



**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТЕХНОЛОГІЙ**

ННІ/факультет	<u>Інформаційних технологій</u>
Кафедра	<u>Інформатики і прикладного програмного забезпечення</u>
Спеціальність	<u>Інженерія програмного забезпечення</u>
Форма навчання	<u>Денна</u>

**«ЗАТВЕРДЖУЮ»**

Завідувач кафедри \_\_\_\_\_ Зеленський О.С.  
(підпис) (Прізвище, ініціали)  
« 11 » червня 2025 року

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ БАКАЛАВРСЬКУ РОБОТУ**

1. Тема роботи Розробка програмного забезпечення продажу дисків студії звукозапису

Керівник роботи к.т.н. Медведєв Д.Г.  
затверджені наказом закладу вищої освіти від «04» березня 2025 р. № 222-ст

2. Строк подання здобувачем роботи до «09» червня 2025 р.

3. Зміст кваліфікаційної роботи, об'єкт, предмет та мета дослідження:

**Розділ 1.** Постановка задачі

**Розділ 2.** Розробка алгоритму розв'язання задачі

**Розділ 3.** Організація інформаційного забезпечення

**Розділ 4.** Розробка програмного забезпечення задачі

*Об'єкт дослідження:* організація продажу дисків студії звукозапису

*Предмет дослідження:* інформаційна підтримка процесу автоматизації продажу дисків студії звукозапису

*Мета кваліфікаційної роботи:* розробка програмного забезпечення продажу дисків студії звукозапису

5. Дата видачі завдання «04» березня 2025 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів МДР	Строк виконання етапів роботи	Відмітка керівника про виконання етапів (дата, підпис)
1	Підготовка розділу 1	04.04.2025-13.04.2025	
2	Підготовка розділу 2	14.04.2025-26.04.2025	
3.	Підготовка розділу 3	27.04.2025-15.05.2025	
4	Підготовка розділу 4	16.05.2025-08.06.2025	
5	Реєстрація завершеної кваліфікаційної роботи	09.06.2025	Реєстраційний №____ «09»червня 2025 р.
6	Отримання відгуку від наукового керівника	10.06.2025	
7	Подання кваліфікаційної роботи на перегляд завідувачу кафедри	11.06.2025	
8	Отримання зовнішньої рецензії	12.06.2025	
9	Попередній захист кваліфікаційної роботи на кафедрі	13.06.2025	
10	Підготовка до захисту в ЕК	16.06.2025-21.06.2025	

Завдання підготував науковий керівник

\_\_\_\_\_ (підпис)

\_\_\_\_\_ Медведєв Д.Г.

\_\_\_\_\_ (прізвище та ініціали)

Завдання одержав

\_\_\_\_\_ (підпис)

\_\_\_\_\_ Лигун М.К.

\_\_\_\_\_ (прізвище та ініціали)

## **ЗГОДА здобувача вищої освіти**

Державного університету економіки і технологій про перевірку кваліфікаційної роботи на прояви академічного плагіату та розміщення в Репозитарії Університету

Я, Лигун Максима Костянтиновича (ПП), підтримую політику Державного університету економіки і технологій з академічної доброчесності і відкритого доступу.

Засвідчую, що кваліфікаційна бакалаврська робота «Розробка програмного забезпечення продажу дисків студії звукозапису» виконана самостійно та не містить академічного плагіату. Я не надавав і не одержував недозволену допомогу під час підготовки цієї роботи. Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Із чинним Положенням про запобігання та виявлення академічного плагіату в роботах здобувачів вищої освіти Державного університету економіки і технологій ознайомлений. Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі порушення норм академічної доброчесності робота не допускається до захисту або оцінюється незадовільно.

Також я поінформований, що відповідно до «Положення про Репозитарій Державного університету економіки і технологій» зазначена робота буде розміщена в Електронному архіві Університету. З умовами такого розміщення ознайомлений(на).

Дата

підпис

ініціали, прізвище (власноруч)

## **АНОТАЦІЯ**

### **на кваліфікаційну бакалаврську роботу**

#### **«Розробка програмного забезпечення продажу дисків студії звукозапису»**

**Лигуна Максима Костянтиновича**

Кваліфікаційна бакалаврська робота на здобуття освітньо-кваліфікаційного рівня бакалавра зі спеціальності 121 «Інженерія програмного забезпечення» – Державний університет економіки і технологій – Кривий Ріг, 2025.

