**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

**КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ**

**КУРСОВИЙ ПРОЄКТ: ОБ’ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ**

на тему: «Розроблення програмної системи для роботи з даними про товари магазина побутової техніки»

Студента 2 курсу 6.04.121.013.22.2 групи

Спеціальності 121 "Інженерія програмного забезпечення"

першого (бакалаврського) рівня

Максакова В.О.

Керівник: доцент кафедри ІС

Парфьонов Ю.Є.

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_Оцінка: ECTS \_\_\_\_\_

Члени комісії

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_Юрій ПАРФЬОНОВ\_\_\_\_

                 (підпис)                        (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_Дмитро ГОЛУБНИЧИЙ\_\_

                 (підпис)                        (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_Володимир БРЕДІХІН\_\_

                 (підпис)                        (прізвище та ініціали)

м. Харків – 2024 рік

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ СЕМЕНА КУЗНЕЦЯ

Факультет інформаційних технологій

Кафедра інформаційних систем

ЗАВДАННЯ

на курсовий проєкт: Об’єктно-орієнтоване програмування

студенту 2-го курсу групи 6.04.121.013.22.2

Владиславу МАКСАКОВУ

1. Тема проєкту: «Розроблення програмної системи для роботи з даними про товари магазина побутової техніки»

2. Термін здачі студентом закінченого проєкту «10» травня 2024 р.

3. Вхідні дані до проєкту: літературні джерела, технічна документація щодо розроблення програм, ДСТУ з оформлення документації.

4. Зміст пояснювальної записки:

Вступ. Специфікація проекту. Програмна документація. Висновки. Додатки.

5. Перелік графічного матеріалу: UML-діаграма класів програмної системи.

Календарний план

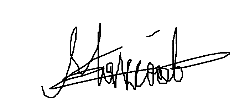
виконання курсового проєкту

|  |  |  |  |
| --- | --- | --- | --- |
| № з/п | Назва етапу роботи | Строк виконання за планом | Відмітка про виконання |
| 1. | З‘ясування загальної постановки завдання. Розроблення загальної архітектури програмної системи. Розроблення чернетки першого розділу пояснювальної записки. | 04.03.24 |  |
| 2. | Деталізація завдань курсового проєктування. Уточнення архітектури програмної системи. Розроблення остаточного варіанту першого розділу пояснювальної записки. |  |  |
| 3. | Програмна реалізація, налагодження та тестування застосунку. | 08.04.24 |  |
| 4. | Остаточне налагодження та тестування застосунку. Розроблення чернетки другого розділу пояснювальної записки, висновків, списку використаних джерел,  додатків, вступу, реферату). | 29.04.24 |  |
| 5. | Розроблення остаточного варіанту пояснювальної записки та підготовка електронної презентації |  |  |

Дата видачі завдання "19" лютого 2024 р.

Керівник  к.т.н., доц.  кафедри ІС\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Юрій ПАРФЬОНОВ

(підпис)



Завдання прийняв до виконання \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Владислав МАКСАКОВ

(підпис)

РЕФЕРАТ

Пояснювальна записка до курсового проекту: 34 с., 21 рис., 5 табл., 5 джерел.

Об'єктом дослідження було розроблення програмного продукту для роботи з даними про товари магазина побутової техніки.

Мета даної роботи є розробка програмного продукту, який забезпечить ефективне управління даними про товари магазина побутової техніки, включаючи їхнє додавання, редагування, видалення та аналіз.

Для досягнення поставленої мети було використано методи аналізу вимог користувачів, проектування бази даних, розробки програмного забезпечення та тестування. Для розробки програмного продукту використовувалися Microsoft Visual Studio, мова програмування C#, а також Entity Framework для зручного взаємодії з базою даних SQLite.

У результаті роботи був розроблений програмний продукт, який надає можливість ефективного управління даними про товари магазина побутової техніки. Новизна полягає у впровадженні інтуїтивного інтерфейсу користувача та функцій аналізу даних для підвищення ефективності роботи.

Основні технологічні й техніко-експлуатаційні характеристики та показники - програмний продукт реалізований на платформі Microsoft Visual Studio з використанням мови програмування C#. Для збереження даних використовується база даних SQLite.

Результати цієї роботи можуть бути використані як основа для подальшого розвитку програмних продуктів для управління даними в сфері роздрібної торгівлі.

З урахуванням швидкого розвитку технологій та зростання потреб споживачів, можна очікувати подальшого розвитку програмного продукту з удосконаленням інтерфейсу та функціональності.

ТОВАРИ, МАГАЗИН, ПОБУТОВА ТЕХНІКА, УПРАВЛІННЯ ДАНИМИ, АНАЛІЗ ДАНИХ.

ЗМІСТ

[ВСТУП 6](#_Toc166851875)

[1. СПЕЦИФІКАЦІЯ ПРОЕКТУ 7](#_Toc166851876)

[1.1 Постановка завдання 7](#_Toc166851877)

[1.2 Вимоги до програмного забезпечення 8](#_Toc166851878)

[1.3 Математичний опис задачі 9](#_Toc166851879)

[2. ПРОГРАМНА ДОКУМЕНТАЦІЯ 10](#_Toc166851880)

[2.1 Архітектура програмного забезпечення 10](#_Toc166851881)

[2.2 Тестування програмної системи 12](#_Toc166851882)

[2.3 Розгортання програмного продукту 17](#_Toc166851883)

[2.4 Керівництво користувача 21](#_Toc166851884)

[2.4.1 Призначення програмного продукту 21](#_Toc166851885)

[2.4.2 Використання програмного продукту 21](#_Toc166851886)

[2.4.3 Повідомлення користувачеві при 26](#_Toc166851887)

[ВИСНОВКИ 31](#_Toc166851888)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 32](#_Toc166851889)

[Додаток А 33](#_Toc166851890)

ВСТУП

У світі сучасних технологій надзвичайно швидко розвиваються нові можливості для підприємств у впровадженні автоматизації та оптимізації бізнес-процесів. Однією з основних галузей, де це стає особливо важливим, є управління даними про товари на складі магазину. Використання спеціалізованого програмного забезпечення стає необхідністю для підприємств, щоб забезпечити зручний і ефективний облік і контроль за запасами.

Потреба у такому програмному забезпеченні обумовлюється різноманітністю галузей, наприклад у торгівлі. Навіть при наявності різних комерційних рішень, багато компаній вирішують розробити власне програмне забезпечення. Це може бути спонукано бажанням мати більш гнучкий інструмент, що краще відповідає їхнім потребам, або стремлінням заощадити на вартості ліцензій або пристосування програми до конкретних потреб.

Розробка програмного забезпечення для управління даними є важливим етапом в навчанні для студентів, особливо у контексті об'єктно-орієнтованого програмування.

Цей курсовий проект допомагає студентам не лише закріпити свої знання, а й отримати практичні навички у вирішенні реальних завдань, пов'язаних з управлінням даними у конкретній сфері, наприклад, управлінням складом. Такий проект дозволяє студентам виявити свій талант у програмуванні та аналізі вимог до програмного забезпечення.

1. СПЕЦИФІКАЦІЯ ПРОЕКТУ

1.1 Постановка завдання

Розробити програмний продукт для роботи з даними про товари магазину побутової техніки.

Підприємство веде облік побутових приладів, з використанням відомості в «паперовому» вигляді. Кількість запитів необмежена.

Програмний продукт повинен генерувати відомості у табличному вигляді із різних задач.

Таблиця має 7 стовпців «ID» – унікальний ідентифікаційний ключ товару, «Type» – тип товару, «Supplier» – виробник товару, «Color» – колір товару, «Price» - ціна товару, «Amount» – кількість товару, «Cost» - повна ціна товару яка розраховується за формулою 1.1:

Costi = Pricei \* Amounti (1.1)

де і – номер індексу, 1 ≤ і ≤ n.

