Міністерство освіти і науки України

Чернівецький національний університет імені Юрія Федьковича

\_\_\_\_\_\_\_Програмного забезпечення комп’ютерних систем\_\_\_\_\_\_\_

(повна назва кафедри, циклової комісії)

**КУРСОВИЙ ПРОЄКТ**

з Бази даних та NoSQL системи

(назва дисципліни)

на тему: **Інформаційна система аеропорту** **засобами мови С#**

Виконав: студент (ка)     3     курсу, групи     343А

спеціальності           121

(шифр спеціальності)

\_\_\_\_Інженерія програмного забезпечення\_\_\_\_\_

(назва спеціальності)

\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_Матюхін О.К.\_\_\_\_

(підпис) (прізвище, ініціали)

Керівник \_\_\_\_\_\_\_Комісарчук В.В.\_\_\_\_\_\_

(прізвище, ініціали)

|  |  |  |
| --- | --- | --- |
|  |  | **Дата захисту** «\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_р. |
|  |  | **Оцінка:** |
|  |  | за національною шкалою \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  | (словами) |
|  |  | кількість балів \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  | (цифра) |
|  |  | за шкалою ECTS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  | (літера) |

Чернівці, 2024



**РЕФЕРАТ**

В рамках курсового проєкту реалізований програмний додаток для керування базою даних «Інформаційна система аеропорту» для настільних та переносних пристроїв на базі операційної системи Windows.

Застосунок призначений для використання персоналом аеропорту без спеціальної підготовки в сфері інформаційних технологій.

Область застосування – програмне забезпечення для ведення обліку персоналу аеропорту, літаків, квитків та розкладу рейсів на базі Windows-пристроїв (ноутбуків або персональних комп'ютерів.

Програмний продукт створено за допомогою інтегрованого середовища розробки Visual Studio на мові C# з використанням графічної підсистеми WPF, яке дозволяє швидко та якісно створювати додатки в операційній системі Windows, та NoSQL бази даних MongoDB, що дозволяє зручно оброблювати велику кількість даних .

Архітектура та структура програмного продукту дає можливість у майбутньому розширити додаток із додаванням нового функціоналу та видозміненою логіки обробки.

Курсовий проект містить:   74 с.,   34   рис.,   5   табл.,   2   додатка,

  26   джерел.

*ОПЕРАЦІЙНА СИСТЕМА, WPF, КОМП’ЮТЕРИ, MongoDB, VISUALSTUDIO, C#.*

**SUMMARY**

Within course project implemnted program aplication for control of Data Base "Music goods" for poartable and desktop devices based on Windows System.

Application made for using by regular user without expirience of work and special thing in the Infomational Technology area.

Use area is software for personal using and small business in selling music content based on Windows-devices (laptop and Pesonal Computers) with limited counting recources.

Program product developed with help of integrated development environment Visual Studio in C# with using graphical sybsystem Windows Forms, wich allow fast and quality develop applications in operational system Windows.

Architecture and structure of program application allow in future expand application with adding new functional and modified processing logic.

The course project contains:   74   p.,   34  fig.,   5   table.,   2   appendice,

  26   sources

OPERATING SYSTEM, *Windows Forms*, COMPUTERS, VISUAL STUDIO, C#.

**ЗМІСТ**

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ………………………………………..…6

1. АРХІТЕКТУРНА РЕАЛІЗАЦІЯ ТА ФУНКЦІОНАЛ ПРОГРАМИ .……7
   1. Загальні вимоги до програми……………………………………………7
   2. Призначення та область застосування……………………………….…9
   3. Структурно-функціональний аналіз предметної області
   4. Виокремлення сутностей об’єктів структурного аналізу предметної області
   5. Функціональні вимоги…………………………………………..……… 9
2. СПЕЦИФІКАЦІЯ ПРОГРАМНОГО ДОДАТКУ…………………………11
   1. Структура програми……………………………………………………..11
   2. Опис методів програми………………………………………………….13
   3. Засоби розробки.………………………………………………...………21
   4. Опис користувацького інтерфейсу………………………………..……22

ВИСНОВКИ……………………………………………………………...………36

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ………………………………..………37

ДОДАТКИ………………………………………………………………..………39

Додаток А. Алгоритми роботи програми…………………………………...39

Додаток Б. Скролінг (текст) програми………………………………….….49

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**

ОС – операційна система.

GUI (Graphical User Interface) – графічний інтерфейс користувача.

БД – база даних.

MessageBox – модальне вікно з текстом для користувача, що блокує інші дії в додатку, доки користувач не закриє його. Окрім тексту може містити кнопки та символи для покращення взаємодії з користувачем.

Microsoft Visual Studio – серія продуктів фірми Майкрософт, які містять інтегроване середовище розробки програмного забезпечення та низку інших інструментальних засобів.

Windows Forms – інтерфейс програмування програм (API), який є частиною програмної платформи Microsoft .NET Framework.

MongoDB – документоорієнтована система управління базами даних, яка не вимагає опису схеми таблиць. Вважається одним із класичних прикладів NoSQL-систем, використовує JSON-подібні документи та схему бази даних.

Windows Presentation Foundation (WPF, кодова назва — Avalon) —платформа інтерфейсу користувача, яка надає комплексний набір функцій розробки додатків, які включають мову XAML, елементи управління, прив'язку до даних, макет, двовимірну і тривимірну графіку, анімацію, стилі, шаблони, документи, мультимедіа, текст і типографічні функції.

# АРХІТЕКТУРНА РЕАЛІЗАЦІЯ ТА ФУНКЦІОНАЛ ПРОГРАМИ

## **Загальні вимоги до програми**

### **Вимоги до графічного інтерфейсу користувача**

1. Обов’язковий україномовний інтерфейс.
2. Використання зрозумілої користувачеві термінології для елементів інтерфейсу.
3. Обов’язкова наявність панелі меню з функціональністю, редагування, додавання та видалення даних.
4. Використання додатково панелі графічних елементів для управління БД.
5. Наявність спливаючої інформації для графічних елементів управління.
6. Забезпечення функціонування «Посібник користувача» з інформацією про використання програми.
7. Обов’язкова наявність спливаючих діалогових вікон в разі некоректного вводу даних.
8. Обов’язкова наявність серед колірних тем наявність теми із загальноприйнятою колірною гаммою.

### **Вимоги до архітектури програми**

1. Реалізація NoSQL бази даних.
2. Забезпечення стійкості програми:
   1. програма має бути працездатною при будь-яких діях користувача, дані або їх цілісність не повинні втрачатись;
   2. програма повинна здійснювати перевірку коректності введення-виведення даних та обробку помилок.

