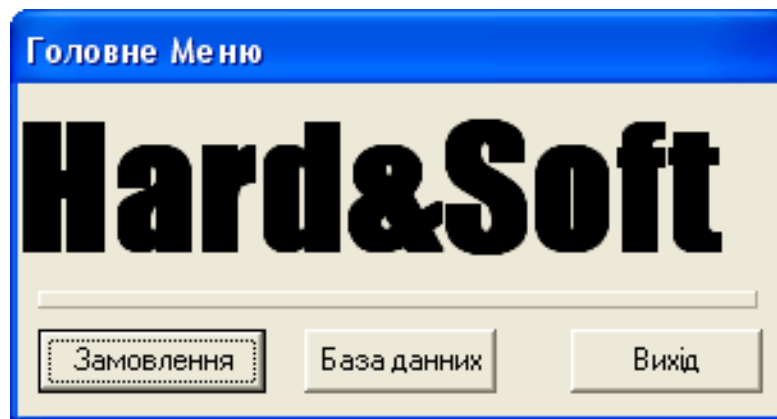


Рисунок 2.1 - Зображення інтерфейсу програми при запуску

Перед початком роботи такої програми необхідно заповнити базу даних. Так спочатку необхідно заповнити таблиці – вказавши товари що знаходяться в магазині. На наведеному прикладі подані такі види товарів як мікропроцесори. При цьому при виборі певної категорії вказується певний вид (характеристика) товару який продається, зазначається його ціна, наприклад, при виборі мікропроцесорів вибрано Core 2 Duo Box. Для добавлення в базу даних потрібної інформації необхідно вказівником мишки вибрати певну категорію товару після цього натиснути в порожньому полі таблиці і ввести необхідні данні, як це показано на рисунку 2.2.



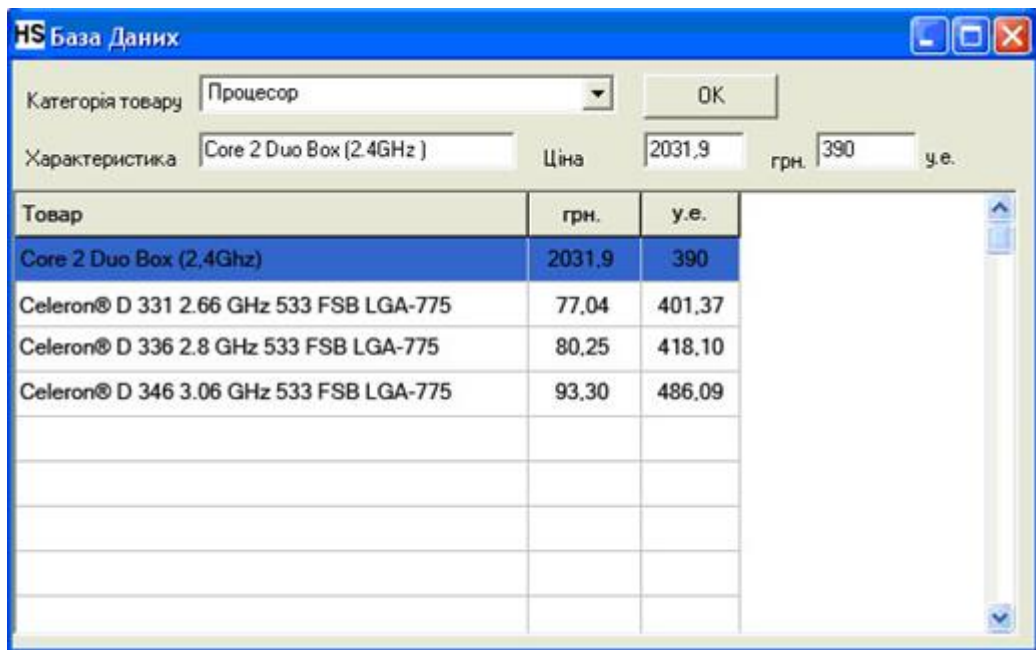


Рисунок 2.2 - Зображення вікна програми для введення нового товару

Після введення потрібної інформації необхідно натиснути „Ок” вся введена інформація запишеться в базу даних. Для збереження інформації в базі даних так само як і для введення її необхідно ввести пароль, як це показано на рисунку 2.3.

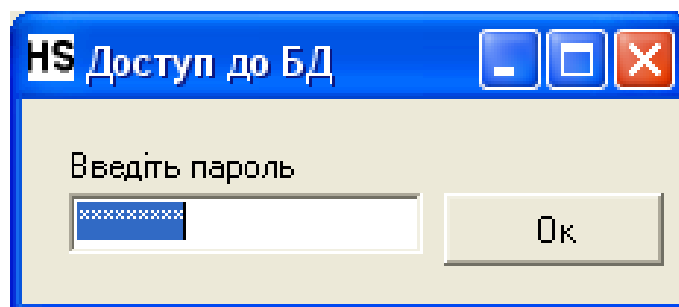


Рисунок 2.3 - Зображення інтерфейсу програми, що призначена для введення паролю

Після заповнення бази даних товарів можна розпочати роботу з програмою. Вибравши в головному меню «Замовлення». З’явиться вікно, як показано на рисунку 2.4.

Рисунок 2.4 – Зображення вікна програми для введення нового замовлення

При заповненні інформації про нове замовлення всі поля перевіряються на правильність вводу. Тут перевіряється щоб в полях де вводиться текст не було введено цифр і навпаки. Для проведення пошуку замовлень які були здійснені раніше і містяться в базі даних використовується кнопка «Пошук замовлень». При її активації відбувається пошук інформації згідно алгоритму поданого на рисунку 2.5