В кінці таблиці є рядок «TotalCost», який показує загальну суму усіх товарів, обчислену за формулою 1.2.

(1.2)

де і – номер індексу, 1 ≤ і ≤ n;

n – загальна кількість записів побутової техніки;

Costi – повна ціна товару.

Таблиця 1.1 – приклад заповнення відомості

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Type | Supplier | Color | Price | Amount | Cost |
| 1 | Кондиціонер | Electrolux | Білий | 36000 | 25 | 900 000 |
| 2 | Соковижималка | Bosch | Сірий | 7550 | 55 | 188 750 |
| … | … | … | … | … | … | … |
| TotalCost | | 1 088 750 | |

Працівник може вносити данні про новоприбулий товар, і вводити данні по відповідним полям «Type», «Supplier», «Color», «Price», «Amount». Після введення поле «Cost» буде автоматично розрахована по формулі 1.1, а поле «TotalCost» автоматично розраховане по формулі 1.2.

1.2 Вимоги до програмного забезпечення

Функціональні вимоги:

а) Створення бази даних SQLite для зберігання інформації про побутову техніку та додавання нових записів у цю базу.

б) Виведення на екран користувача всіх даних про побутову техніку, які зберігаються у сховищі.

в) Можливість користувача додавати нові дані про побутову техніку до сховища.

г) Оновлення будь-якого запису даних про побутову техніку, який знаходиться у сховищі.

ґ) Видалення будь-якого запису даних про побутову техніку зі сховища.

д) Розрахунок та відображення користувачу підсумкової інформації, наприклад, загальну суму усіх товарів побутових приладів у сховищі або їхньої загальної вартості.

е) Перевірка правильності введених користувачем основних даних перед додаванням чи оновленням запису в сховищі.

є) Виведення на екран користувача попереджень у разі виникнення помилок чи неправильного введення даних.

Нефункціональні вимоги:

Мінімальні вимоги до інтерфейсу користувача:

a) інтерфейс користувача має складатись з текстових повідомлень українською мовою, які відносяться до введення та виведення основних та допоміжних даних. Використання транслітерації заборонено;

б) наявність текстового меню для вибору користувачем варіантів дій;

в) данні, що зберігаються в сховищі даних, мають відображатись у табличному вигляді.

Вимоги до архітектури програми:

а) використання не менше однієї структури даних із стандартної бібліотеки колекцій;

б) використання потоків введення-виведення даних;

г) використання механізму винятків для обробки помилок.

Вимоги до вихідного коду застосунку:

а) додержання принципу інкапсуляції щодо рівнів доступу до полів та методів класів.

б) вихідний код кожного з класів програми повинен міститися в окремому файлі.

в) імена класів, інтерфейсів, полів класів, властивостей, методів, локальних змінних та констант тощо повинні мати смислове навантаження в певній предметній області.

1.3 Математичний опис задачі

«Cost» – це загальна сума товару одного типу і виробника. Формула для обчислення:

Costi = Pricei \* Amounti (1.1)

де і – номер індексу, 1 ≤ і ≤ n.

«TotalCost» – це загальна сума усіх товарів. виробника. Формула для обчислення:

(1.2)

де і – номер індексу, 1 ≤ і ≤ n;

n – загальна кількість записів побутової техніки;

Costi – повна ціна товару.

2. ПРОГРАМНА ДОКУМЕНТАЦІЯ

2.1 Архітектура програмного забезпечення

Діаграму класів програми наведено на рис. 2.1.

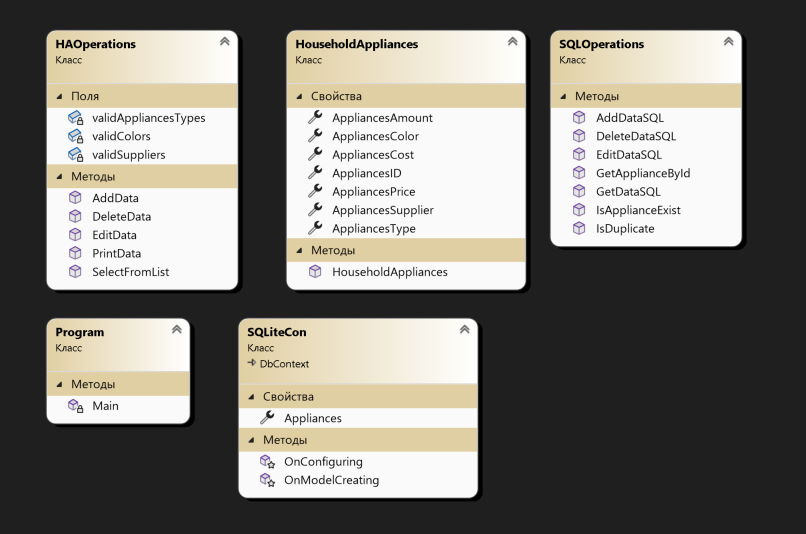


Рисунок 2.1 – UML діаграма програми

У моєму застосунку 5 класів. Ось детальний опис кожного з них:

а) файл HAOperations.cs: Цей файл містить клас HAOperations, який відповідає за виконання операцій з побутовою технікою. Клас містить методи для виведення, додавання, видалення та редагування даних про побутову техніку.

Методи PrintData(), AddData(), DeleteData(), EditData() викликаються з об'єкта HAOperations з класу Program.cs. Ці методи взаємодіють з базою даних та надають користувачеві можливість перегляду, додавання, видалення та редагування даних про побутову техніку.

Клас також містить ряд внутрішніх словників, які використовуються для перевірки валідності типів побутової техніки, кольорів та виробників, а також метод SelectFromList(), який допомагає вибрати варіанти зі списку.

б) у файлі Program.cs знаходиться головний виконуваний клас Program, який містить метод Main, що є точкою входу в програму. В цьому методі встановлюється кодування консолі, створюється об'єкт для доступу до бази даних, ініціалізується об'єкт HAOperations, а потім виконується головний цикл програми.

в) файл SQLiteCon.cs (Див. [Додаток А](#Додаток_Б)) містить клас SQLiteCon, який відповідає за з'єднання з базою даних SQLite де зберігаються дані про побутову техніку [1]. Використовується підходящий для Entity Framework Core, цей клас наслідує DbContext [2]. У методі OnConfiguring встановлюється рядок підключення до бази даних SQLite. Крім того, у методі OnModelCreating визначається структура таблиці та індекси для полів.

г) в файлі SQLOperations.cs реалізовано клас SQLOperations, який відповідає за виконання операцій з базою даних у контексті побутової техніки. Цей клас містить методи для роботи з базою даних SQLite, такі як перевірка наявності дублікатів, перевірка наявності побутового приладу за його ID, отримання побутового приладу за його ID, отримання всіх даних з бази даних, додавання нових даних, видалення даних та редагування даних [3].

Цей клас є важливим компонентом програми, оскільки він забезпечує взаємодію з базою даних та забезпечує основні операції з даними про побутову техніку. Він використовує об'єкт контексту бази даних SQLiteCon, який дозволяє взаємодіяти з базою даних SQLite за допомогою Entity Framework Core [4].

ґ) файл HouseholdAppliances.cs (Див. [Додаток А](#Додаток_В)): У цьому файлі розміщений клас HouseholdAppliances, який представляє об'єкти побутової техніки. Цей клас має наступні властивості:

AppliancesID: Ідентифікатор побутового приладу.

AppliancesType: Тип побутового приладу.

AppliancesSupplier: Виробник побутового приладу.

AppliancesPrice: Ціна побутового приладу.

AppliancesColor: Колір побутового приладу.