### **Структурно-функціональний аналіз предметної області**

Аналізуючи умови завдання курсового проекту та структурну схему служб аеропорту, було розроблено власну ієрархію служб

аеропорту. Головними у ієрархії схеми є департаменти, див додаток A.1, кожному департаменту підпорядковується служби.

Кожна служба має

власний спектр спеціалістів з яких в свою чергу формуються команди для повноцінної роботи аеропорту.

Першим департамент - департамент

### **Вимоги до функціональності додатка**

1. Реалізація збереження даних у NoSQL базі даних MongoDB;
2. Реалізувати можливість змін та зчитування даних з бази користувачем безпосередньо у додатку, залежно від прав.
3. Реалізувати можливість створення нового користувача у
4. Реалізація подачі вихідної інформації користувачем на GUI шляхом введення даних до полів таблиці.
5. Виконання стандартних операцій обробки елементів БД (обробка даних та управління БД).
6. Відображення результатів обробки даних в вигляді таблиці головного вікна програми.
7. Виконання операцій фільтрації, сортування та пошуку елементів БД за різними ключами, обраними користувачем, з відображенням результатів в графічному інтерфейсі.
8. Перевірка коректності введення даних, що передаються користувачем через графічний інтерфейс програмного додатку.
9. Інформування користувача за допомогою інформаційних та діалогових вікон під час роботи користувача з БД програмного додатку.

### **Вимоги до вихідного коду**

1. Вихідний код кожного з класів програми повинен міститися в окремому файлі.
2. Наявність коментарів (для класів – призначення класу; для методів – призначення методу, опис параметрів та значення, що повертається).

## **Призначення та область застосування**

Функціональним призначеннямєвикористання програмного додатку для зручного адміністрування бази даних «Музичні товари» (управління та перегляд зі внесенням змін до списку товарів: додавання, редагування, видалення, фільтрацію та сортування).

Областю застосування є експлуатація програмного додатку приватними особами та підприємцями у сфері малого бізнесу з метою покращення торгівлі музичним контентом на базі Windows-пристроїв (ноутбуків або персональних комп'ютерів) з обмеженими обчислювальними ресурсами.

## **Функціональні вимоги**

1. Створення файлу БД.
2. Вхідні дані подаються користувачем на GUI.
3. Результат редагування та обробки БД подаються в табличному вигляді.
4. Текстовий файл бази даних має оброблятись шляхом:
   1. збереження даних у файл;
   2. зчитування даних з файлу;
5. Меню та панель управління повинні забезпечувати наступні можливості:
6. оновлення даних таблиці відображення елементів БД;
7. редагування даних БД;
8. додавання нового рядка до таблиці елементів БД;
9. додавання даних до файлу БД;
10. видалення даних з БД;
11. Додатково меню повинно забезпечувати можливість перегляду посібника користувача у віконному режимі.
12. Сортування інформації, що відображується в графічному інтерфейсі користувача, за зростанням та спаданням:
13. за номером в каталозі;
14. за виконавцем;
15. за ціною;
16. за кількістю пісень;
17. Фільтрація записів, що відображаються в графічному інтерфейсі користувача, за критерієм ціни.
18. Пошук записів музичних товарів, що відображаються в графічному інтерфейсі користувача, за різними критеріями:
19. за назвою;
20. за виконавцем;
21. за номером в каталозі;
22. за назвою пісні;
23. Отримання та відображення підсумкової інформації в результаті фільтрації, сортування та пошуку в таблиці головного вікна програми.
24. Забезпечення перевірки допустимості даних, що вводяться користувачем.
25. Видача користувачу попереджуючих та інформаційних повідомлень.

# СПЕЦИФІКАЦІЯ ПРОГРАМНОГО ДОДАТКУ

## **Структура програми**

### **Модулі програми**

### BlackThemeMainView – клас який реалізує введення даних користувачем у елементи управління для виконання операцій та відображення результатів у таблиці DataGridView.

BlackThemeModifData – клас, який відображує довідкову інформацію, щодо внесення змін до БД (видалення, додавання, редагування) через таблицю головного вікна програми.

BlackThemeAboutAutor – клас який відображує довідкову інформацію, про автора в темній колірній гамі.

BlackThemePanelElments – клас, який відображує довідкову інформацію, щодо роботи елементів панелі управління.

BlackThemeSeletionData – клас, який відображує довідкову інформацію, щодо відбору даних в таблиці головного вікна програми (сортування, фільтрація, пошук).

IMainView – інтерфейс для зв’язку класів View (BlackThemMainView та LightThemeMainView) з класом MainWindowPresentor.

MainPresentor – клас, який реалізує логічні операції програмного додатку та виводить результати в головне вікно програми, завдяки підписці на події та отриманню параметрів View за допомогою інтерфейсу IMainView.

LightThemeMainView – клас, який наслідується від BlackThemeMainView та має видозмінену світлу кольорову гаму, без зміни логіки роботи.

LightThemeModifData – клас, який наслідується від BlackThemeModifData та має видозмінену світлу кольорову гаму.

LightThemePanelElements – клас, який наслідується від BlackThemePanelElements та має видозмінену світлу кольорову гаму.

LightThemeSeletionData – клас, який наслідується від BlackThemeSeletionData та має видозмінену світлу кольорову гаму.

Albom – клас, який реалізує прототип альбому, який має властивості: назва, виконавець перелік пісень, кількість пісень.

MemStorage – клас, який наслідується від класу Albom та реалізує прототип носію інформації, який має властивості: тип носія, час звучання.

Product – клас, який наслідується від класу MemStorage та реалізує прототип товару, який має властивості: каталоговий номер, ціна.

ListBase – клас, що містить всередині список, в який зчитуються дані з текстового файлу під час виконання програми, та записує й зберігає дані в текстовий файл, при внесенні змін в список, реалізуючи модель БД.

ListTable – клас, що функціонально копіює поле-список з класу ListBase, для реалізації операцій фільтрації, сортування та пошуку, редагування й додавання елементів таблиці головного вікна, без збереження змін у записи текстового файлу.

### **Алгоритми роботи програми**

Загальний алгоритм роботи основної програми та підпрограм описані у вигляді блок-схем в Додатку.А:

1. Рисунок А.1 – Загальна блок-схема алгоритму роботи програми.
2. Рисунок А.2 – Узагальнена блок-схема алгоритму сортування даних БД.
3. Додаток А.3 – Узагальнена блок-схема алгоритму фільтрації даних БД.
4. Додаток А.4 – Узагальнена блок-схема алгоритму пошуку записів БД.
5. Додаток А.5 – Узагальнена блок-схема алгоритму зміни вмісту комірки таблиці головного вікна.
6. Додаток А.6 – Узагальнена блок-схема додавання пустого рядка в таблицю.
7. Додаток А.7 – Узагальнена блок-схема додавання запису в БД.
8. Додаток А.8 – Узагальнена блок-схема збереження редагування запису в БД.
9. Додаток А.9 – Узагальнена блок-схема видалення елементу з БД.
10. Додаток А.10 – Узагальнена блок-схема роботи меню головного вікна.