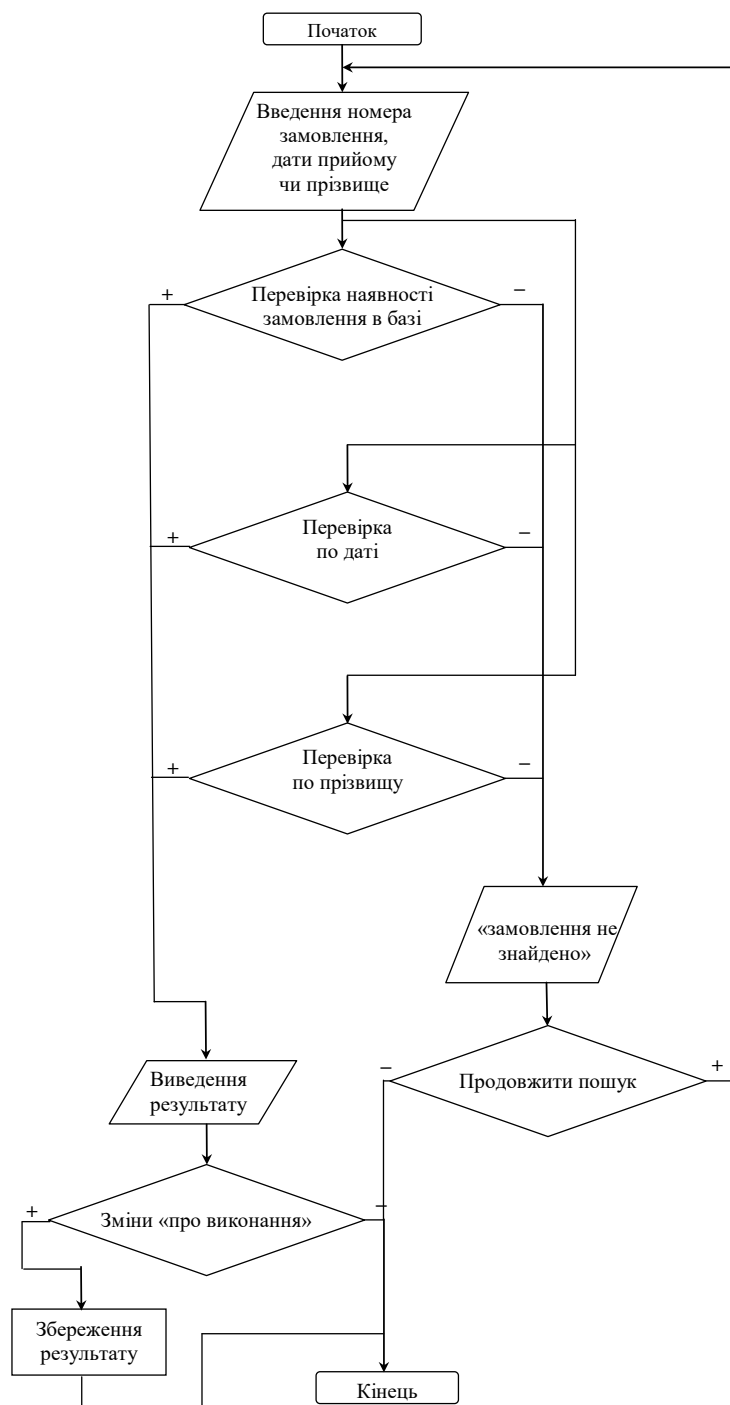


Рисунок 2.5 – Блок схема алгоритму пошуку інформації по замовленнях

Для проведення замовлення необхідно вибрати категорію товару і категорії вибрати конкретний товар, наприклад пам'ять. Якщо в типовій конфігурації комп'ютера не вистачає деякого товару. Наприклад, в категорії мультимедіа клієнт хоче замовити плату відео монтажу. В такому разі користувачу програми треба вибрати кнопку «Додатково» із додаткового меню обрати необхідний товар який буде додано в поточне замовлення. Для

завершення замовлення необхідно заповнити інформаційні поля про клієнта. Тут потрібно вказати прізвище, ім'я, телефон та орієнтовану дату виконання замовлення. Дата здійснення замовлення автоматично буде відображена в певному полі. Для відмітки в базі даних про виконання замовлення використовується елемент керування «замовлення виконано». Після здійснення замовлення інформація вноситься в базу даних та друкується на папері у вигляді замовлення, як показано на рисунку 2.6

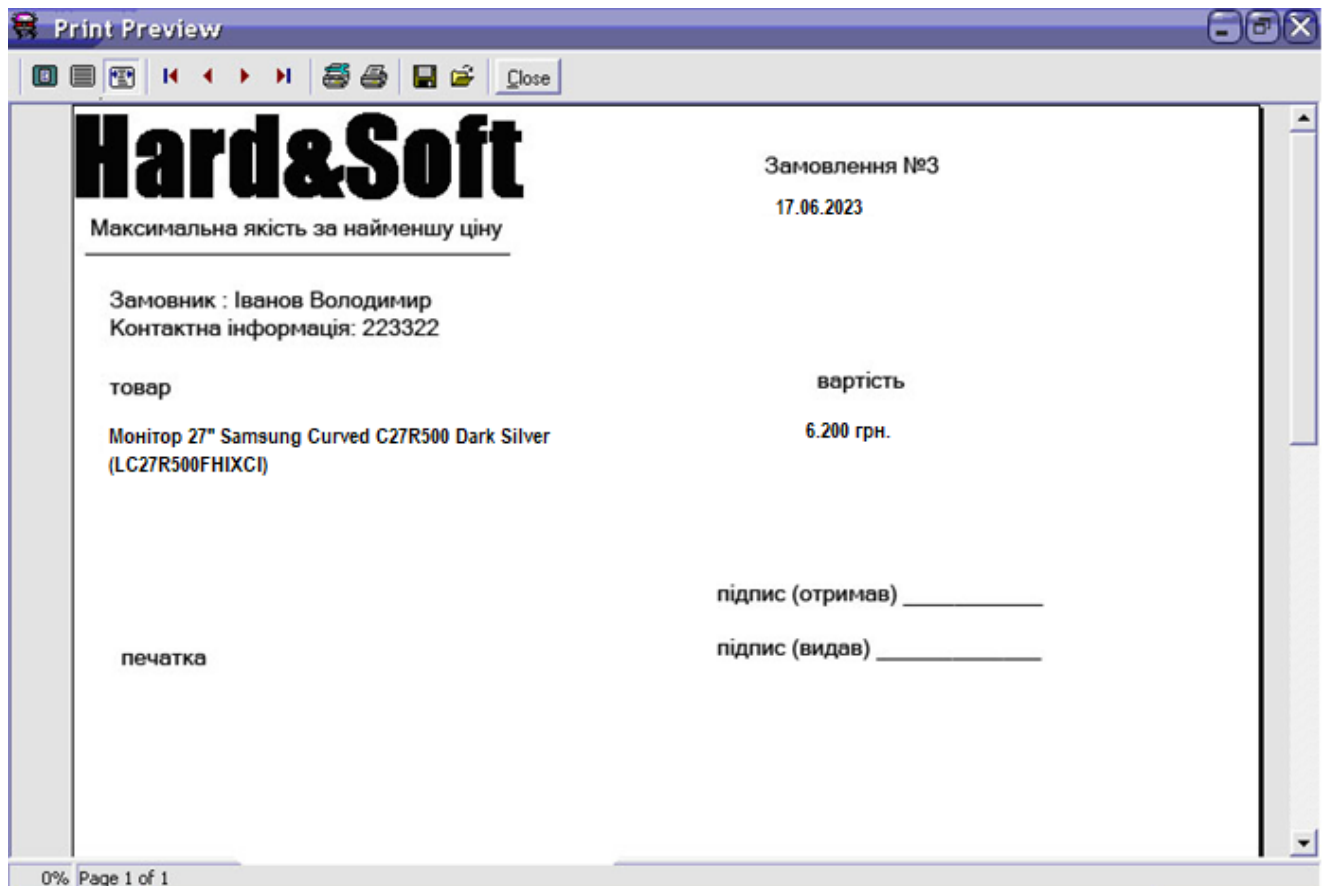


Рисунок 2.6 – Форма замовлення яка відправляється на друк

Форма подана на рисунку 2.6 може бути надрукована. Як приклад на рисунку зображено замовлення на придбання монітора.