AppliancesCost: Загальна вартість побутового приладу (розраховується як ціна \* кількість).

AppliancesAmount: Кількість побутового приладу.

Крім того, в класі HouseholdAppliances присутній конструктор, який ініціалізує властивості об'єкту при створенні нового екземпляру класу. Цей клас використовується для моделювання та збереження даних про побутові прилади.

Кожен з цих класів відповідає за певний аспект функціоналу програми та допомагає у виконанні операцій з побутовою технікою та зберіганні цих даних у базі даних.

2.2 Тестування програмної системи

Тестовий приклад: №1

Призначення: перевірка що програма створює таблицю даних та успішно записує дані побутових приладів в певному форматі

Тест-вимоги, що перевіряється: функціональна вимога №1

Передумови для тесту: програма повинна бути запущена, на диску D комп’ютера не повинно бути вже створеної бази даних Test.bd

Критерії проходження тесту: реальна поведінка програмної системи збігається з очікуванням

Таблиця 2.1 – Тестовий приклад №1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № з/п | Крок сценарію | Очікуваний результат | Отриманий результат | Відмітка про проходження кроку сценарію (Так/Ні) |
| 1. | Оберіть опцію "Записати в таблицю" у меню програми. | Якщо файл .db відсутній, програма автоматично створить його і запросить ввести дані про перший продукт. | Програма автоматично створила файл .db і запитала вас про дані для першого продукту. | Так |
| 2. | Введіть дані про перший товар у програму. | Після введення вами даних через консоль, програма занесе ці дані в базу даних. Після завершення  операції введення, з'явиться повідомлення про успішне внесення даних до бази даних. | Операція внесення всіх даних в базу даних успішно завершена, і  з'явилось повідомлення про це. | Так |

Відмітка про проходження тесту (Пройдений/ не пройдений): пройдений

Тестовий приклад: №2

Призначення: перевірка того, що програма виводить дані побутової техніки та є можливість змінювати дані в базі даних.

Тест-вимоги, що перевіряються: функціональна вимога №2, функціональна вимога №4, функціональна вимога №6.

Передумови для тесту: програмна система повинна бути запущена, на диску обов’язково повинен бути .db файл.

Критерії проходження тесту: реальна поведінка програмної системи збігається з очікуванням

Таблиця 2.2 – Тестовий приклад №2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № з/п | Крок сценарію | Очікуваний  результат | Отриманий  результат | Відмітка про проходження кроку сценарію (Так/Ні) |
| 1 | 2 | 3 | 4 | 5 |
| 1. | Оберіть опцію "1. Вивести дані про побутову  техніку" у меню програми. | Виведеться таблиця з даними про побутову техніку. | Програма вивела таблицю з даними про побутову техніку. | Так |
| 2. | Оберіть опцію "4. Редагувати дані про побутову техніку" у меню програми. | Програма запропонує ввести ID товару в якому користувач хочете змінити якісь поля. | Програма запропонувала ввести ID товару в якому користувач хоче щось змінити. | Так |
| 3. | Введіть ID товару в якому треба змінити якісь поля. | Програма виведе данні про цей товар за його ID та запропонує обрати поле яке користувач хочете змінити. | Програма вивела данні про цей товар за його ID та запропонувала обрати поле яке користувач хоче змінити. | Так |
| 4. | Оберіть будь яку опцію у меню програми,  та додайте нові дані. | Програма оновить базу даних змінивши поле яке обрав користувач. | Програма оновила базу даних змінивши поле  яке обрав користувач. | Так |

Відмітка про проходження тесті (Пройдений/ не пройдений): пройдений

Тестовий приклад: №3

Призначення: перевірка того, що програма може видаляти дані про товари побутової технік.

Тест-вимоги, що перевіряється: функціональна вимога №5

Передумови для тесту: програмна система повинна бути запущена, на диску комп’ютера має знаходитись файл бази даних SQLite з внесеними в нього даними у визначеному форматі

Критерії проходження тесту: реальна поведінка програмної системи збігається з очікуванням

Таблиця 2.3 – Тестовий приклад №3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № з/п | Крок сценарію | Очікуваний  результат | Отриманий  результат | Відмітка про проходження кроку сценарію (Так/Ні) |
| 1 | 2 | 3 | 4 | 5 |
| 1. | Оберіть опцію "3. Видалити дані про  побутову техніку" у меню програми. | Програма запропонує ввести  ID товару який треба видалити. | Програма запропонувала ввести ID товару  який треба видалити. | Так |
| 2. | Введіть ID товару для його видалення. | Після введення вами коректного та існуючого ID через консоль, програма видалить дані про цей товар. Після завершення операції  з'явиться повідомлення про успішне видалення. | Товар з ID який увів користувач видалено, і з'явилось повідомлення про це. | Так |

Відмітка про проходження тесті (Пройдений/ не пройдений): пройдений

Тестовий приклад: №4

Призначення: перевірка того, що програма обробляє винятки, і видає користувачу попереджувальні повідомлення

Тест-вимоги, що перевіряється: функціональна вимога №8

Передумови для тесту: програмна система повинна бути запущена, на диску комп’ютера має знаходитись файл з внесеними в нього даними у визначеному форматі

Критерії проходження тесту: реальна поведінка програмної системи збігається з очікуванням

Таблиця 2.4 – Тестовий приклад №4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № з/п | Крок сценарію | Очікуваний  результат | Отриманий  результат | Відмітка про проходження кроку сценарію (Так/Ні) |
| 1 | 2 | 3 | 4 | 5 |
| 1. | Оберіть опцію "3. Видалити дані про побутову техніку" у меню програми. | Програма повідомить про успішність операції та виведе в консоль "Введіть ID товару,  який ви хочете видалити:" | Програма запропонувала ввести ID товару який треба видалити. | Так |
| 2. | Ввести індекс товару якого Немає в створеній базі даних | Програма повідомить про помилку в консолі. Програма повернеться до головного меню | Програма повідомила про помилку в консолі. Програма повернулася до головного меню | Так |

Продовження таблиці 2.4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 3. | Ввести індекс товару буквами | Програма повідомить про не правильний формат введення в консолі. Програма повернеться до  головного меню | Програма повідомила про не правильний формат введення в консолі. Програма  повернулась до головного меню | Так |
| 4. | Введіть коректний та існуючий ID товару для його видалення. | Після введення вами коректного та існуючого ID через консоль, програма видалить дані про цей товар. Після завершення операції з'явиться повідомлення про успішне видалення. | Товар з ID який увів користувач видалено, і з'явилось повідомлення про це. | Так |

Відмітка про проходження тесті (Пройдений/ не пройдений): пройдений

Тестових прикладів виконано: 4

Тестових прикладів пройдено: 4

2.3 Розгортання програмного продукту

Вимоги до апаратних засобів:

а) процесор – двох ядерний, 2 ГГц або більш швидкий;

б) доступний простір ОЗУ – не менше 4 Гб;

в) вільний дисковий простір – не менше 1 Гб.

Вимоги до програмних засобів:

а) операційна система;

1) Windows 10, 11;

2) більшість сучасних дистрибутивів Linux;

3) Max OS x 10.14 або більш нова.

б) .NET Runtime 6.0 або більш нове.