## **Опис методів програми**

Список методів класу BlackThemeMainView та їх опис наведено в табл.1.

Таблиця 1 - Основні методи класу BlackThemeMainView

|  |  |  |
| --- | --- | --- |
| № | Назва методу | Опис методу |
| 1 | 2 | 3 |
| 1 | AssociateAndRaiseViewEvents() | Здійснює виклик подій головного вікна та їх прив’язку до подій, що передаватимуться у MainWindowPresentor.. |
| 2 | dataGridView1\_CellBeginEdit(object sender,DataGridViewCellCancelEventArgs e) | Здійснює запам’ятовування даних комірки на початку редагування, для використання перевірки даних комірки у функції класу MainWindowPresentor. |
| 3 | ChooseSearchButton\_Click (object sender, EventArgs e) | Здійснює переключення меню обробки даних на вкладку «Пошук». |

Продвження табл.1 - Основні методи класу BlackThemeMainView

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 4 | СhooseSortButton\_Click(object sender, EventArgs e) | Здійснює переключення меню обробки даних на вкладку «Сортування». |
| 5 | ChooseFiltrButton\_Click(object sender, EventArgs e) | Здійснює відкриття інформаційного вікна «Фільтрація» при натисненні на відповідний підпункт, пункту панелі меню «Про програму». |
| 6 | SelectionToolStripMenuItem\_Click\_1(object sender, EventArgs e) | Здійснює відкриття інформаційного вікна «Відбір даних» при натисненні на відповідний підпункт, пункту панелі меню «Про програму». |
| 7 | ModifyDataToolStripMenuItem\_Click\_1  (object sender, EventArgs e) | Здійснює відкриття інформаційного вікна «Зміна даних» при натисненні на відповідний підпункт, пункту панелі меню «Про програму». |
| 8 | ElementPanelToolStripMenuItem\_Click\_1  (object sender, EventArgs e) | Здійснює відкриття інформаційного вікна «Елементи панелі меню» при натисненні на відповідний підпункт, пункту панелі меню «Про програму». |
| 9 | AboutDeveloperToolStripMenuItem\_Click (object sender, EventArgs e) | Здійснює відкриття інформаційного вікна «Про розробника» при натисненні на відповідний пункт панелі меню. |
| 10 | LightThemToolStripMenuItem\_Click(object sender, EventArgs e) | Здійснює заміну темної теми на світлу при натисненні на відповідний підпункт пункту панелі меню «Зміна теми». |

Продвження табл.1 - Основні методи класу BlackThemeMainView

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 11 | RestartToolStripMenuItem\_Click(object sender, EventArgs e) | Здійснює виклик події натиснення кнопки «Оновити» при натисненні на відповідний підпункт пункту панелі меню «Інструменти». |
| 12 | AddElToolStripMenuItem\_Click(object sender, EventArgs e) | Здійснює виклик події натиснення кнопки «Додати новий елемент в базу» при натисненні на відповідний підпункт пункту панелі меню «Інструменти». |
| 13 | RedactToolStripMenuItem(object sender, EventArgs e) | Здійснює виклик події натиснення кнопки «Зберегти редагування в базу» при натисненні на відповідний підпункт пункту панелі меню «Інструменти». |
| 14 | RemoveToolStripMenuItem1\_Click(object sender, EventArgs e) | Здійснює виклик події натиснення кнопки «Видалити» при натисненні на відповідний підпункт пункту панелі меню «Інструменти». |
| 15 | AddRowInTableToolStripMenuItem\_Click  (object sender, EventArgs e) | Здійснює виклик події натиснення кнопки «Додати рядок в таблицю» при натисненні на відповідний підпункт пункту панелі меню «Інструменти» . |
| 16 | SearchingElment\_KeyDown(object sender, KeyEventArgs e) | Прибирає фокус з поля пошуку при натисненні кнопки «Enter». |

Продвження табл.1 - Основні методи класу BlackThemeMainView

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 17 | FiltrationByPriceFrom\_KeyDown(object sender, KeyEventArgs e) | Прибирає фокус з поля верхньої границі фільтрації при натисненні кнопки «Enter». |
| 18 | FiltrationByPriceTo\_KeyDown(object sender, KeyEventArgs e) | Прибирає фокус з поля нижньої границі фільтрації при натисненні кнопки «Enter». |

Список методів інтерфейсу IMainView та їх опис наведено в табл.2.

Таблиця 2 – Основні методи інтерфейсу IMainView

|  |  |  |
| --- | --- | --- |
| № | Назва методу | Опис методу |
| 1 | 2 | 3 |
| 1 | SetTableSoure(BindingSource TABELELIST) | Здійсню виклик подій головного вікна та їх прив’язку до подій, що передаватимуться у MainWindowPresentor.. |
| 2 | IfDataInTableSelected() | Здійснює запам’ятовування даних комірки на початку редагування, для використання перевірки даних комірки у функції MainWindowPresentor. |
| 3 | GetCatalogNumberOFSelectedElement() | Здійснює передачу значення каталогового номера виділеного елемента у MainWindowPresentor. |
| 4 | GetFiltrationByPriceFrom() | Здійснює передачу значення верхньої границі фільтрації у MainWindowPresentor. |
| 5 | GetFiltratiц33 onByPriceTo() | Здійснює передачу значення верхньої границі фільтрації у MainWindowPresentor. |
| 6 | GetEditingСolumnIndex() | Здійснює передачу в MainWindowPresentor номеру стовпчика змінюваної комірки. |

Продовження табл.2 – Основні методи інтерфейсу IMainView

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 7 | GetPreviousValue() | Здійснює передачу у MainWindowPresentor значення комірки перед зміною. |
| 8 | GetChangedValue() | Здійснює передачу у MainWindowPresentor значення комірки таблиці після редагування. |
| 9 | GetSelectedRowIndex() | Здійснює передачу у MainWindowPresentor номера виділеного рядка. |
| 10 | GetSearchingElment() | Здійснює передачу у MainWindowPresentor текст поля пошуку. |
| 11 | GetSortDirection() | Здійснює передачу у MainWindowPresentor напряму сортування. |
| 12 | GetTypeOfSorting() | Здійснює передачу у MainWindowPresentor типу сортування. |
| 13 | CancelChangesInTableCell() | Здійснює скасування змін у комірці таблиці. |
| 14 | СlearAllFields() | Здійснює очищення у MainWindow всіх полів обробки даних. |
| 15 | СlearSortFields() | Здійснює очищення у MainWindow всіх полів обробки даних. |
| 16 | ClearFiltraionFields() | Здійснює очищення у MainWindow поля фільтрації. |

Список методів класу MainPresentor та їх опис наведено в табл.3.