### **2.3 Висновок до другого розділу**

В другому розділі кваліфікаційної роботи описані основні компоненти Delphi для роботи з базами даних здійснена розробка інтерфейсу для взаємодії з базою даних та приведено опис основних елементів інтерфейсу програми та кроків по роботі з програмою.

## РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

### 3.1 Зміст та завдання БЖД

Безпека життєдіяльності (БЖД) наука, що вивчає проблеми безпечного перебування людини в довкіллі в процесі різних видів її діяльності (в т.ч. трудової). Вона більш універсальна дисципліна, ніж охорона праці чи цивільна оборона, адже дві останні розглядають лише окремі випадки безпеки в конкретних ситуаціях. Охорона праці цікавиться людиною, яка знаходиться в умовах виробництва, цивільна оборона – в надзвичайних ситуаціях, а безпека життєдіяльності у всіх життєвих обставинах.

В центрі уваги БЖД – людина як самоціль розвитку суспільства, а його лейтмотивом є вислів старогрецького філософа Протагора “Людина є мірило всіх речей”. Людина має цінність не тільки як робоча сила, яку потрібно охороняти в процесі трудової діяльності (чим з певним успіхом і займалася охорона праці), а як неповторний індивідуум, який зберігає свою цінність незалежно від свого конкретного місцезнаходження і виконуваних функцій, признаних суспільством необхідними і корисними (відпочинок, побут, заняття спортом, виконання громадських обов’язків тощо).

Вся сукупність видів людської активності утворює поняття діяльності. Якраз діяльність і вирізняє людину від інших живих істот, вона є специфічно людською формою активності, необхідною умовою існування людського суспільства. Форми діяльності різноманітні. Вони охоплюють практичні, інтелектуальні і духовні процеси, які протікають в побуті, громадській, культурній, виробничій, науковій та інших сферах життя.

Діяльністю займаються всі – діти, дорослі, люди похилого віку, тому безпека діяльності має відношення до всіх людей. Небезпеки підстерігають людей не тільки на виробництві, тому вивчення лише виробничого травматизму в системі загальної безпеки життєдіяльності не висвітлює проблеми.

Безпека – це стан діяльності, при якому з певною ймовірністю виключається прояв небезпек. Безпека – це мета, а безпека життєдіяльності – засоби, шляхи, методи її досягнення.

Актуальність дисципліни ще більше зростає у зв'язку з існуванням аксіоми про потенційну небезпеку діяльності: в жодному виді діяльності неможливо досягнути абсолютної безпеки, будь-яка діяльність потенційно небезпечна.

Завдання БЖД є розробка методів прогнозування, вивчення та ідентифікації шкідливих факторів, їх впливу на людину і довкілля.

БЖД призначена:

- сприяти усвідомленню, що в центрі уваги повинна бути людина, як головна цінність суспільства, та виховати в людині гуманне, свідоме ставлення до питань особистої безпеки та безпеки оточуючих в усіх сферах відносин;

- виробити навички ідентифікації небезпечних та шкідливих факторів і створення сприятливих умов життєдіяльності людей на певній території;

- тримати на контролі проектування нової техніки і технологічних процесів згідно з сучасними вимогами екології і з урахуванням стійкості функціонування господарських об'єктів та технічних систем;

- прогнозувати можливу обстановку і приймати грамотні рішення в умовах надзвичайних ситуацій щодо захисту населення та персоналу об'єктів від можливих негативних наслідків;

- забезпечити якісне засвоєння нового стереотипу поведінки людини з метою виживання в нових природних та антропогенних умовах.

Безпека життєдіяльності базується на досягненнях таких дисциплін, як інженерна психологія, фізіологія людини, охорона праці, екологія, ергономіка, економіка тощо. Вона була і є в центрі уваги людей. З древніх часів до наших днів людина прагнула забезпечити свою безпеку. З розвитком промисловості це потребує спеціальних знань. БЖД особливо актуальна зараз, в добу науково-технічного прогресу. Вона покликана відіграти важливу роль в стабілізації людського суспільства.

Завдання БЖД полягає у чіткому розумінні небезпечних чинників у ситуаціях, що виникають як у середовищі проживання людини, так і у середовищі навчання і праці.

БЖД – це ступінь захисту людини від надзвичайної небезпеки, де під терміном “небезпека” мається на увазі вплив на людину факторів, які можуть викликати відхилення стану її здоров'я від нормального. Природа цих факторів може бути пов'язана як з причинами природного або соціально-економічного характеру (екологічними катастрофами, низьким економічним рівнем життя та ін.), так і з причинами техногенного характеру (з рівнем забруднення навколишнього середовища як наслідком виробничої діяльності людини, аваріями, катастрофами на підприємстві, транспорті, війнами та ін.).

Зміни, що відбувається в нашій країні сьогодні: приватні підприємства, приватизація, кооперація, ринкові відносини, інфляція і т.д., здатні загострити проблеми безпеки – життєдіяльності людини. Так, зараз скорочуються наукові дослідження з проблем безпеки, зменшується чисельність інспекторів, скорочується служба охорони праці. Надзвичайні ситуації в Україні виникають у 5-8 разів частіше, ніж в інших промислово розвинутих країнах світу, в результаті чого щорічно гине понад 1% населення (більше 50 тисяч осіб). Майже третина аварій і нещасних випадків на виробництві пов'язана з незадовільним знанням людей порядку дій у надзвичайних ситуаціях. Це – наслідок серйозних недоліків у навчанні як на виробництві, так і в освітніх закладах. Вивчення предмету БЖД є важливим кроком у розширенні світогляду майбутніх фахівців про безпеку людини та її захист у процесі виробничої і побутової діяльності, в умовах надзвичайних ситуацій мирного та воєнного часу.

Вся сукупність видів людської активності утворює поняття діяльності. Якраз діяльність і вирізняє людину від інших живих істот, вона є специфічно людською формою активності, необхідною умовою існування людського суспільства. Форми діяльності різноманітні. Вони охоплюють практичні,



інтелектуальні і духовні процеси, які протікають в побуті, громадській, культурній, виробничій, науковій та інших сферах життя.

Діяльністю займаються всі – діти, дорослі, люди похилого віку, тому безпека діяльності має відношення до всіх людей. Небезпеки підстерігають людей не тільки на виробництві, тому вивчення лише виробничого травматизму в системі загальної безпеки життєдіяльності не висвітлює проблеми.