в) Entity Framework 8.4 або більш нове

Розгортання програмного продукту на комп’ютері користувача у вигляді автономного застосунку:

а) відкрити Setup\_Course\_Work\_Spring2024\_Maksakov.msi файл для інсталювання застосунку (рис 2.2);

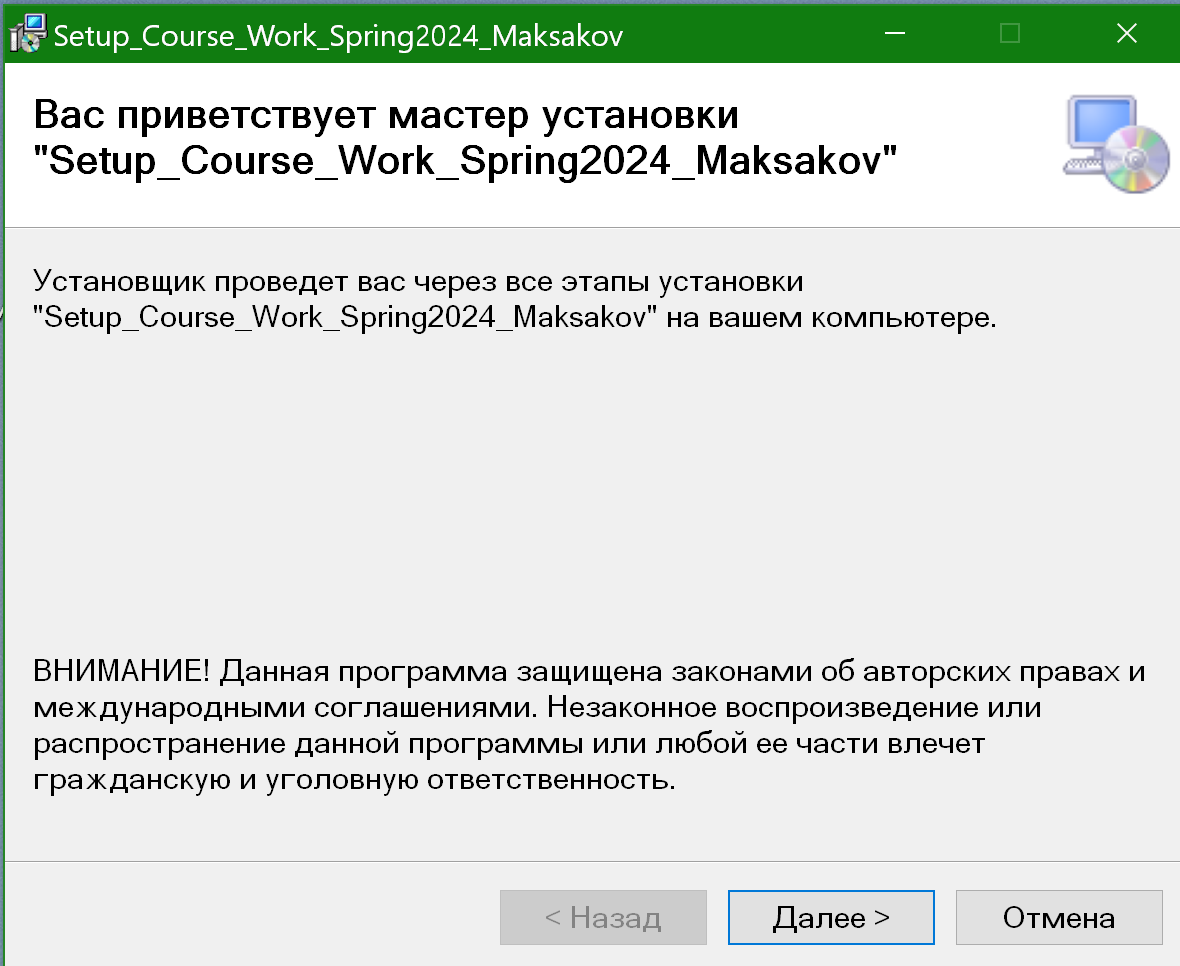


Рисунок 2.2 – Меню Setup\_Course\_Work\_Spring2024\_Maksakov.msi

б) оберіть шлях для інсталювання застосунку та створіть там порожню папку (рис. 2.3);

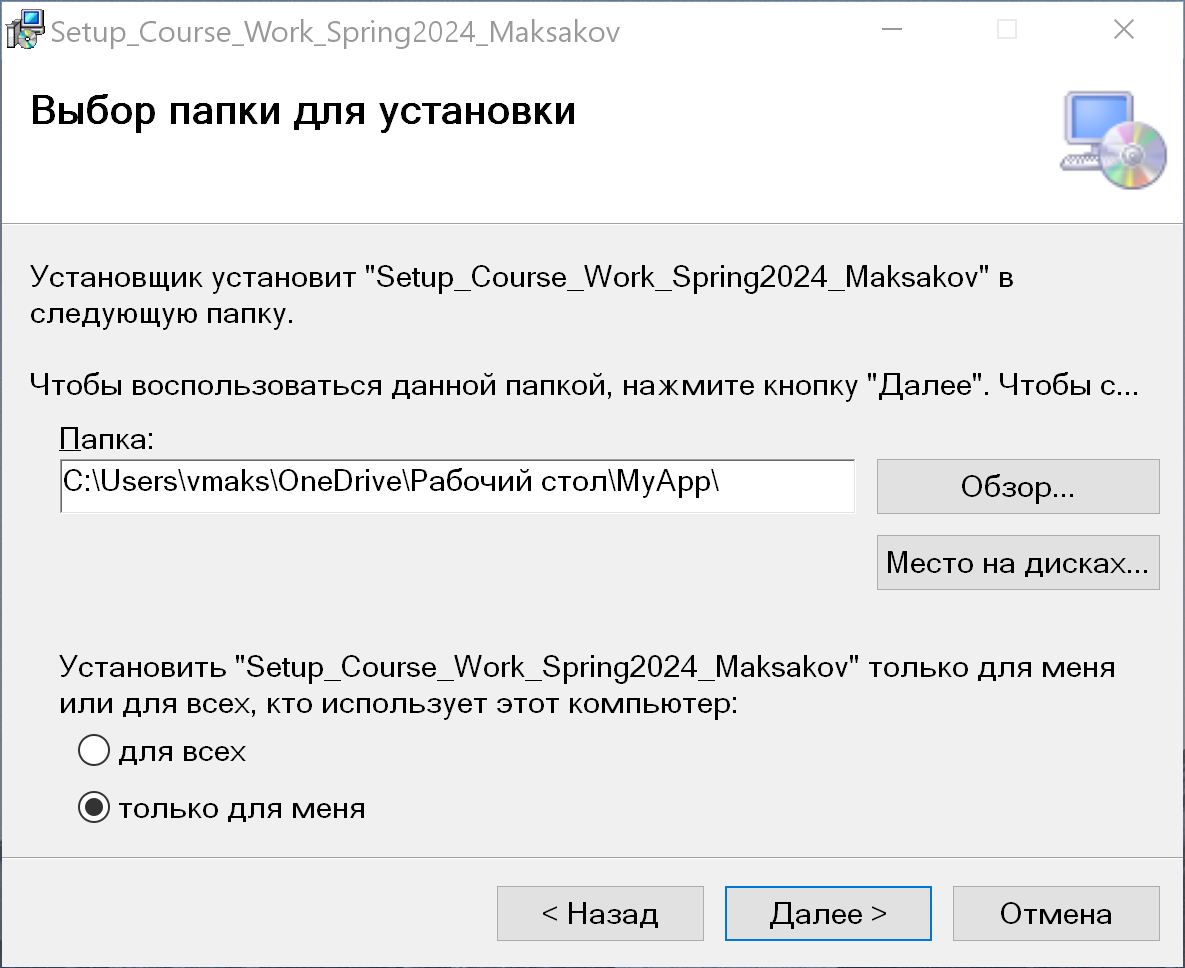


Рисунок 2.3 – Меню вибору папки для інсталювання

в) натисніть кнопку «Далее» для інсталювання (рис. 2.4);