Таблиця 3 – Основні методи класу MainPresentor

|  |  |  |
| --- | --- | --- |
| № | Назва методу | Опис методу |
| 1 | 2 | 3 |
| 1 | ReloadViewTable() | Здійснює відміну всіх незбережених змін у таблиці. |
| 2 | ReloadAllInformation() | Здійснює відміну всіх незбережених змін у таблиці та очищує всі поля обробки даних. |
| 3 | AddNewElementToTable() | Здійснює додавання нового рядка в таблицю. |
| 4 | ControlDataInCell() | Здійснює перевірку даних таблиці та скасовує зміни при введенні некоректних значень. |
| 5 | ErrorInData() | Здійснює вивід помилки при вводі некоректних типів даних у комірки таблиці головного вікна. |
| 6 | RedactElementInBase() | Здійснює збереження змін рядка таблиці головного вікна у базу даних. |
| 7 | ReloadDataInTable() | Здійснює перезавантаження даних у таблицю головного вікна. |
| 8 | ChangingBySearchLine() | Здійсню оновлення таблиці та скидання полів сортування та фільтрації головного вікна при повному стиранні значення пошукового поля. |
| 9 | Sort() | Здійсню сортування даних таблиці головного вікна. |
| 10 | RemoveElement() | Здійснює видалення елементу з бази даних. |
| 11 | AddElementInBase() | Здійснює додавання нового елементу у базу даних. |
| 12 | CancelChangesInTableCell() | Здійснює скасування змін у комірці таблиці. |

Список методів класу ListTable та їх опис наведено в табл.4.

Таблиця 4 – Основні методи класу ListTable

|  |  |  |
| --- | --- | --- |
| № | Назва методу | Опис методу |
| 1 | 2 | 3 |
| 1 | RewriteDataInViewsByDataFromDatabase  (ListBase ChangedBase) | Здійснює перезапис даних списку-копії даними з БД. |
| 2 | TestIfSelectedElementNotNull(int NumberOfRowSelectElement) | Здійснює перевірку чи заповнений виділений рядок таблиці. |
| 3 | IfElementChanged(Product TestingElement, ListBase DatabaseData) | Здійснює перевірку чи даний елемент таблиці змінено. |
| 4 | IfThereNonUnsavedChangesInTable(ListBase DatabaseData) | Здійснює перевірку чи немає незбережених змін елементів таблиці. |
| 5 | IfThereNonUnsavedElementInTable(ListBase DatabaseData) | Здійснює перевірку чи немає нових незбережених елементів таблиці. |
| 6 | SortingByCatalogNumber(ref List<Product> ListBeSorted,string Direction) | Здійснює сортування списку за каталоговим номером |
| 7 | SortingByPrice(ref List<Product> ListBeSorted, string Direction) | Здійснює сортування списку за ціною. |
| 8 | SortingByNumberOfSongs(ref List<Product> ListBeSorted, string Direction) | Здійснює сортування списку за кількістю пісень. |
| 9 | SortingByGroupName(ref List<Product> ListBeSorted, string Direction | Здійсню сортування даних таблиці головного вікна за назвою групи. |
| 10 | SearchByName(string Name, ListBase DataFromBase) | Здійснює пошук в базі даних за назвою носія. |
| 11 | SearchByCatalogNumber(string CatalogNumber, ListBase DataFromBase) | Здійснює пошук в базі даних за каталоговим номером. |
| 12 | SearchBySongName(string SongName, ListBase DataFromBase) | Здійснює пошук в базі даних за назвою пісні. |

Продовження табл.4 – Основні методи класу ListTable

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 13 | SearchByGroupName(string Name, ListBase DataFromBase) | Здійснює пошук в базі даних за назвою групи. |
| 14 | public bool TestCountOfSongs(Product ProductItem) | Здійснює перевірку чи відповідності кількості пісень, до реальної. |
| 15 | TestIFCatalogNumberExist(int CatatlogNumber, ListBase DataFromBase) | Здійснює перевірку чи запис з даним каталоговим номером є в базі даних. |

Список методів класу ListBase та їх опис наведено в табл.5.

Таблиця 5 – Основні методи класу ListBase

|  |  |  |
| --- | --- | --- |
| № | Назва методу | Опис методу |
| 1 | 2 |  |
| 1 | RemoveElementFromBase(int DeleteCatalogNumber) | Здійснює видалення елементу з БД за каталоговим номером. |
| 2 | WriteData( List<Product> storage) | Здійснює запис зі списку даних БД у текстовий файл. |
| 3 | WriteToFIle(string PathToFile, List<List<string>> list) | Здійснює запис списку рядків у текстовий файл бази даних і є допоміжним для методу WriteData. |
| 4 | IFElementExistInBase(int CatalogNumberOfElmenet) | Здійснює перевірку чи товар з даним каталоговим номером існує в БД. |
| 5 | ReadData() | Здійснює зчитування даних з файлу БД в список даних БД. |

Продовження табл.5 – Основні методи класу ListBase

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 7 | public void FormatingData(Product ProductItem) | Здійснює форматування даних перед збереженням у текстовий файл. |
| 6 | ReadInformationFromDataFile(string pathtofile) | Здійснює зчитування даних з текстового файлу у список рядків та є допоміжним для методу ReadData. |

## **Засоби розробики**

Програмний додаток створено на мові С# за допомогою середовища розробки програмних продуктів Visual Studio. Ряд продуктів компанії Microsoft під торговою маркою Microsoft Visual Studio являє собою потужний засіб, який містить у собі інтегроване середовище розробки (IDE) програмного забезпечення та низку інших корисних інструментальних засобів.

Ігри та програми з графічним інтерфейсом (також з підтримкою технології Windows Forms, UWP), веб-сайти, веб-додатки, веб-служби в коді всіх платформ, підтримуваних Windows, Windows Mobile, Windows CE, .NET Framework, .NET Core, .NET, MAUI та інших – все це можливо реалізувати за допомогою вищевказаної лінійки продуктів.

IDE Visual Studio об’єднує можливість написання, редагування, налагодження та складання коду, а також подальше розгортання програми. Крім того Visual Studio включає компілятори, засоби завершення коду, систему керування версіями, розширення та багато інших функцій для покращення кожного етапу розробки програмного забезпечення. Зокрема редактор вихідного коду, технологію IntelliSense, налагоджувач, редактор форм, веб-редактор, дизайнер класів значно спрощують створення програмного продукту.