Безпека – це стан діяльності, при якому з певною ймовірністю виключається прояв небезпек. Безпека – це мета, а безпека життєдіяльності – засоби, шляхи, методи її досягнення.

БЖД має на меті:

- вивчення структури, змісту і взаємозв'язку життєдіяльності людини із середовищем праці й проживання;
- визначення чинників, причин і параметрів, що сприяють виникненню надзвичайних ситуацій;
- визначення принципів і способів захисту людей в умовах повсякденного життя, а також в умовах надзвичайних ситуацій.

### **3.2 Оцінка ризиків БЖД у життєдіяльності людини**

У виробничих умовах розрізняють індивідуальний і колективний ризик.

*Індивідуальний ризик* характеризує реалізацію небезпеки відповідного виду діяльності для конкретного індивіда.

*Колективний ризик* – це травмування чи загибель двох або більше чоловік від впливу небезпечних та шкідливих виробничих факторів.

Класифікації джерел небезпек та рівні ризику загибелі людини за даними літературних джерел наведені в табл. 4.1.

*Прийнятний ризик* поєднує в собі технічні, економічні, соціальні і політичні аспекти і є деяким компромісом між рівнем безпеки і можливим її досягненням.

Перш за все треба мати на увазі, що підвищення рівня захисту від небезпеки автоматично підвищує загальну вартість виробництва. У вимогах досягнення нульового ризику ситуація може повернутися до людей соціальною трагедією за сценаріями безробіття.

При підвищенні витрат на удосконалення обладнання технічний ризик знижується, але зростає соціальний. Сумарний ризик мінімальний, коли створене необхідне спів відношення між інвестиціями в технічну і соціальну сфери. Ці обставини треба урахувувати під час вибору прийнятного ризику.

Таблиця 3.1

Класифікація джерел і рівнів ризиків загибелі людини в промислово розвинутих країнах ( $R$  – кількість смертельних випадків люд<sup>-1</sup> • рік<sup>-1</sup>)

Джерела	Принципи	Середнє значення
Внутрішнє середовище організму	Генетичні і соматичні захворювання, старіння	$R_{cp} = (0,6-1) \cdot 10^{-2}$
Природне середовище існування	Нещасні випадки від стихійних лих (землетруси, урагани, повені та ін.)	$R_{cp} = 10^{-6}$ ; повені $4 \cdot 10^{-5}$ ; землетруси $3 \cdot 10^{-5}$ ; грози $6 \cdot 10^{-7}$ ; урагани $10^{-3}$
Техносфера	Нещасні випадки в побуті, на транспорті, захворювання від забруднення навколишнього середовища	$R_{cp} = 10^{-3}$
Професійна діяльність	Професійні хвороби, нещасні випадки на виробництві (під час професійної діяльності)	Професійна діяльність: безпечна $R_{cp} < 10^{-4}$ ;

		<p>відносно безпечна  <math>R_{cp}=10^{-4}-10^{-3}</math>;</p> <p>небезпечна  <math>R_{cp}=10^{-4}-10^{-2}</math>;</p> <p>особливо небезпечна  <math>R_{cp}&gt;10^{-2}</math></p>
Соціальне середовище	Самогубства, каліцтва, злочинні дії, військові дії та ін.	$R_{cp}=(0,5-1,5)*10^{-4}$

Рівень прийняттого ризику за міжнародною домовленістю вирішено рахувати в межах  $10^{-7}$ – $10^{-6}$  (смертельних випадків люд<sup>-1</sup> • рік<sup>-1</sup>), а величина  $10^{-6}$  є максимально прийнятним індивідуальним ризиком.

*Вмотивований (обґрунтований) і немотивований (необґрунтований) ризик.* У випадках виробничих аварій, пожеж, з метою рятування людей, які постраждали від аварій та іншого, людині доводиться йти на ризик. Обґрунтованість такого ризику пов'язана з необхідністю надання допомоги людям, що потерпіли, бажанням урятувати від руйнування цінне обладнання чи споруди господарства.

Невиконання робітниками правил безпеки, технологічних процесів, невикористання засобів захисту і таке інше формують необґрунтований ризик, який, як правило, становить передумови виникнення травм і аварій на виробництві.

В технічній сфері поняття ризику визначають дещо інакше порівняно зі звичайними оцінками. Так, за ризик мають кількісну характеристику дії небезпек, які формуються за конкретною діяльністю людини, чи інакше – кількість смертельних випадків, випадків захворювання, випадків тимчасової і стійкої непрацездатності (інвалідності), зумовлених дією на людину конкретної небезпеки відносно до визначеної кількості жителів (робітників).

Ризик формулюється як величина, що визначає добуток величини події на міру можливого її початку.

### 3.3 Соціальні небезпеки для людини

На цей час в Україні загострилася і стає все більш актуальною проблема професійної захворюваності. З 1989 по 1994 рр. кількість щорічно виявлених випадків профзахворювань збільшилась майже в 6 разів (з 2850 до 15 428), а потім почала зменшуватись: до 9016 – в 1995 р., 2558 – в 1997 р. і 2874 – у 1998 р. Таке становище пов'язано як зі зменшенням потоку пенсіонерів, що звертаються за "професійним діагнозом", так і зі станом виявленої профпатології, але ні в якому разі не з поліпшенням умов праці.

Наявність небезпек та шкідливостей у виробничому середовищі, за статистичними даними, сприяло тому, що тільки на виробництвах АПК у 2004 році було смертельно травмовано 265 працівників. На залізничному транспорті загинуло 24 особи, у будівельній галузі 142 особи. Велика кількість людей гине під час ДТП.

Аналіз причин високого рівня травматизму свідчить про низький рівень технологічної дисципліни, безвідповідальне ставлення посадових осіб до стану справ з охорони праці на робочих місцях, вкрай незадовільний стан основних фондів. Такі високі рівні травматизму призводять до значних матеріальних збитків, збільшення собівартості продукції і ускладнення економічного становища країни.

Умови праці і життя формують специфічний психофізіологічний стан населення. В умовах кризових ситуацій в окремих верствах населення можливі виникнення психічних відхилень та захворювань. У зв'язку з тим, що кожна людина реагує на події виходячи з свого стану здоров'я, розуміння і уявлення сьогодення, то її реагування і дії можуть відхилитися від відповідної реакції багатьох членів суспільства. Загострення ситуації може формуватися впливом на свідомість і підсвідомість засобами масової інформації та соціальними технічними засобами.