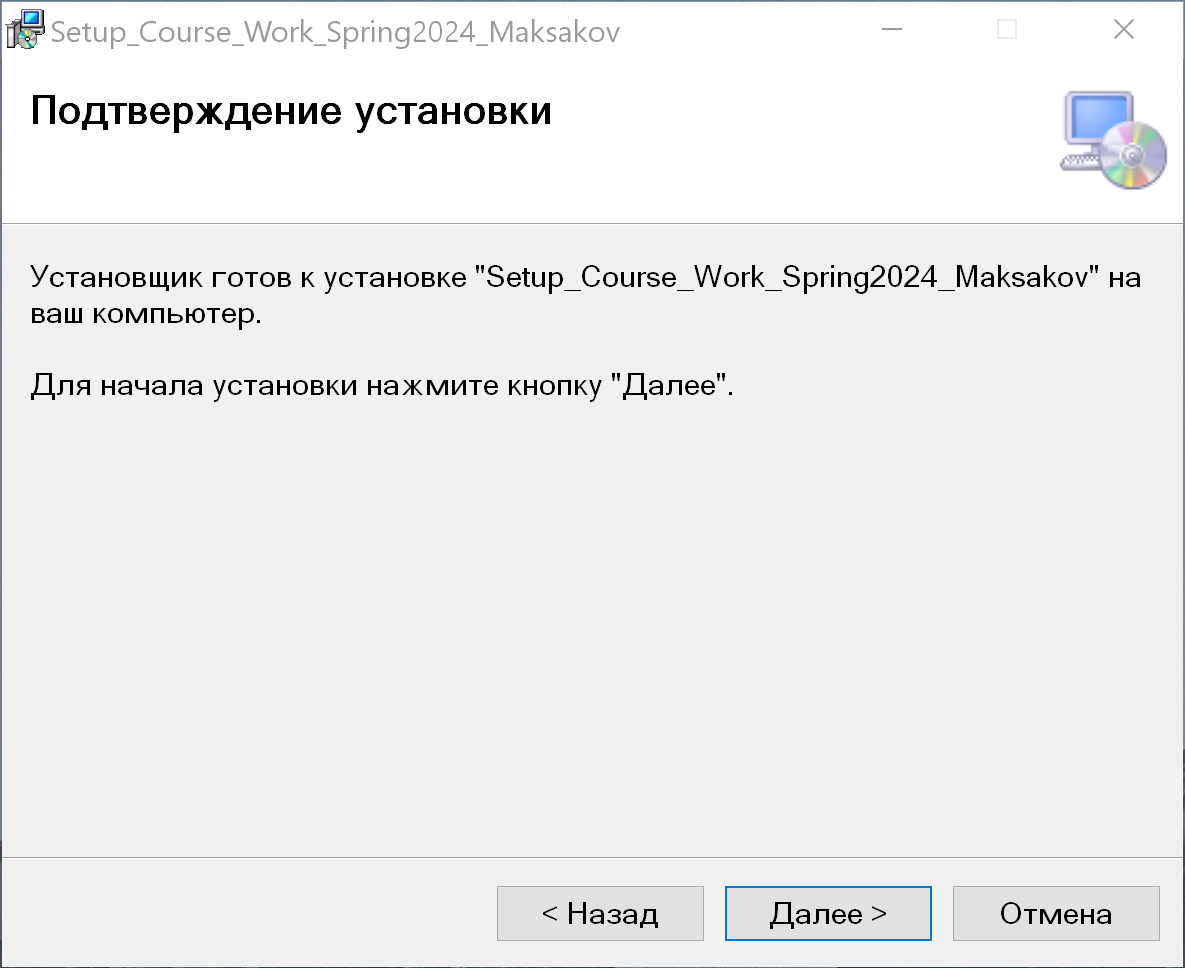


Рисунок 2.4 – Меню для підтвердження інсталювання

г) дочекайтеся кінця інсталювання (рис. 2.5);

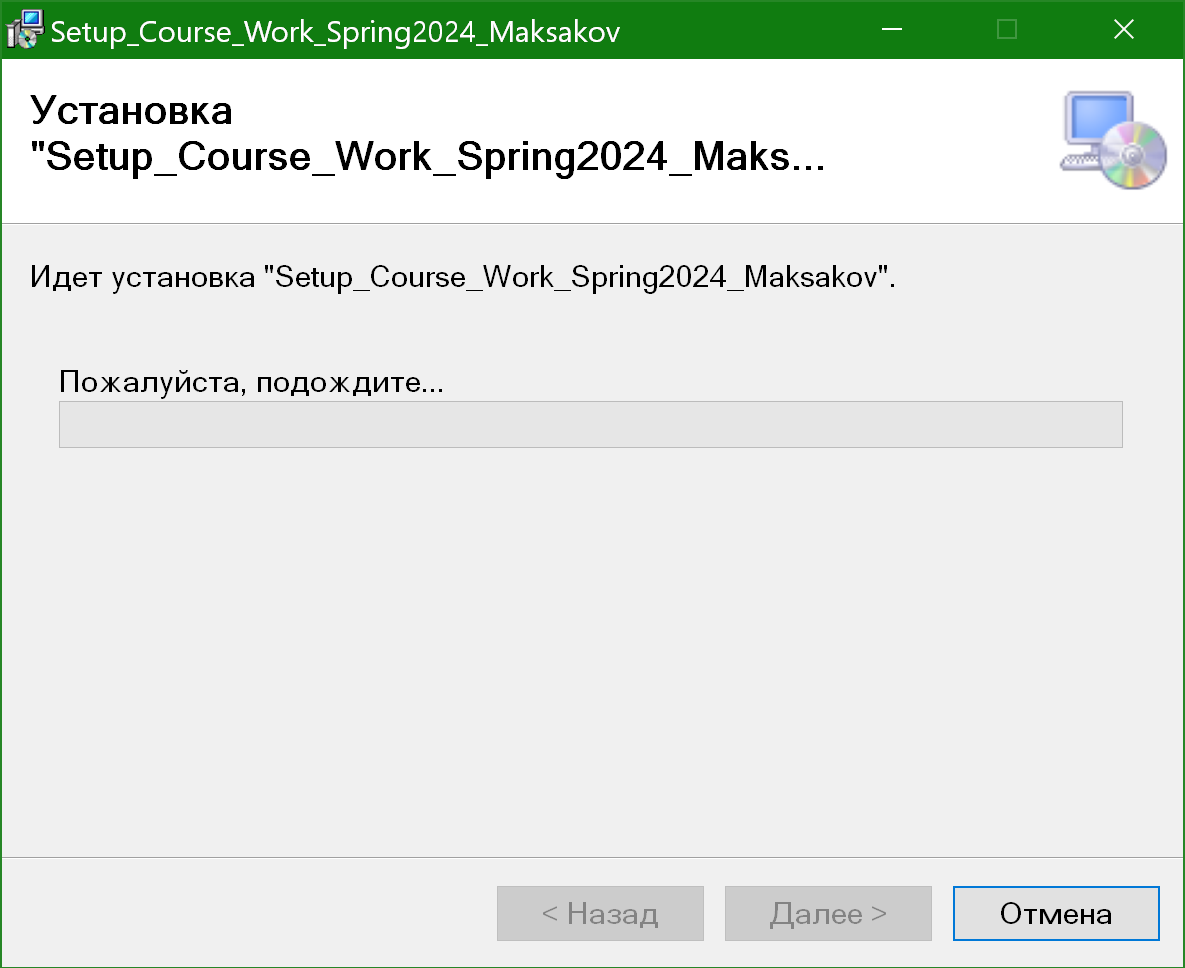


Рисунок 2.5 – Процес інсталювання

ґ) інсталювання завершено, натисніть кнопку «Закрыть»(рис. 2.6)