Таким чином, Microsoft Visual Studio є зручним для виконання всього циклу розробки різноманітного ПЗ в одному місці для різноманітних платформ Windows.

## **Опис користувацького інтерфейсу**

Головне вікно, яке відкривається після запуску програми, є основним інтерфейсом для взаємодії користувача з БД, і містить наступні елементи (див. рис.2.1):

* панель меню (позначено цифрою 1);
* панель з графічними елементами управління (позначено цифрою 2);
* вкладки функціоналу обробки даних БД (позначено цифрою 3);
* таблиця для відображення записів БД та результатів взаємодії з даними (позначено цифрою 4).

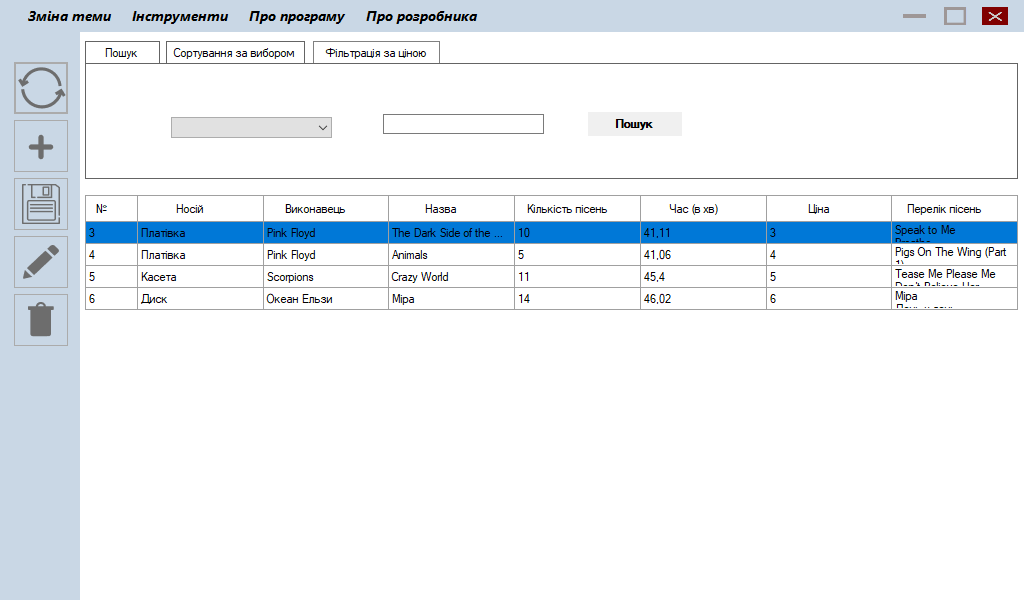


Рисунок 2.1 – Структурні елементи головного вікна

**2**

**3**

**1**

**4**

В панелі меню можна вибрати світлу та темну теми інтерфейсу програми (див. рис.2.2).



Рисунок 2.2 – Пункт меню «Зміна теми»

Пункт меню «Інструменти» (див. рис.2.3) дозволяє формувати БД (додавати, редагувати, видаляти елементи БД і т.д.). Функціонал пункту меню дублює панель з графічними елементами меню управління.

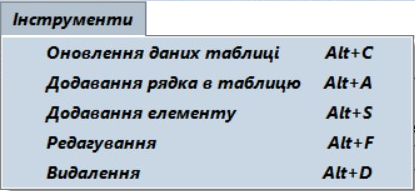


Рисунок 2.3 – Пункт меню «Інструменти»

Пункт меню «Про програму» (див. рис.2.4) містить інформацію про роботу програми (кожен підпункт містить посилання на відповідне інформаційне вікно).

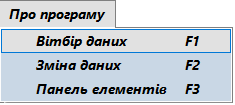


Рисунок 2.4 – Пункт меню «Про програму»

Пункт меню «Про розробника» посилається на вікно, яке містить інформацію про розробника програми.

В панелі графічних елементів управління (див. рис.2.5) знаходяться наступні функціональні кнопки:

* «Оновити» (позначено цифрою 1);
* «Додати ряд у таблицю» (позначено цифрою 2);
* «Додати до бази» (позначено цифрою 3);
* «Редагувати» (позначено цифрою 4);
* «Видалити» (позначено цифрою 5).



Рисунок 2.5 – Панель з графічними елементами управління

Кнопка «Оновити» перезавантажує записи таблиці даними з БД, використовується для відміни змін таблиці, незбережених у базу і скидання до необробленого пошуком, фільтрацією і сортуванням вигляду. При цьому виводиться за допомогою MessageBox попереджувальне повідомлення про втрату незбережених даних (див. рис.2.6).

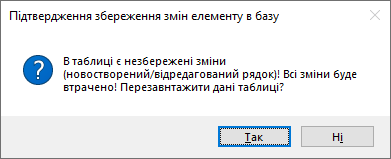


Рисунок 2.6 – Діалогове вікно до функції «Оновити»

Кнопка «Додати ряд у таблицю» додає новий незаповнений рядок у таблицю (за умови якщо немає попередньо доданого рядка, незбереженого у базу). У випадку наявності незбереженого рядка при натисканні на кнопку виводиться відповідне інформаційне повідомлення (див. рис.2.7).

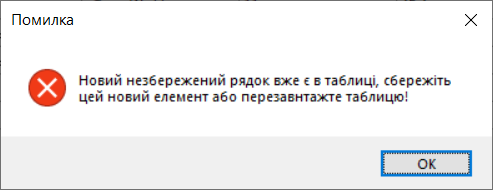


Рисунок 2.7 – Інформаційне повідомлення до функції «Додати ряд у таблицю»

Кнопка «Додати до бази» дозволяє зберегти новий рядок у базу. Після натискання кнопки виводиться за допомогою MessageBox діалогове вікно для підтвердження додавання у базу (див. рис.2.8), в якому треба натиснути кнопку «Так» для збереження даних у базу.

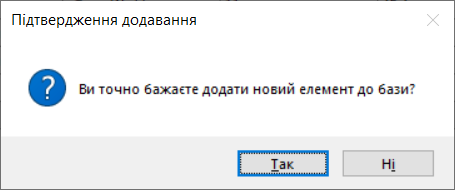


Рисунок 2.8 – Діалогове вікно до функції «Додати до бази»

У випадку якщо рядок незаповнений або такий запис вже є в базі, виведуться відповідні інформаційні повідомлення за допомогою MessageBox (див. рис.2.9).