Професійні хвороби є наслідком патологічного впливу на працівників певних факторів шкідливого виробництва. Нозологічні форми їх визначені затвердженим "Списком професійних захворювань". До нього входять 27 груп хвороб і окремі захворювання (отруєння, пневмокніоз, бериліоз, хронічний пиловий і хронічний токсичний бронхіт, пневмосклероз, емфізема легенів, бронхіальна астма, інфекційні та паразитарні хвороби, гостре і хронічне

перегрівання, тепловий удар, судорожні стани, кесонна хвороба, облітеруючий ендартеріт, тромбофлебіт, варикозне розширення вен, вібраційна хвороба, кохлеарний неврит, бурсит, шийні і попереково-крижові радикулопатії, в тому числі радикуліти, новоутвори, писальний спазм, захворювання м'язів, зв'язок і суглобів, захворювання шкіри, променева хвороба, захворювання, спричинені іонізуючим випромінюванням, катаракта, електроофтальмія, кон'юктивіт, психоневроз та ін.). Список має перелік професій, у яких ця хвороба трапляється, а також виробництв і шкідливих факторів, які спричиняють професійні захворювання.

Усі виявлені випадки професійних захворювань реєструються в санітарно-епідеміологічних установах.

В основу сучасних класифікацій професійних хвороб покладено етіологічний принцип.

1.Професійні хвороби, зумовлені впливом фізичних виробничих факторів.

2.Хвороби, спричинені впливом хімічних виробничих факторів.

3.Хвороби, зумовлені нераціональною організацією праці, нервовоемоційним і фізичним навантаженням, монотонною працею та ін.

Хвороби, зумовлені дією біологічних виробничих факторів.

### **3.4. Висновок до третього розділу**

В третьому розділі кваліфікаційної роботи описано основні питання, які стосуються змісту та завдань БЖД. Також розписані питання оцінка ризиків БЖД у життєдіяльності людини та основні соціальні небезпеки для людини.

## ВИСНОВКИ

Під час виконання кваліфікаційної випускної роботи було розроблено базу даних, яка дозволєє проводити облік замовлень в магазині «Hard & Soft».

Для реалізації поставленого завдання було використано мову програмування Delphi (вимога замовника).

В ході виконання роботи було розроблено:

- структуру бази даних;
- інтерфейс програми;
- проведено тестування програми.

Створене програмне забезпечення взаємодіє з локальною базаою даних InterBase вона є простим і зручним у вкористуванні. Інтерфейс програми дозволяє працювати навіть непідготовленому користувачу завдяки логічно зрозумілим органам керування. Під час написання кваліфікаційної роботи був описаний додатковй розділ «Безпека життєдіяльності, основи хорони праці» в якому описано основні питання, які стосуються змісту та завдань БЖД. Також розписані питання оцінка ризиків БЖД у життєдіяльності людини та основні соціальні небезпеки для людини.

**ПЕРЕЛІК ДЖЕРЕЛ**

1. В.Э. Гофманн, А.Д. Хомоненко. Delphi. СПб.:БХВ-Петербург, 2002.- 1152 с.
2. Б. Карпов. Delphi: специальный справочник. СПб.:Питер, 2001. – 688 с.
3. Джеймс Р. Грофф, Пол Н. Вайнберг. SQL: полное руководство. – К.: ВНУ, 2000. – 608 с.
4. Губський А.І., Цивільна оборона. - К.: Міністерство освіти, 995. - 216 с.
5. Бородич Ю.С., Вальвачев А.Н., Кузьмич А.И. Паскаль для персональных компьютеров.-Минск., 1991.
6. Пильщиков В.Н. Сборник упражнений по языку Паскаль.-М., 1989.
7. Програмування. Практикум / Укл.: Семенюк А.Д., Сопронюк Ф.О. – Чернівці: Рута, 2001.– 143 с.
8. Фаронов В.В. Delphi. Программирование на языке высокого уровня.– СПб.: Питер, 2004.– 640 с.
9. Гофман В.Э., Хомоненко А.Д. Delphi 6.– М., 2002.– 1145 с.
- 10.Культин Н.Б. Delphi 6. Программирование на Object Pascal.– М., 2002.– 526 с.
- 11.Електронні таблиці Microsoft Excel: Методичні вказівки до лабораторних робіт / Укл.: В.С. Сікора, І.В. Юрченко.– Чернівці: Рута, 2002.– 48 с.
- 12.Основи інформатики: Методичні вказівки до лабораторних робіт: У 2 ч./ Укл.: І.В. Юрченко.– Чернівці: Рута, 2000.– 79 с.
- 13.Система управління базами даних Microsoft Access: Методичні вказівки до лабораторних робіт / Укл.: В.С. Сікора, І.В. Юрченко.– Чернівці: Рута, 2002.– 40 с.



14. Комп'ютерні мережі: Методичні вказівки до лабораторних робіт / Укл.: В.С. Сікора, І.В. Юрченко.– Чернівці: Рута, 2002.– 43 с.
15. Операційна система Microsoft Windows: Методичні вказівки до лабораторних робіт / Укл.: В.С. Сікора, І.В. Юрченко.– Чернівці: Рута, 2003.– 48 с.
16. Текстовий редактор Microsoft Word: Методичні вказівки до лабораторних робіт / Укл.: В.С. Сікора, І.В. Юрченко.– Чернівці: Рута, 2003.– 56 с.
17. Семчук А.Р., Юрченко І.В. Економічна інформатика. Навчальний посібник.– Чернівці: МВІЦ "Місто", 2008.– 426 с.
18. Симонович С., Евсеев Г. Практическая информатика: универсальный курс.– М.: АСТ–ПРЕСС; Инфорком–Пресс, 1999.– 480 с.
19. В.А. Гайченко, Г.М. Коваль : “Основи БЖД людини” : Навч. посіб. – 2-ге вид. стереотип. – К.: МАУП, 2004.-232с.: іл.-Бібліогр.: с. 225-226
20. Джигрей В.С., Жидецький В.Ц. “Безпека життєдіяльності”. Підручник.- Вид. 4-те, доповнене. – Львів: Афіша, 2001. – 256с.
21. Скобло Ю. С, Соколовська Т. Б., Мазоренко Д. І., Тіщенко Л. М., Троянов М. М. – Безпека життєдіяльності: Навчальний посібник для вищих навчальних закладів III – IV рівнів акредитації. – Київ: Кондор, 2003. – 424с.
22. Желібо Є.П., Заверуха Н.М., Зацарний В.В. : “Безпека життєдіяльності” : Навчальний посібник для студентів вищих закладів освіти України I-IV рівнів акредитації / За ред. Є.П. Желібо, В.М. Пічі. – Львів : Піча Ю.В., К.: “Каравела”, Львів:, Новий світ – 2000, 2002, - 328с.
23. Пам'ятка-інструктаж з питань охорони праці для студентів Тернопільського державного технічного університету імені Івана Пулюя. – Тернопіль.