У кваліфікаційній роботі розроблено програмне забезпечення, що дозволяє автоматизувати процес продажу дисків студії звукозапису та дозволяє вирішити наступні задачі:

- ведення даних по дискам;
- ведення даних по клієнтам;
- організація продажу дисків, запис їх до бази даних та автоматичне формування чеків;
- автоматизоване формування звітів та їх експорт в MS Excel та HTML.

Програмне забезпечення розроблено на мові C# з використанням технології ADO .NET для обробки баз даних.

Результати роботи рекомендуються до застосування у студіях звукозапису з метою автоматизації продажу дисків.

Ключові слова: програмне забезпечення, СУБД, ADO .NET, студія звукозапису.

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

.NET Framework	Платформа фірми Microsoft, призначена для розробки нативних та web-додатків
ПЗ	Програмне забезпечення
БД	База даних
СУБД	Система управління базами даних
ADO .NET	ActiveX Data Object для .NET – технологія доступу і управління базами даних для платформи .NET

## ЗМІСТ

ВСТУП .....	7
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ.....	9
1.1. Характеристика задачі .....	9
1.2. Вхідна інформація.....	11
1.3. Вихідна інформація.....	12
РОЗДІЛ 2. РОЗРОБКА АЛГОРИТМУ РОЗВ’ЯЗАННІ ЗАДАЧІ .....	13
2.1. Функціональне забезпечення задачі.....	13
2.2. Алгоритм розв’язання задачі .....	15
2.3. Розробка алгоритму вирішення задачі «Diski_3D» .....	18
РОЗДІЛ 3. ОРГАНІЗАЦІЯ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ.....	20
РОЗДІЛ 4. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗАДАЧІ.....	23
4.1. Розробка програмного забезпечення.....	23
4.2. Опис програми організації продажу дисків «Diski_3D».....	45
4.3. Опис скриптів для формування діаграм у форматі HTML. ....	47
ВИСНОВКИ.....	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	51
ДОДАТКИ.....	53

## ВСТУП

У сучасних умовах інформаційне забезпечення студії звукозапису набуває все більшого значення. Першим кроком до створення повноцінного програмного забезпечення, яке дозволить реалізовувати аудіопродукцію на різних студіях із можливістю формування звітів і довільних документів, є розробка автоматизованої бази даних для ведення обліку на студії.

Оскільки ключову роль у формуванні вимог до інформаційної системи відіграють цілі й завдання, визначені замовником, розробнику необхідно ретельно їх проаналізувати для точного розуміння. З цією метою була організована робоча зустріч з представниками студії, під час якої виявлено головну проблему – відсутність електронного каталогу з відомостями про музичні твори та їх виконавців. Це значно ускладнює пошук композицій для клієнтів і знижує здатність студії своєчасно виконувати замовлення.

Відсутність інформаційної системи обліку заважає оперативно знаходити потрібні записи, враховувати запити клієнтів і забезпечувати високий рівень сервісу. Це призводить до втрати часу, неефективного використання ресурсів та підвищених витрат.

Автоматизація щоденних процесів, що включають багато рутинних операцій і потребують значних витрат часу, є актуальним завданням для підвищення продуктивності. Впровадження сучасних ІТ-рішень дозволить значно покращити ефективність роботи студії, оптимізувати процеси обслуговування клієнтів і закладе підґрунтя для подальшої модернізації всієї системи.

Мета роботи – розробка програмного забезпечення продажу дисків студії звукозапису.

Об'єкт роботи – організація продажу дисків студії звукозапису.

Предмет роботи – інформаційна підтримка процесу автоматизації продажу дисків студії звукозапису.

При розробці програмного забезпечення буде використано мову C# у

середовищі Microsoft Visual Studio 2019.

Дана система призначена для ведення та оформлення продажу дисків студії звукозапису. При цьому планується вирішення наступних задач:

1. Ведення даних по дискам.
2. Ведення даних по клієнтам.
3. Ведення продаж дисків, запис їх до бази даних та автоматичне формування чеків.
4. Автоматизоване формування звітів та їх експорт в MS Excel та в HTML.