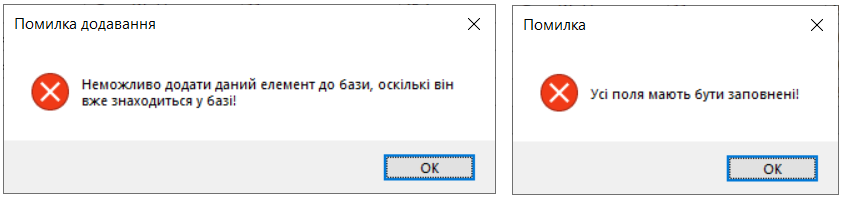


Рисунок 2.9 – Інформаційні повідомлення до функції «Додати до бази»

я

Редагування елементу бази даних здійснюється наступним чином. Спочатку вносяться зміни в обраний рядок. Для збереження внесених змін потрібно натиснути кнопку «Редагувати», після чого відбувається збереження виділеного рядка. У випадку, якщо рядок відсутній у базі, незаповнений або зміни не внесені, виведуться відповідні інформаційні повідомлення за допомогою MessageBox (див. рис.2.10).

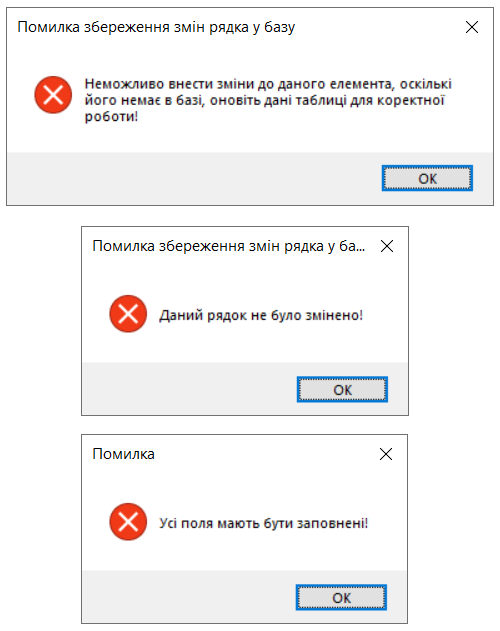


Рисунок 2.10 – Інформаційні повідомлення до функції «Редагувати»

Кнопка «Видалити» дозволяє виключити обраний в таблиці рядок з бази. Після натискання кнопки за допомогою MessageBox виводиться діалогове вікно для підтвердження виконання дії видалення (див. рис.2.11).

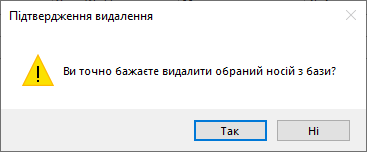


Рисунок 2.11 – Діалогове вікно до функції «Видалити»

При у мові якщо рядок доданий до таблиці, але незбережений в БД, або не має не заповнені комірки, за допомогою MessageBox виведуться відповідні інформаційні повідомлення (див. рис.2.12).

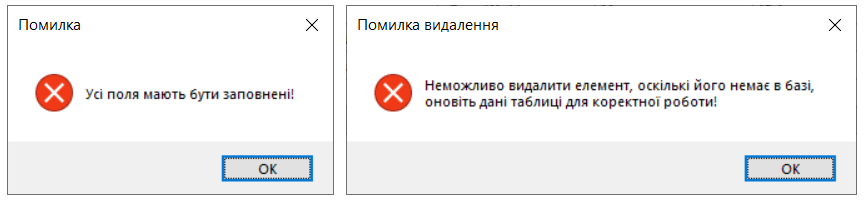


Рисунок 2.12 – Інформаційні повідомлення до функції «Видалити»

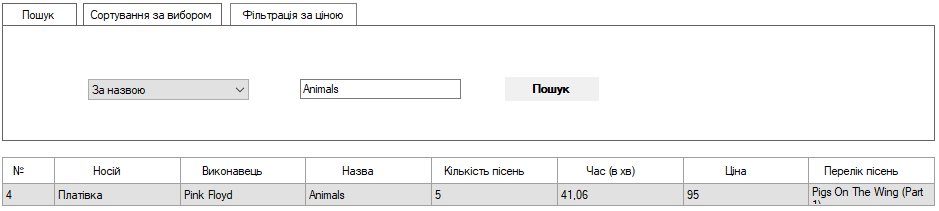
Меню функціоналу обробки БД містить наступні елементи:

* вкладка «Пошук»;
* вкладка «Сортування»;
* вкладка «Фільтрація за ціною».

Вкладка «Пошук» (див. рис.2.13) містить наступні елементи:

* поле «Тип пошуку»;
* поле «Елемент пошуку»;
* кнопка «Пошук».

Поле «Тип пошуку» (позначено цифрою 1) дозволяє обрати певні критерії, за якими буде виконуватись відбір необхідних користувачу записів БД. Поле «Елемент пошуку» (позначено цифрою 2) використовується для введення даних, які відповідають критеріям типу пошуку. Після вибору необхідного критерія та елементу пошуку, при натисканні на кнопку «Пошук» (позначено цифрою 3), здійснюється процедура відбору певних записів БД.



**1**

**2**

**3**

Рисунок 2.13 – Вкладка «Пошук»

Якщо поле «Тип пошуку» або «Елемент пошуку» не заповнені, виведуться відповідні інформаційні повідомлення (див. рис.2.14)

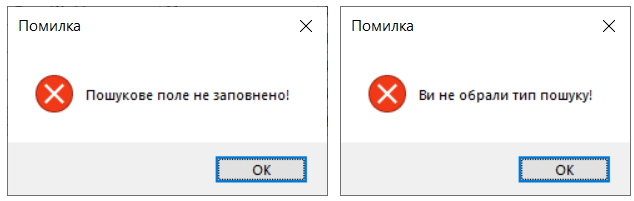
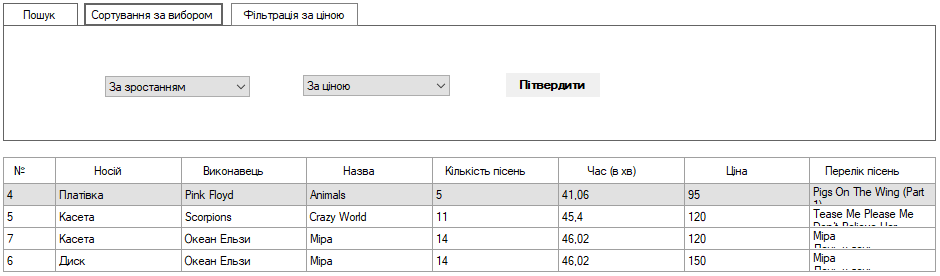


Рисунок 2.14 – Інформаційні повідомлення до функції «Пошук»

Вкладка «Сортування» (див. рис.2.15) містить наступні елементи:

* поле «Напрям сортування»;
* поле «Тип сортування»;
* кнопка «Підтвердити».