# ДОДАТКИ

## Назва додатка А

## Фрагмент лістингу програми «головна форма»

```
unit PC;

interface

uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls,
Forms, Dialogs,
    ExtCtrls, StdCtrls, DBCtrls, Buttons;

type
    collens=record
        kompl:string[20];
        name:string[20];
        myprise:string[7];
        noprise:string[7];
    end;
    stok=record
        name:string[40];
        prise:string[5];
    end;
    database=record
        mb:stok;
        cdu:stok;
        ram:stok;
        svga:stok;
        hdd:stok;
        fdd:stok;
        monitor:stok;
        cd:stok;
        sb:stok;
        spiker:stok;
        modem:stok;
        printer:stok;
        skaner:stok;
    end;
    podsh=record
        number:string[4];
        firstname:string[20];
        secondname:string[10];
        tel:string[7];
        begindate:string[8];
        enddate:string[8];
```

```
        ok:boolean;
        prise:string[6];
        bd:database;
end;
sum=record
    sum1:integer;
    sum2:integer;
end;
TComplect = class(TForm)
    Label1: TLabel;
    Bevel1: TBevel;
    Exit: TButton;
    Panel1: TPanel;
    Label2: TLabel;
    Label3: TLabel;
    GroupBox1: TGroupBox;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    ComboBox2: TComboBox;
    ComboBox3: TComboBox;
    ComboBox4: TComboBox;
    ComboBox5: TComboBox;
    ComboBox6: TComboBox;
    ComboBox7: TComboBox;
    GroupBox2: TGroupBox;
    Label11: TLabel;
    Label12: TLabel;
    Label13: TLabel;
    ComboBox8: TComboBox;
    ComboBox9: TComboBox;
    ComboBox10: TComboBox;
    GroupBox3: TGroupBox;
    Label14: TLabel;
    Label15: TLabel;
    Label16: TLabel;
    ComboBox11: TComboBox;
    ComboBox12: TComboBox;
    ComboBox13: TComboBox;
    Label18: TLabel;
    GroupBox4: TGroupBox;
    Edit1: TEdit;
    ComboBox1: TComboBox;
    Label17: TLabel;
    Label19: TLabel;
    Label20: TLabel;
    Label21: TLabel;
    Label22: TLabel;
    Label23: TLabel;
```

```
Label24: TLabel;  
Label25: TLabel;  
Label26: TLabel;  
Label27: TLabel;  
Label28: TLabel;  
Label29: TLabel;  
Label31: TLabel;  
Label32: TLabel;  
Edit3: TEdit;  
Edit4: TEdit;  
CheckBox1: TCheckBox;  
Label33: TLabel;  
Label34: TLabel;  
Label35: TLabel;  
Edit5: TEdit;  
Label36: TLabel;  
Label37: TLabel;  
Label38: TLabel;  
Label39: TLabel;  
Label41: TLabel;  
Label42: TLabel;  
Label43: TLabel;  
Label44: TLabel;  
Label45: TLabel;  
Label46: TLabel;  
Label47: TLabel;  
Label48: TLabel;  
Label49: TLabel;  
Label50: TLabel;  
Label51: TLabel;  
Label52: TLabel;  
Label53: TLabel;  
Label54: TLabel;  
Label55: TLabel;  
Button1: TButton;  
Button2: TButton;  
Edit2: TEdit;  
BitBtn1: TBitBtn;  
BitBtn2: TBitBtn;  
BitBtn3: TBitBtn;  
Label40: TLabel;  
BitBtn4: TBitBtn;  
Edit6: TEdit;  
Label56: TLabel;  
Label30: TLabel;  
procedure ExitClick(Sender: TObject);  
procedure FormCreate(Sender: TObject);  
procedure ComboBox1Change(Sender: TObject);  
procedure ComboBox2Change(Sender: TObject);  
procedure ComboBox3Change(Sender: TObject);  
procedure ComboBox4Change(Sender: TObject);  
procedure ComboBox5Change(Sender: TObject);  
procedure ComboBox6Change(Sender: TObject);
```

```

    procedure ComboBox7Change(Sender: TObject);
    procedure ComboBox8Change(Sender: TObject);
    procedure ComboBox9Change(Sender: TObject);
    procedure ComboBox10Change(Sender: TObject);
    procedure ComboBox11Change(Sender: TObject);
    procedure ComboBox12Change(Sender: TObject);
    procedure ComboBox13Change(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Edit1Change(Sender: TObject);
    procedure Edit2Change(Sender: TObject);
    procedure Edit3Change(Sender: TObject);
    procedure Edit4Change(Sender: TObject);
    procedure Edit5Change(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);

private
    { Private declarations }
public
    { Public declarations }
end;

var
    Complect: TComplect;
    g:file of podsh;
    npod:array[1..9999]of podsh;
    npodsh:podsh;
    f:file of collens;
    indexs:array[1..200]of collens;
    index:collens;
    i,num:integer;
    su:array[1..15]of sum;
    secret:boolean;
implementation

uses PL, BD, Unit2;

{$R *.DFM}

procedure TComplect.Button2Click(Sender: TObject);
var kkk:integer;
begin
    num:=num+1;
    npod[num].number:=label37.caption;
    npod[num].firstname:=edit1.text;
    npod[num].secondname:=edit2.text;
    npod[num].tel:=edit3.text;
    npod[num].begindate:=edit4.text;
    npod[num].enddate:=edit5.text;
    npod[num].ok:=checkbox1.checked;
    npod[num].prise:=label39.caption;

```

```

npod[num].bd.mb.name:=combobox1.text;
npod[num].bd.mb.prise:=label17.caption;
npod[num].bd.cdu.name:=combobox2.text;
npod[num].bd.cdu.prise:=label18.caption;
npod[num].bd.ram.name:=combobox3.text;
npod[num].bd.ram.prise:=label19.caption;
npod[num].bd.svga.name:=combobox4.text;
npod[num].bd.svga.prise:=label20.caption;
npod[num].bd.hdd.name:=combobox5.text;
npod[num].bd.hdd.prise:=label21.caption;
npod[num].bd.fdd.name:=combobox6.text;
npod[num].bd.fdd.prise:=label22.caption;
npod[num].bd.monitor.name:=combobox7.text;
npod[num].bd.monitor.prise:=label23.caption;
npod[num].bd.cd.name:=combobox8.text;
npod[num].bd.cd.prise:=label24.caption;
npod[num].bd.sb.name:=combobox9.text;
npod[num].bd.sb.prise:=label25.caption;
npod[num].bd.spiker.name:=combobox10.text;
npod[num].bd.spiker.prise:=label26.caption;
npod[num].bd.modem.name:=combobox11.text;
npod[num].bd.modem.prise:=label27.caption;
npod[num].bd.printer.name:=combobox12.text;
npod[num].bd.printer.prise:=label28.caption;
npod[num].bd.skanner.name:=combobox13.text;
npod[num].bd.skanner.prise:=label29.caption;
rewrite(g);
for kkk:=1 to num do write(g,npod[kkk]);
edit1.text:='';
edit2.text:='';
edit3.text:='';
edit4.text:='';
edit5.text:='';
end;

procedure resetfile2;
begin
reset(g);
  num:=0;
  while not eof(g) do
    begin
      num:=num+1;
      read(g,npod[num]);

    end;
end;

procedure resetfile;
begin
reset(f);
  i:=1;