Зацікавленими організаціями в розробці програмного забезпечення є студії звукозапису. Також програмне забезпечення може бути використане в навчальному процесі при викладанні дисциплін, пов'язаних з .NET-технологіями.

## РОЗДІЛ 1

### ПОСТАНОВКА ЗАДАЧІ

#### 1.1. Характеристика задачі

Розробка програмного забезпечення для реалізації дисків у студії звукозапису поділяється на три ключові етапи:

1. *Аналіз предметної області, проектування та створення інтерфейсу програми.* На цьому етапі передбачається опитування замовника та потенційних користувачів для вивчення їхніх потреб. Будуть створюватися первинні ескізи майбутнього програмного продукту. Для роботи можуть знадобитися ліцензійні інструменти, зокрема Adobe Photoshop CS6 та Microsoft Project, які використовуються для візуального оформлення і планування процесу розробки.

2. *Розробка логіки та архітектури програми. Створення компонентів та тестування.* Цей етап включає проектування алгоритмів, програмування окремих модулів системи, збирання повноцінного програмного продукту, а також комплексне тестування з метою перевірки його функціональності, надійності та зручності використання.

3. *Підготовка технічної документації. Створення інструкцій.* На завершальному етапі розробляються документація до програми, інструкції для користувачів і розробників. Для цього використовуються офісні пакети, зокрема Microsoft Office.

У даній роботі буде розроблятися програма на мові C# у середовищі Visual Studio 2019, яка буде забезпечувати автоматизацію продажу дисків для студії звукозапису. У ході виконання даної роботи планується розробити наступні класи та функції в них:

1. Клас Form1 – головна форма програми.

2. Клас Form2 – форма ведення довідників (довідник по групам (виконавцям) та довідник по клієнтам).
3. Клас Form3 – форма ведення інформації по дискам (у кожного виконавця може бути багато дисків).
4. Клас Form4 – форма «Про програму».
5. Клас Form5 – форма експорту до формату html.
6. Клас Form6 – форма оформлення продажу дисків.
7. Клас Form7 – форма виведення інформації по продажам дисків.
8. Клас Name\_List – клас для заповнення списку.
9. Клас Klient – клас для заповнення списку клієнтів.
10. Клас Disk – клас для заповнення списку дисків.
11. ExportXls – універсальна статична функція програми для експорту довільної таблиці у формат \*.xls.
12. Export\_Html – універсальна статична функція програми для експорту довільної таблиці у формат \*.htm.

Ведення бази даних відбувається за допомогою зручних візуалізованих форм, де передбачена робота не тільки з текстовою, а й з графічною інформацією. Уся звітна документація виводиться до пакету MS Excel, а також до формату html.

Процес вирішення задачі проходить в режимі діалогу з програмою, що дозволяє:

- вести довідникову інформацію по дискам та клієнтам;
- формувати дані по продажу дисків за рахунок зручного інтерфейсу, а також автоматично формувати розрахункові документи з подальшим їх записом до БД;
- робити зручні налаштування з їх записом до файлу ініціалізації.

Система працює на платформі .Net Framework версії 4.5. Тобто може працювати на операційних системах Windows 10 та Windows 11.

Дане програмне забезпечення матиме попит на ринку за рахунок своєї якості, простоти у використанні та інсталяції, інтуїтивно зрозумілого інтерфейсу, а також повноти виконання поставлених задач.

Програма не потребує роботи з нею спеціально навченого персоналу, цілком достатньо поверхневих знань роботи з комп'ютером. На рис. 1.1. наведемо послідовність розробки програмного забезпечення.



Рис. 1.1. Послідовність розробки програмного забезпечення

## 1.2. Вхідна інформація

Вхідною інформацією вважаються ті дані, які необхідні для вирішення аналітичних задач. Вона є найдетальнішою та служить базою для подальшої логічної й числової обробки. До складу вхідної інформації можуть входити

як змінні, так і постійні або умовно-постійні дані, при цьому особливе значення має саме умовно-постійна інформація.