Поле «Напрям сортування» (позначено цифрою 1) використовується для сортування за зростанням або спаданням критерію сортування. Поле «Тип сортування» (позначено цифрою 2) дозволяє вибрати критерій сортування. Після натискання кнопки «Підтвердити» (позначено цифрою 3) виконується процедура сортування з виведенням результату в таблицю.



**1**

**2**

**3**

Рисунок 2.15 – Вкладка «Сортування за вибором»

У випадку некоректного заповнення полів «Напрям сортування» або «Тип сортування» або відсутності даних в одному з полів виведуться відповідні інформаційні повідомлення (див. рис.2.16).

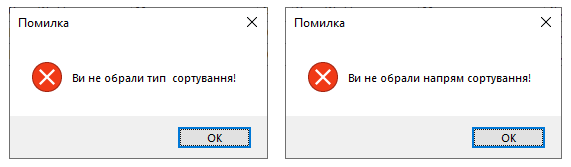
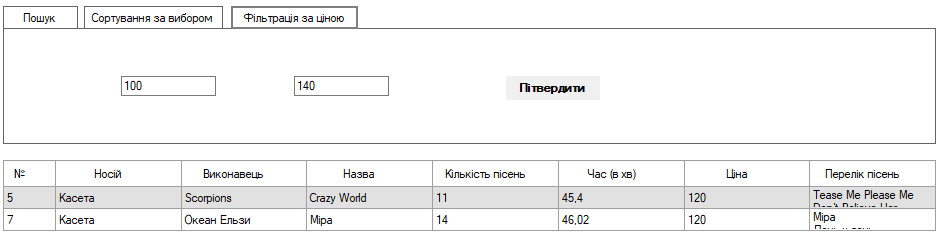


Рисунок 2.16 – Інформаційні повідомлення до функції «Сортування»

Вкладка «Фільтрація за ціною» (див. рис.2.17) містить наступні елементи:

* поле «Від»;
* поле «До»;
* кнопка «Підтвердити»

Поля «Від» (позначено цифрою 1) та «До» (позначено цифрою 2) використовуються для введення верхньої та нижньої границь фільтрації. При натисканні кнопки «Підтвердити» (позначено цифрою 3) виконується процедура фільтрації з виведенням результату в таблицю.



**1**

**2**

**3**

Рисунок 2.17 – Вкладка «Фільтрація за ціною»

У випадку некоректного заповнення полів «Від» або «До» або відсутності даних в обох полях виведуться відповідні інформаційні повідомлення (див. рис.2.18). При наявності заповненого одного з полів, процедура фільтрації виконується в будь-якому випадку.

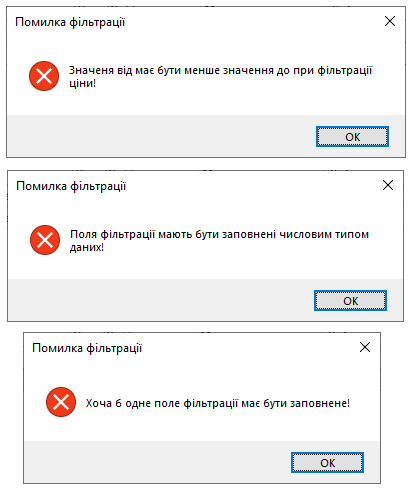


Рисунок 2.18 – Інформаційні повідомлення до функції «Фільтрація»

Таблиця (див. рис.2.19) як елемент головного вікна призначений для відображення всіх записів, результатів їх обробки (за допомогою функціоналу обробки БД) та внесення змін (за допомогою меню інструментів) до БД.

Таблиця містить наступні стовбці:

* «Каталоговий номер»;
* «Носій»;
* «Виконавець»;
* «Кількість пісень»;
* «Час»;
* Ціна;
* «Перелік пісень».



**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

Рисунок 2.19 – Таблиця в головному вікні

Для внесення змін в таблиці потрібно двічі натиснути лівою кнопкою миші на відповідну комірку обраного для змін рядка. Комірка перейде у режим редагування і користувач зможе ввести та зберегти зміни, натиснувши Enter. Зміни попередньо зберігаються тільки в таблиці без збереження в БД (для збереження в базу змін таблиці використовуються відповідні пункти меню інструментів).

При введені некоректних даних в комірку стовбця «Каталоговий номер» (позначено цифрою 1), після натиснення «Enter» виводяться відповідні інформаційні повідомлення (див. рис.2.20), в залежності від типу помилки.

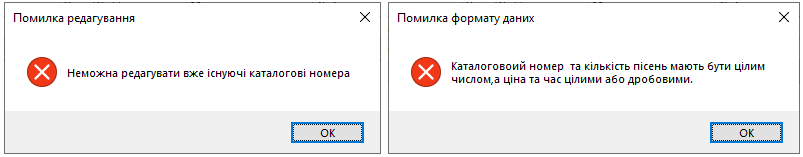


Рисунок 2.20 – Інформаційні повідомлення до комірки стовбця «Каталоговий номер»

Введення некоректних даних в комірку стовбця «Носій» (позначено цифрою 2) після натиснення «Enter» викликає повідомлення з інформацією про можливі значення комірки (див. рис.2.21).

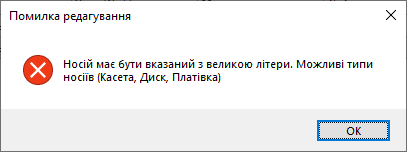


Рисунок 2.21– Інформаційне повідомлення до комірки стовбця «Носій»

Формат введених даних в комірки стовбців «Виконавець» та «Назва» (позначено цифрами 3 та 4) є довільним та не викликає повідомлень з інформацією про можливі помилки.

Комірка стовбця «Кількість пісень» (позначено цифрою 5) може містити тільки цілочисельний тип даних та викликає відповідне інформаційне вікно (див. рис.2.22) при некоректному вводі.

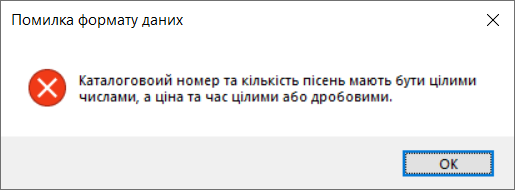


Рисунок 2.22 – Інформаційне повідомлення до комірки стовбця «Кількість пісень»

Формат введених даних в комірки стовбців «Час» та «Ціна» (позначено цифрами 6 та 7) є цифровим з довільним значенням та викликає відповідне інформаційне вікно (див. рис.2.23) при некоректному вводі.