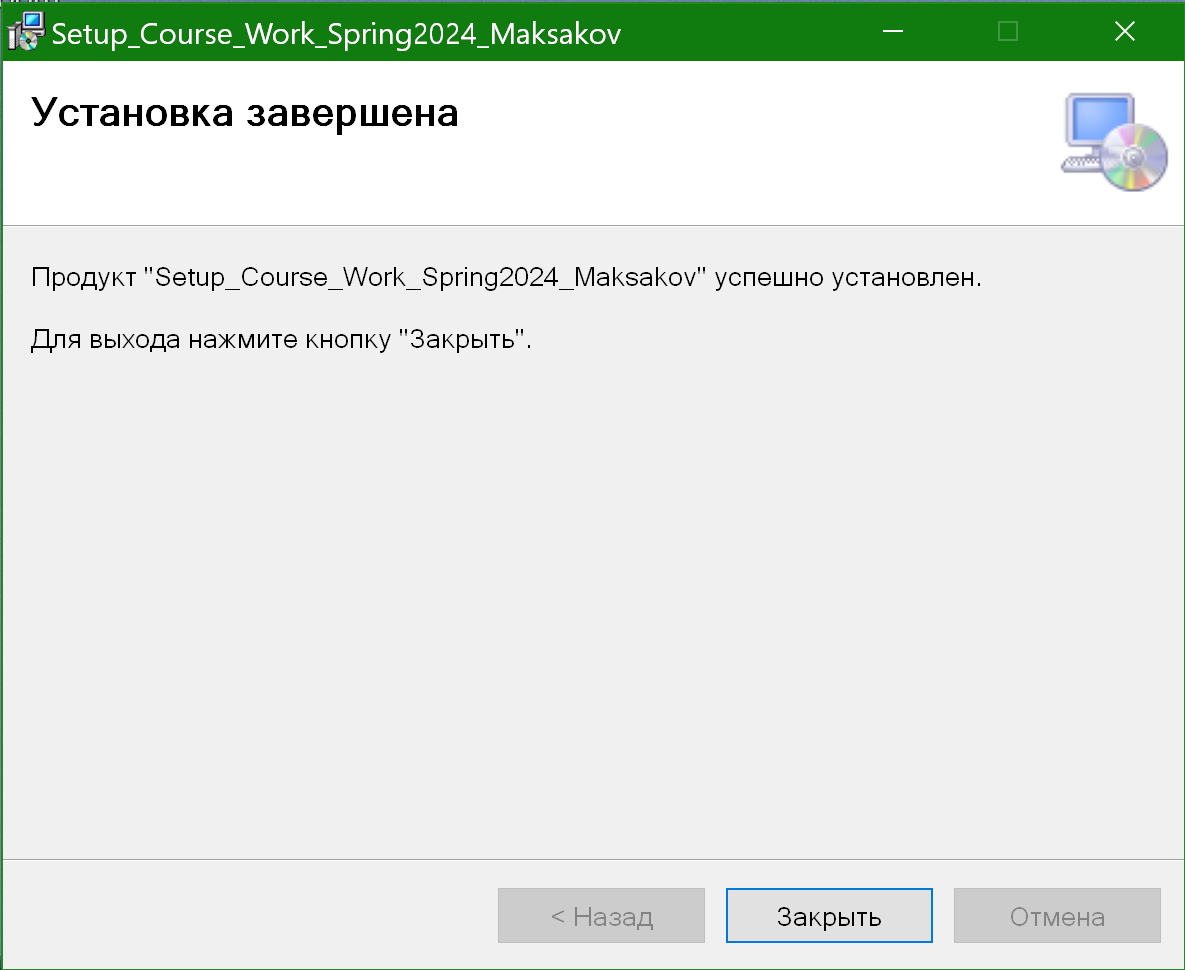


Рисунок 2.6 – Процес інсталювання завершено

2.4 Керівництво користувача

2.4.1 Призначення програмного продукту

Програмний продукт призначений для автоматизації обліку товарів магазину побутової техніки та аналізу даних для майбутньої закупівлі товару. Він дозволяє ввести облік за такими параметрами кожного типу товару: тип, виробник, колір, кількість, ціна; додавати дані про новий товар, обчислювати суму ціни всього привозу товару та зберігати дані про товар.

2.4.2 Використання програмного продукту

Запуск програми

Запуск програми в операційній системі сімейства Windows здійснюється одним з стандартним способів:

а) Подвійно клацніть лівою клавішею миші по ярлику програми MyCourseWorkSpring2024;

б) зайдіть в папку створену на робочому столі після інсталювання програми та подвійно клацніть лівою клавішею миші по MyCourseWorkSpring2024.exe;

в) Натисненням кнопки «Пуск» панелі завдань з подальшим вибором пункту «Усі програми» та подвійно клацніть лівою клавішею миші по ярлику програми.

Основні елементи користувацького інтерфейсу програми

Користувацький інтерфейс програми складається з головного меню (рис. 2.7), яке з’являєтеся після її запуску на виконання

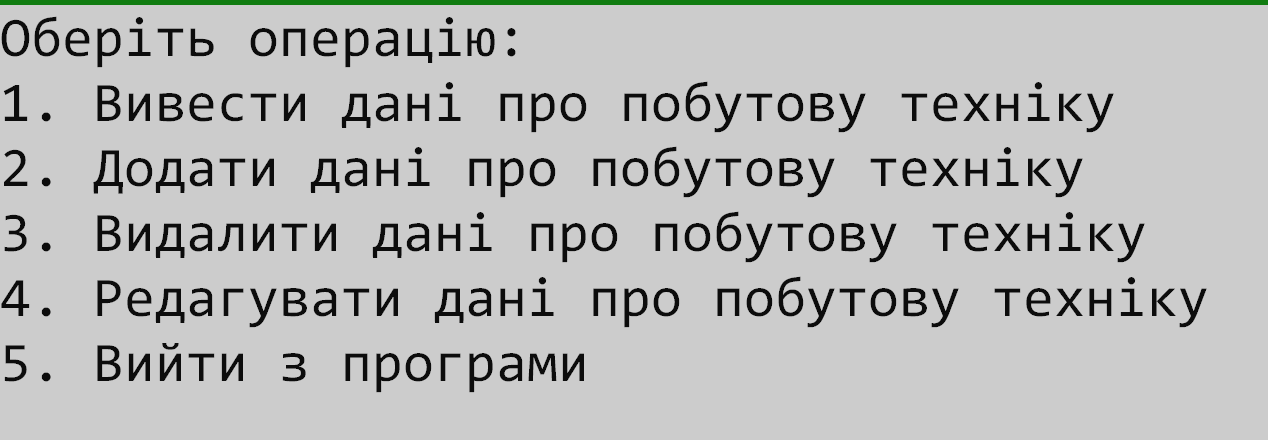


Рисунок 2.7 – Головне меню програми

Це меню містить усі команди для корегування її виконавцем:

1. «1. Вивести дані про побутову техніку» – призначена для виведення таблиці бази даних в консольну програму
2. «2. Додати дані про побутову техніку» – призначена для створення нового товару з такими характеристиками як: тип товару «Type», виробник товару «Supplier», колір товару «Color», кількість товару «Amount», ціна товару «Price», та запису її в таблицю бази даних
3. «3. Видалити дані про побутову техніку» – Видаляє одну строчку з таблиці бази даних по ID. Після виконання цієї операції, відповідний запис буде вилучено.
4. «4. Редагувати дані про побутову техніку» – призначена для редагування даних по ID. Після виконання цієї операції, відповідний запис буде змінено.
5. «5. Вийти з програми» – призначена для виходу з програми консольним шляхом

Робота з програмою

Введення початкових даних

Для початку роботи необхідно вибрати пункт головного меню «2. Додати дані про побутову техніку» шляхом введення цифри 2 та подальшого натискання Enter на клавіатурі. Після чого на екрані послідовно виводитимуться повідомлення, для вибору типу нового товару, його виробника та кольору а також для введення його кількості, та ціни (рис. 2.8). Після введення значення кожного з параметрів необхідно натиснути Enter.