```

```

while not eof(f) do
  begin
    read(f, indexs[i]);
    i:=i+1;
  end;
end;

function clicks:boolean;
begin
  clicks:=(complect.edit1.text<>'') and
(complect.edit2.text<>'') and
  and (complect.edit3.text<>'') and
(complect.edit4.text<>'') and (complect.edit5.text<>'') and (su[14].sum1>0);
end;

procedure summa(txt,txt1,txt2:string;ij:integer);
var k,j:integer;
begin
  su[14].sum1:=su[14].sum1-su[ij].sum1;
  su[14].sum2:=su[14].sum2-su[ij].sum2;
  k:=1;
  while (k<=i) do
    if indexs[k].name=txt2 then
      begin
        val(indexs[k].myprise,su[ij].sum1,j);
        val(indexs[k].noprise,su[ij].sum2,j);
        su[14].sum1:=su[14].sum1+su[ij].sum1;
        su[14].sum2:=su[14].sum2+su[ij].sum2;
        k:=i+1;
      end
    else
      begin
        k:=k+1;
        su[ij].sum1:=0;
        su[ij].sum2:=0;
      end;
  su[15].sum2:=su[14].sum1-su[14].sum2;
  complect.button2.visible:=clicks;
end;

procedure TComplect.ExitClick(Sender: TObject);
begin
  close;
end;

procedure TComplect.FormCreate(Sender: TObject);
var k:integer;
    s:string;
begin
  secret:=false;
  str(num,s);
  label37.caption:=s;

```



```

    for k:=1 to 15 do
    begin
        su[k].sum1:=0;
        su[k].sum2:=0;
    end;
    for k:=0 to i do
    begin
        if          indexes[k].kompl='Ìàòåðèíñêàÿ'          then
        combobox1.items.add(indexes[k].name);
        if          indexes[k].kompl='Ïðîöåññîð'          then
        combobox2.items.add(indexes[k].name);
        if          indexes[k].kompl='RAM'                  then
        combobox3.items.add(indexes[k].name);
        if          indexes[k].kompl='Âèääî'                then
        combobox4.items.add(indexes[k].name);
        if          indexes[k].kompl='HDD'                  then
        combobox5.items.add(indexes[k].name);
        if          indexes[k].kompl='FDD'                  then
        combobox6.items.add(indexes[k].name);
        if          indexes[k].kompl='Ìííèèòîð'            then
        combobox7.items.add(indexes[k].name);
        if          indexes[k].kompl='CD-ROM'                then
        combobox8.items.add(indexes[k].name);
        if          indexes[k].kompl='Çâóêîââÿ'            then
        combobox9.items.add(indexes[k].name);
        if          indexes[k].kompl='Êîëîíîèè'            then
        combobox10.items.add(indexes[k].name);
        if          indexes[k].kompl='Ïîääî'                then
        combobox11.items.add(indexes[k].name);
        if          indexes[k].kompl='Ïðèèòåð'              then
        combobox12.items.add(indexes[k].name);
        if          indexes[k].kompl='Ñêàíáð'                then
        combobox13.items.add(indexes[k].name);
    end;
end;

procedure TComplect.ComboBox1Change(Sender: TObject);
var s:string;
begin
    resetfile;
    //if combobox1.text='<None>' then label17.caption:='0';
    summa(label17.caption,label41.caption,combobox1.text,1);
    str(su[1].sum1,s);
    label17.caption:=s;
    str(su[14].sum1,s);
    label39.caption:=s;
    label30.caption:=s;
    if secret then
    begin
        str(su[1].sum2,s);
        label41.caption:=s;
        str(su[14].sum2,s);
        label54.caption:=s;
    end;
end;

```

```

        str(su[15].sum2,s);
        label55.caption:=s;
    end;

end;

procedure TComplect.ComboBox2Change(Sender: TObject);
var s:string;
begin
    resetfile;
    if combobox2.text='<None>' then label18.caption:='0';
    summa(label18.caption,label42.caption,combobox2.text,2);
    str(su[2].sum1,s);
    label18.caption:=s;
    str(su[14].sum1,s);
    label30.caption:=s;
    label39.caption:=s;
    if secret then
        begin
            str(su[2].sum2,s);
            label42.caption:=s;
            str(su[14].sum2,s);
            label54.caption:=s;
            str(su[15].sum2,s);
            label55.caption:=s;
        end;
    clicks;
end;

procedure TComplect.ComboBox3Change(Sender: TObject);
var s:string;
begin
    resetfile;
    if combobox3.text='<None>' then label19.caption:='0';
    summa(label19.caption,label43.caption,combobox3.text,3);
    str(su[3].sum1,s);
    label19.caption:=s;
    str(su[14].sum1,s);
    label30.caption:=s;
    label39.caption:=s;
    if secret then
        begin
            str(su[3].sum2,s);
            label43.caption:=s;
            str(su[14].sum2,s);
            label54.caption:=s;
            str(su[15].sum2,s);
            label55.caption:=s;
        end;
end;

procedure TComplect.ComboBox4Change(Sender: TObject);
var s:string;

```

```

begin
  resetfile;
  if combobox4.text='<None>' then label20.caption:='0';
  summa(label20.caption,label44.caption,combobox4.text,4);
  str(su[4].sum1,s);
  label20.caption:=s;
  str(su[14].sum1,s);
  label30.caption:=s;
  label39.caption:=s;
  if secret then
    begin
      str(su[4].sum2,s);
      label44.caption:=s;
      str(su[14].sum2,s);
      label54.caption:=s;
      str(su[15].sum2,s);
      label55.caption:=s;
    end;
end;

procedure TComplect.ComboBox5Change(Sender: TObject);
var s:string;
begin
  resetfile;
  if combobox5.text='<None>' then label21.caption:='0';
  summa(label21.caption,label45.caption,combobox5.text,5);
  str(su[5].sum1,s);
  label21.caption:=s;
  str(su[14].sum1,s);
  label30.caption:=s;
  label39.caption:=s;
  if secret then
    begin
      str(su[5].sum2,s);
      label45.caption:=s;
      str(su[14].sum2,s);
      label54.caption:=s;
      str(su[15].sum2,s);
      label55.caption:=s;
    end;
end;

procedure TComplect.ComboBox6Change(Sender: TObject);
var s:string;
begin
  resetfile;
  if combobox6.text='<None>' then label22.caption:='0';
  summa(label22.caption,label46.caption,combobox6.text,6);
  str(su[6].sum1,s);
  label22.caption:=s;
  str(su[14].sum1,s);
  label30.caption:=s;
  label39.caption:=s;