До вхідної інформації належить файл бази даних «diski.mdb», який вміщує інформацію по дискам студії звукозапису, клієнтам які їх купують, а також інформацію по їх продажу на конкретну дату.

### 1.3. Вихідна інформація

Вихідними даними є результати роботи з базою даних. Обробка даних, що містяться в базі або додаються до неї, здійснюється за допомогою алгоритмів розробленої програми. Управління базою даних реалізується через зручні візуалізовані інтерфейси, що підтримують роботу як з текстовою, так і з графічною інформацією. Вся звітна документація експортується до пакету MS Excel та зберігається у форматі HTML.

Розроблене програмне забезпечення дозволить автоматизувати процес продажу дисків студії звукозапису та вирішити наступні задачі:

- ведення даних по дискам;
- ведення даних по клієнтам;
- організація продажу дисків, запис їх до бази даних та автоматичне формування чеків;
- автоматизоване формування звітів та їх експорт в MS Excel та HTML.

### Висновки до розділу 1

В цьому розділі було сформовано головну мету роботи, характеристику роботи, засоби виконання поставленої задачі. Також було наведено вхідні та вихідні повідомлення, які дозволяють створити та побудувати базу даних для реалізації задачі, а також визначити архітектуру для розробки програмного забезпечення.

## РОЗДІЛ 2

### РОЗРОБКА АЛГОРИТМУ РОЗВ'ЯЗАННІ ЗАДАЧІ

#### 2.1. Функціональне забезпечення задачі

Для розробки програмного забезпечення для автоматизації продажу дисків звукозаписної студії було використано сучасну мову програмування C#. Для розробки застосовувалося середовище Visual Studio 2019. Проект створений для операційної системи Windows із застосуванням Windows Forms.

При розробці програмного забезпечення необхідно створити функціональну схему задачі. У нашому випадку програма повинна виконувати наступні функціональні задачі:

1. Ведення довідників:
  - 1.1. Ведення довідника по групам (виконавцям).
  - 1.2. Ведення інформації по дискам.
  - 1.3. Ведення інформації по клієнтам.
2. Продаж дисків.
  - 2.1. Оформлення продажу диску.
  - 2.2. Виведення інформації по продажам.

На рис. 2.1. наведемо функціональну схему даної задачі.

При розробці програмного забезпечення за даною функціональною схемою використаємо наступні назви класів:

1. Клас OleDbConnection – клас для підключення до бази даних.
2. Клас OleDbDataAdapter – клас для взаємодії між таблицею бази даних та локальним контейнером DataTable.
3. Клас DataSet – контейнер таблиць DataTable.
4. Клас OleDbCommandBuilder – клас для формування SQL-запитів до таблиці бази даних.
5. Клас OleDbCommand – клас для виконання команд.
6. Клас OleDbDataReader – клас для роботи з об'єктом для читання

даних.

7. Клас DataTable – клас для роботи з таблицею даних.

8. Клас DataGridView – клас для модифікації даних таблиці (видалення, додавання, редагування записів).

9. Клас BindingSource – клас для прив'язки даних набору записів.

10. Клас BindingNavigator – клас рядку навігації по таблиці бази даних.



Рис. 2.1. Функціональна схема задачі «Автоматизація продажу дисків»

Крім перелічених класів роботи з базами даних на основі технології ADO.NET планується розробити власні 7 візуальних класів:

1. Клас Form1 – головна форма програми.
2. Клас Form2 – форма ведення довідників (довідник по групам (виконавцям) та довідник по клієнтам).
3. Клас Form3 – форма ведення інформації по дискам (у кожного виконавця може бути багато дисків).
4. Клас Form4 – форма «Про програму».
5. Клас Form5 – форма експорту до формату html.
6. Клас Form6 – форма оформлення продажу дисків.
7. Клас Form7 – форма виведення інформації по продажам дисків.

## 2.2. Алгоритм розв'язання задачі

Для реалізації задачі за функціональною схемою наведемо алгоритм кожної з функціональних задач, а саме ведення довідників (по групам (виконавцям), або клієнтам) та ведення інформації по дискам.

На рис. 2.2. представимо блок-схему алгоритму ведення довідників (по групам (виконавцям), або клієнтам) (Клас Form2).