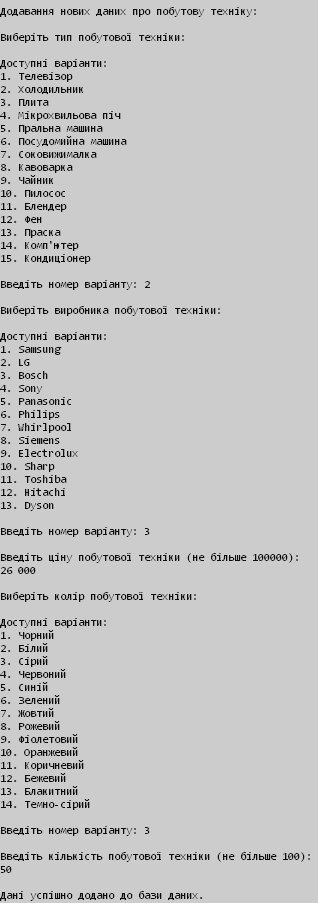


Рисунок 2.8 – Інтерфейс введення даних

Слід зазначити що у деяких параметрів, таких як «Ціна» та «Кількість», має обмеження. «Ціна» – не повинна бути менше нуля або дорівнювати нулю та перевищувати ліміт в 100 000, якщо ціна не виконує ці обмеження то програма попередить про помилку користувачеві. «Кількість» - повинна бути цілим додатнім числом яке не дорівнює нулю. «Кількість» – повинна бути додатнім числом яке не дорівнює 0 та не перевищує ліміт в 100 шт.

Виведення списку товарів в консоль

Для цього треба вибрати пункт в головному меню «1. Вивести дані про побутову техніку» шляхом введення цифри 1 та натиснувши клавішу Enter на клавіатурі.

Після виконання зазначених дій на екрані комп’ютера з’явиться список товарів (рис. 2.9).

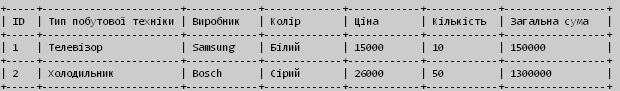


Рисунок 2.9 – Список товарів у табличній формі

Слід зазначити що програма виведе таблицю тільки в тому випадку якщо файл MyCourseWorkSpring2024.db наявний в директорії: «C:\Users\YourUsername\Desktop\MyCourseWorkFolder» або .db файл має хоч одну строчку з товаром.

Видалення даних про товар

Для цього треба вибрати пункт в головному меню «3. Видалити дані про побутову техніку» шляхом введення цифри 3 і натиснувши клавішу Enter на клавіатури.

Після виконання зазначених дій на екрані комп’ютера з’явиться запит на введення унікального номеру (ID) товару який ми хочемо (рис. 2.10) видалити після введення коректного ID товар буде успішно видалений.

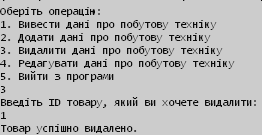


Рисунок 2.10 – Видалення даних про товар

Редагування даних

Для редагування даних треба вибрати пункт меню «4. Редагувати дані про побутову техніку» шляхом введення цифри 4 і натиснути клавішу Enter на клавіатурі

Після виконання зазначених дій на екрані комп’ютера з’явиться запит на введення унікального номеру (ID) товару який ми хочемо відредагувати, після введення коректного ID програма запропонує змінити поля, такі як «Тип», «Виробник», «Колір», «Кількість» та «Ціна». Користувач сам обирає які поля він хоче змінити, після зміненого якогось поля, програма показує відредаговані дані. Після введення коректних даних програма підтвердить ввід (рис. 2.11) та поверниться до головного меню.



Рисунок 2.11 – Редагування одного з товарів

Завершення роботи з програми

Для цього необхідні вибрати пункт меню «5. Вийти з програми» введення цифри 5 і натиснути клавішу Enter.

2.4.3 Повідомлення користувачеві при

Якщо користувач програми в головному меню ввів некоректне ЧИСЛО – номер параметру то на екрані комп’ютера з’явиться повідомлення «Невірний вибір. Будь ласка, виберіть правильний пункт з меню.» (рис. 2.12)

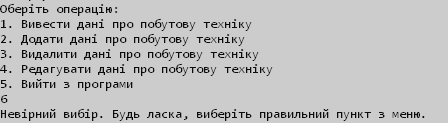


Рисунок 2.12 – Обробка помилкової ситуації 1

Якщо користувач програми в головному меню ввів символи або букви замість номеру операції то на екрані комп’ютера з’явиться повідомлення «Невірний вибір. Будь ласка, виберіть правильний пункт з меню.» (рис. 2.13)

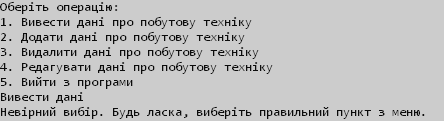


Рисунок 2.13 – Обробка помилкової ситуації 2

Якщо користувач програми в головному меню нічого не ввів та натиснув Enter замість номеру операції то на екрані комп’ютера з’явиться повідомлення «Невірний вибір. Будь ласка, виберіть правильний пункт з меню.» (рис. 2.14)

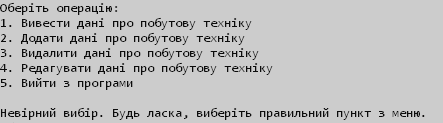


Рисунок 2.14 – Обробка помилкової ситуації 3

Якщо користувач програми вибрав пункт «1. Вивести дані про побутову техніку», «3. Видалити дані про побутову техніку» або «4. Редагувати дані про побутову техніку» в головному меню, але в директорії «C:\Users\YourUsername\Desktop\MyCourseWorkFolder» немає файлу MyCourseWorkSpring2024.db, або в базу даних не було внесено жодного товару на екрані комп’ютера з’явиться повідомлення «У базі даних немає даних про побутову техніку.» (рис. 2.15)

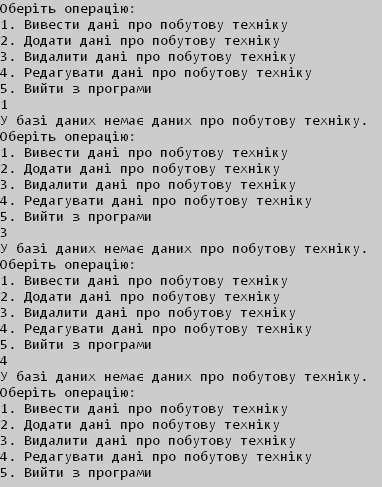


Рисунок 2.15 *–* Обробка помилкової ситуації 4, 5, 6

Якщо користувач програми обрав пункт «2. Додати дані про побутову техніку» або «4. Редагувати дані про побутову техніку» і при виборі «Типу» побутової техніки введе не коректні дані, виведеться повідомлення (рис. 2.16).

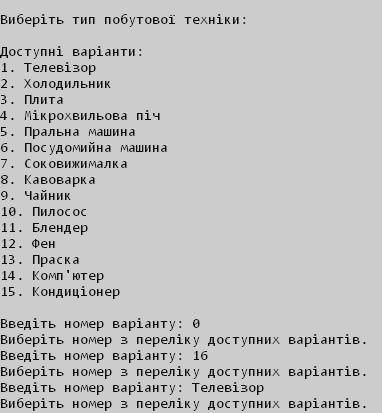


Рисунок 2.16 – Обробка помилкової ситуації 7, 8, 9

Якщо користувач програми обрав пункт «2. Додати дані про побутову техніку» або «4. Редагувати дані про побутову техніку» і при виборі «Виробника» побутової техніки введе не коректні дані, виведеться повідомлення (рис. 2.17).