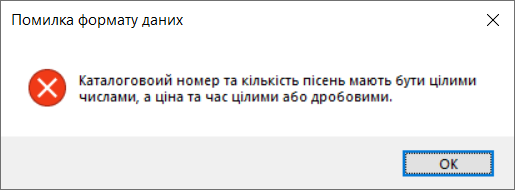


Рисунок 2.23 – Інформаційне повідомлення до комірок стовбців «Час» та «Ціна»

На відміну від попередніх стовбців, комірка стовбця «Перелік пісень» має особливий формат введення даних. Через свій великий об’єм текст не відображається в комірці в повному обсязі. Щоб одержати повну інформацію про вміст комірки достатньо навести курсор на комірку. В такому випадку з’являється спливаюче вікно з повним переліком пісень (див. рис.2.24).

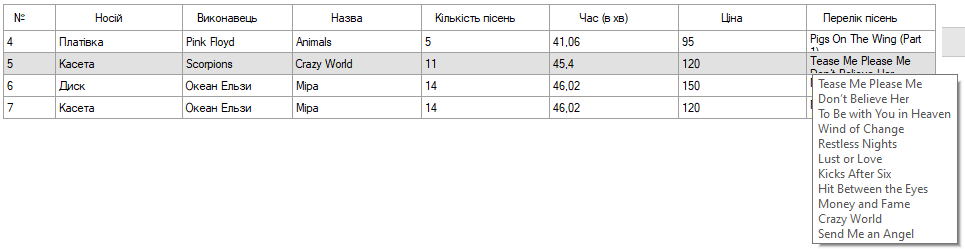


Рисунок 2.24 – Спливаюча інформація для комірки стовбця «Перелік пісень»

Дані в комірку стовбця «Перелік пісень» додаються за певною процедурою. Після введення чергової назви пісні натискання комбінації клавіш «Shift+Enter» дозволяє зробити перенесення курсору на наступний рядок без виходу з режиму редагування комірки. Якщо користувачу потрібно повернутись до попереднього рядка комірки, достатньо використати клавішу «Вгору». Для завершення редагування потрібно натиснути «Enter».

**ВИСНОВКИ**

В процесі виконання курсового проекту на тему «Створення інформаційно-довідкової системи «Музичні товари» засобами мови С#» створено програмний додаток з функціоналом керування БД з певною спеціалізацією.

Отримано цінні навички:

* роботи з середовищем розробки Microsoft Visual Studio на мові С#;
* самостійного пошуку методів розв’язку проблем в ході виконання поставлених задач;
* практичної роботи щодо процесу керування та обробки записів БД;
* виконання оформлення результатів роботи в вигляді докладного та добре аргументованого тексту пояснювальної записки;

Набутий досвід дає повноцінне уявлення про:

* можливості створення додатку на мові С# на базі Windows Forms;
* використання принципів ООП;
* формування архітектури та структури додатка;
* особливості функціоналу програмного продукту.

Додаток є повноцінним застосунком:

* із забезпеченням вимог стійкості до будь-яких можливих дій користувача;
* з архітектурою та структурою, яка дає можливість у майбутньому розширити програму шляхом додавання нового функціоналу та видозмінення логіки обробки.

Одержаний програмний продукт повністю готовий до використання пересічним споживачем без спеціальної підготовки. Застосунок призначений для особистого використання та малого бізнесу в сфері торгівлі музичним контентом. При створені додатку враховано необхідність його використання на базі Windows-пристроїв з обмеженими обчислювальними ресурсами.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

Електронні джерела:

1. <https://learn.microsoft.com/en-us/dotnet/desktop/winforms/controls/column-fill-mode-in-the-windows-forms-datagridview-control?view=netframeworkdesktop-4.8>
2. <https://learn.microsoft.com/en-us/dotnet/desktop/winforms/controls/column-fill-mode-in-the-windows-forms-datagridview-control?view=netframeworkdesktop-4.8>
3. <https://metanit.com/sharp/wpf/11.php>
4. <https://learn.microsoft.com/en-us/dotnet/desktop/winforms/controls/bindingsource-component-overview?view=netframeworkdesktop-4.8>
5. <https://learn.microsoft.com/en-us/dotnet/api/system.windows.forms.control.databindings?view=windowsdesktop-7.0>
6. <https://learn.microsoft.com/en-us/dotnet/api/system.windows.forms.datagridviewcolumn.autosizemode?view=windowsdesktop-7.0#system-windows-forms-datagridviewcolumn-autosizemode>
7. <https://metanit.com/sharp/wpf/5.14.php>
8. <https://learn.microsoft.com/en-us/dotnet/api/system.windows.forms.datagridview.rowcount?view=windowsdesktop-7.0>
9. <https://ru.stackoverflow.com/questions/1081685/c-datagridview->
10. <https://learn.microsoft.com/en-us/dotnet/api/system.windows.forms.control.refresh?view=windowsdesktop-7.0>
11. <https://www.youtube.com/watch?v=Ubl4vzjXLl4&t=839s>
12. <https://ru.stackoverflow.com/questions/759534>
13. <https://metanit.com/sharp/tutorial/15.3.php>
14. <https://learn.microsoft.com/en-us/dotnet/api/system.windows.forms.datagridview.cellenter?view=windowsdesktop-7.0#system-windows-forms-datagridview-cellenter>
15. <https://metanit.com/sharp/tutorial/3.13.php>
16. <https://metanit.com/sharp/tutorial/5.4.php>
17. <https://metanit.com/sharp/windowsforms/2.4.php>
18. <https://ru.stackoverflow.com/questions/664671>
19. <https://metanit.com/sharp/windowsforms/4.7.php>
20. <https://learn.microsoft.com/en-us/dotnet/api/system.windows.forms.datagridviewcomboboxcolumn?view=windowsdesktop-7.0>
21. <https://metanit.com/sharp/tutorial/3.35.php>
22. <https://ru.stackoverflow.com/questions/258989>
23. <https://www.youtube.com/watch?v=ltvFrWCqL6g&t=1745s>
24. <https://www.youtube.com/watch?v=Ubl4vzjXLl4>
25. <https://www.youtube.com/watch?v=WSBy_Ypgk38&list=PLwG-AtjFaHdO4Xs6u9RZkYchZL8M2H9Gj>
26. <https://www.youtube.com/watch?v=ljrJOhsuZC8&t=594s>

**ДОДАТКИ**

**ДОДАТОК А. Алгоритми роботи програми**

Рисунок А.1 – Загальна блок-схема алгоритму роботи програми