```

```

    if secret then
    begin
        str(su[6].sum2,s);
        label46.caption:=s;
        str(su[14].sum2,s);
        label54.caption:=s;
        str(su[15].sum2,s);
        label55.caption:=s;
    end;
end;

procedure TComplect.ComboBox7Change(Sender: TObject);
var s:string;
begin
    resetfile;
    if combobox7.text='<None>' then label23.caption:='0';
    summa(label23.caption,label47.caption,combobox7.text,7);
    str(su[7].sum1,s);
    label23.caption:=s;
    str(su[14].sum1,s);
    label30.caption:=s;
    label39.caption:=s;
    if secret then
    begin
        str(su[7].sum2,s);
        label47.caption:=s;
        str(su[14].sum2,s);
        label54.caption:=s;
        str(su[15].sum2,s);
        label55.caption:=s;
    end;
end;

procedure TComplect.ComboBox8Change(Sender: TObject);
var s:string;
begin
    resetfile;
    if combobox8.text='<None>' then label24.caption:='0';
    summa(label24.caption,label48.caption,combobox8.text,8);
    str(su[8].sum1,s);
    label24.caption:=s;
    str(su[14].sum1,s);
    label30.caption:=s;
    label39.caption:=s;
    if secret then
    begin
        str(su[8].sum2,s);
        label48.caption:=s;
        str(su[14].sum2,s);
        label54.caption:=s;
        str(su[15].sum2,s);
        label55.caption:=s;
    end;
end;

```

```
end;
```

```
procedure TComplect.ComboBox9Change(Sender: TObject);
var s:string;
begin
  resetfile;
  if combobox9.text='<None>' then label25.caption:='0';
  summa(label25.caption,label49.caption,combobox9.text,9);
  str(su[9].sum1,s);
  label25.caption:=s;
  str(su[14].sum1,s);
  label30.caption:=s;
  label39.caption:=s;
  if secret then
    begin
      str(su[9].sum2,s);
      label49.caption:=s;
      str(su[14].sum2,s);
      label54.caption:=s;
      str(su[15].sum2,s);
      label55.caption:=s;
    end;
end;
```

```
procedure TComplect.ComboBox10Change(Sender: TObject);
var s:string;
begin
  resetfile;
  if combobox10.text='<None>' then label26.caption:='0';
```

```
summa(label26.caption,label50.caption,combobox10.text,10);
  str(su[10].sum1,s);
  label26.caption:=s;
  str(su[14].sum1,s);
  label30.caption:=s;
  label39.caption:=s;
  if secret then
    begin
      str(su[10].sum2,s);
      label50.caption:=s;
      str(su[14].sum2,s);
      label54.caption:=s;
      str(su[15].sum2,s);
      label55.caption:=s;
    end;
end;
```

```
procedure TComplect.ComboBox11Change(Sender: TObject);
var s:string;
begin
  resetfile;
  if combobox11.text='<None>' then label27.caption:='0';
```

```

summa (label27.caption, label51.caption, combobox11.text, 11);
    str (su[11].sum1, s);
    label27.caption:=s;
    str (su[14].sum1, s);
    label30.caption:=s;
    label39.caption:=s;
    if secret then
        begin
            str (su[11].sum2, s);
            label51.caption:=s;
            str (su[14].sum2, s);
            label54.caption:=s;
            str (su[15].sum2, s);
            label55.caption:=s;
        end;
end;

procedure TComplect.ComboBox12Change (Sender: TObject);
var s:string;
begin
    resetfile;
    if combobox12.text='<None>' then label28.caption:='0';

summa (label28.caption, label52.caption, combobox12.text, 12);
    str (su[12].sum1, s);
    label28.caption:=s;
    str (su[14].sum1, s);
    label30.caption:=s;
    label39.caption:=s;
    if secret then
        begin
            str (su[12].sum2, s);
            label52.caption:=s;
            str (su[14].sum2, s);
            label54.caption:=s;
            str (su[15].sum2, s);
            label55.caption:=s;
        end;
end;

procedure TComplect.ComboBox13Change (Sender: TObject);
var s:string;
begin
    resetfile;
    if combobox13.text='<None>' then label29.caption:='0';

summa (label29.caption, label53.caption, combobox13.text, 13);
    str (su[13].sum1, s);
    label29.caption:=s;
    str (su[14].sum1, s);
    label30.caption:=s;
    label39.caption:=s;

```

```
    if secret then
        begin
            str(su[13].sum2,s);
            label53.caption:=s;
            str(su[14].sum2,s);
            label54.caption:=s;
            str(su[15].sum2,s);
            label55.caption:=s;
        end;
    end;

procedure TComplect.Button1Click(Sender: TObject);
var s:string;
begin
    secret:=not secret;
    if secret then
        begin
            str(su[1].sum2,s);
            label41.caption:=s;
            str(su[2].sum2,s);
            label42.caption:=s;
            str(su[3].sum2,s);
            label43.caption:=s;
            str(su[4].sum2,s);
            label44.caption:=s;
            str(su[5].sum2,s);
            label45.caption:=s;
            str(su[6].sum2,s);
            label46.caption:=s;
            str(su[7].sum2,s);
            label47.caption:=s;
            str(su[8].sum2,s);
            label48.caption:=s;
            str(su[9].sum2,s);
            label49.caption:=s;
            str(su[10].sum2,s);
            label50.caption:=s;
            str(su[11].sum2,s);
            label51.caption:=s;
            str(su[12].sum2,s);
            label52.caption:=s;
            str(su[13].sum2,s);
            label53.caption:=s;
            str(su[14].sum2,s);
            label54.caption:=s;
            str(su[15].sum2,s);
            label55.caption:=s;
        end
    else begin
        label41.caption:='';
        label42.caption:='';
        label43.caption:='';
        label44.caption:='';
    end;
end;
```

```
    label45.caption:='';
    label46.caption:='';
    label47.caption:='';
    label48.caption:='';
    label49.caption:='';
    label50.caption:='';
    label51.caption:='';
    label52.caption:='';
    label53.caption:='';
    label54.caption:='';
    label55.caption:='';
    end;
end;

procedure TComplect.Edit1Change(Sender: TObject);
begin
    complect.button2.visible:=clicks;
end;

procedure TComplect.Edit2Change(Sender: TObject);
begin
    complect.button2.visible:=clicks;
end;

procedure TComplect.Edit3Change(Sender: TObject);
begin
    complect.button2.visible:=clicks;
end;

procedure TComplect.Edit4Change(Sender: TObject);
begin
    complect.button2.visible:=clicks;
end;

procedure TComplect.Edit5Change(Sender: TObject);
begin
    complect.button2.visible:=clicks;
end;

procedure TComplect.BitBtn3Click(Sender: TObject);
begin
    Form2.Show;
end;

begin
    assign(f, 'bd.bd');
    assign(g, 'orders.1');

    resetfile2;
    resetfile;
    close(g);
    close(f);
end.
```