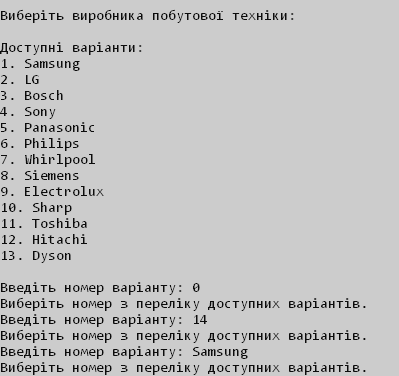


Рисунок 2.17 – Обробка помилкової ситуації 10, 11, 12

Якщо користувач програми вибрав пункт «2. Додати дані про побутову техніку» або «4. Редагувати дані про побутову техніку» і при введені даних в строку «Введіть ціну побутової техніки (не більше 100000):» ввів дробове, від’ємне число, букви або символи, нуль або більше 100 000 то програма попередить користувача повідомленням і зробить повторний запит на ціну товару (рис. 2.18).

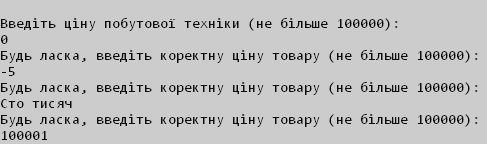


Рисунок 2.18 – Обробка помилкової ситуації 13, 14, 15, 16

Якщо користувач програми обрав пункт «2. Додати дані про побутову техніку» або «4. Редагувати дані про побутову техніку» і при виборі «Колір» побутової техніки введе не коректні дані, виведеться повідомлення (рис. 2.19).

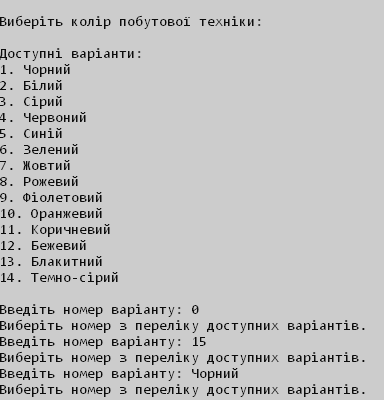


Рисунок 2.19 – Обробка помилкової ситуації 17, 18, 19

Якщо користувач програми вибрав пункт «2. Додати дані про побутову техніку» або «4. Редагувати дані про побутову техніку» і при введені даних в строку «Введіть кількість побутової техніки (не більше 100):» ввів дробове, від’ємне число, букви або символи, нуль або більше 100 то програма попередить користувача повідомленням і зробить повторний запит на кількість товару (рис. 2.20).

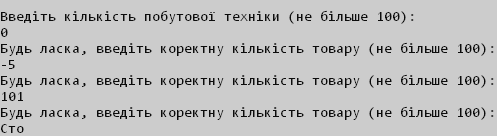


Рисунок 2.20 – Обробка помилкової ситуації 20, 21, 22, 23

Якщо користувач програми обрав пункт «3. Видалити дані про побутову техніку» або «4. Редагувати дані про побутову техніку» і при введені даних в строчку «Введіть ID товару, який ви хочете видалити:» або «Введіть ID товару, який ви хочете змінити:» ввів некоректний ID який не відноситься до жодного товару то програма попередить користувача повідомленням «Товар з таким ID не знайдено.» (рис. 21), та поверне користувача в головне меню.

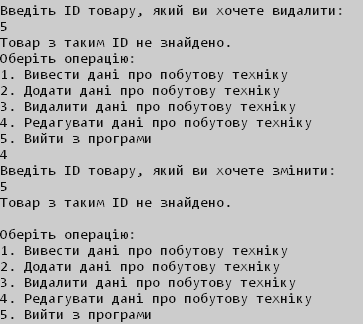


Рисунок 2.21 – Обробка помилкової ситуації 24, 25

ВИСНОВКИ

За результатами цієї роботи було розроблено консольну програму на мові програмування C#. Протягом процесу розробки було здійснено глибокий аналіз функціоналу програми, ретельно розглянуто її структуру та класи.

Було проведено живе тестування програми, презентуючи кожен її крок користувачеві, щоб переконатися в її коректному функціонуванні. Крім того, було детально проаналізовано всі можливі сценарії помилок та надано розгорнутий опис математичної формули, що використовується у програмі.

Під час виконання цього проекту вдосконалено навички програмування, особливо у написанні структурованих програм, а також розширено знання про класи та ознайомлено з різними Framework для роботи з базами даних. Детально ознайомлено з документацією Entity Framework та SQLite, що значно розширило знання.

Ця робота над проектом вдосконалила сконцентрованість та наполегливість. Вона також надихнула на бажання подальшого розвитку та вдосконалення програмного коду для досягнення ще більшої продуктивності та ефективності програми.

У майбутньому можливі напрямки модернізації включають впровадження міграцій, адаптацію під сучасний інтерфейс користувача та розширення функціоналу. Наприклад, можливе введення функції пошуку продукту за різними параметрами та створення додаткових баз даних та зв'язків між ними.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. YouTube. "Міні курс по Entity Framework." Отримано з: <https://www.youtube.com/watch?v=SryQxUeChMc&list=PLdo4fOcmZ0oX7uTkjYwvCJDG2qhcSzwZ6>

2. Entity Framework Core CLI Отримано з: <https://learn.microsoft.com/en-us/ef/core/cli/>

3. Microsoft. "Робота з ключами в Entity Framework Core." Отримано з: <https://learn.microsoft.com/en-us/ef/core/modeling/keys?tabs=data-annotations>

4. Entity Framework Core Queries Отримано з: <https://learn.microsoft.com/en-us/ef/core/querying/>

5. C# документація Отримано з: <https://learn.microsoft.com/ru-ru/dotnet/csharp/>

Додаток А

using Microsoft.EntityFrameworkCore;

namespace MyCourseWorkSpring2024

{

public class SQLiteCon : DbContext

{

public DbSet<HouseholdAppliances> Appliances { get; set; }

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder){

string desktopPath = Environment.GetFolderPath(Environment.SpecialFolder.Desktop);

string folderPath = Path.Combine(desktopPath, "MyCourseWorkFolder");

if (!Directory.Exists(folderPath))

{

Directory.CreateDirectory(folderPath);

}

string dbPath = Path.Combine(folderPath, "MyCourseWorkSpring2024.db");

optionsBuilder.UseSqlite($"Filename={dbPath}");

base.OnConfiguring(optionsBuilder);

}

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

modelBuilder.Entity<HouseholdAppliances>().ToTable("Tovar", "test");

modelBuilder.Entity<HouseholdAppliances>(entity =>

{

entity.HasKey(k => k.AppliancesID);

entity.HasIndex(i => i.AppliancesID);

});

base.OnModelCreating(modelBuilder);

}

}

}

Продовження додатку А

namespace MyCourseWorkSpring2024

{

public class HouseholdAppliances

{

public int AppliancesID { get; set; }

public string AppliancesType { get; set; }

public string AppliancesSupplier { get; set; }

public float AppliancesPrice { get; set; }

public string AppliancesColor { get; set; }

public float AppliancesCost { get; set; }

public int AppliancesAmount { get; set; }

public HouseholdAppliances(string appliancesType, string appliancesSupplier, float appliancesPrice,

string appliancesColor, int appliancesAmount)

{

AppliancesType = appliancesType;

AppliancesSupplier = appliancesSupplier;

AppliancesPrice = appliancesPrice;

AppliancesColor = appliancesColor;

AppliancesAmount = appliancesAmount;

AppliancesCost = appliancesPrice \* appliancesAmount;

}

}

}