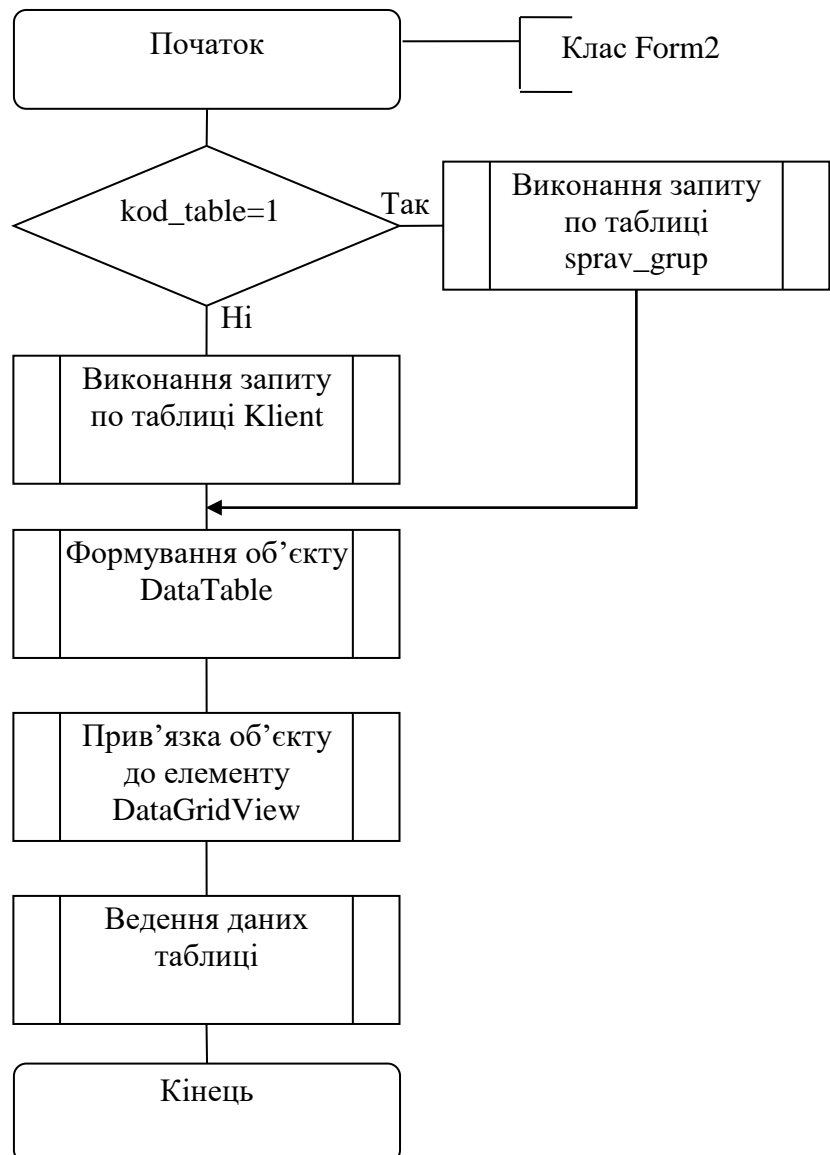


Рис. 2.2. Блок-схема алгоритму вирішення функціональної задачі «Ведення довідників»

На першому етапі необхідно створити об'єкт да класу OleDbDataAdapter для отримання SQL-запиту по групам виконавців, або окремим виконавцям. Таблиця в якій зберігається інформація по виконавцям

має назву `sprav_group`, таблиця з інформацією по клієнтам – `Klient`.

При завантаженні форми вказується змінна `kod_table`, яка відповідає за виконання певного SQL-запиту. Якщо змінна `kod_table` дорівнює нулю завантажуються таблиця `sprav_grup`, в іншому випадку – таблиця `Klient`.

Після цього за допомогою методу `Fill` об'єкту `da` створюється контейнер `DataTable` з локальними даними таблиці, які треба прив'язати до елемента управління таблиці `DataGridView` та до рядка навігації. Після цього можна повноцінно працювати з даними таблиці.

На рис. 2.3. представимо блок-схему алгоритму роботи задачі «Ведення інформації по дискам» (Клас `Form3`).

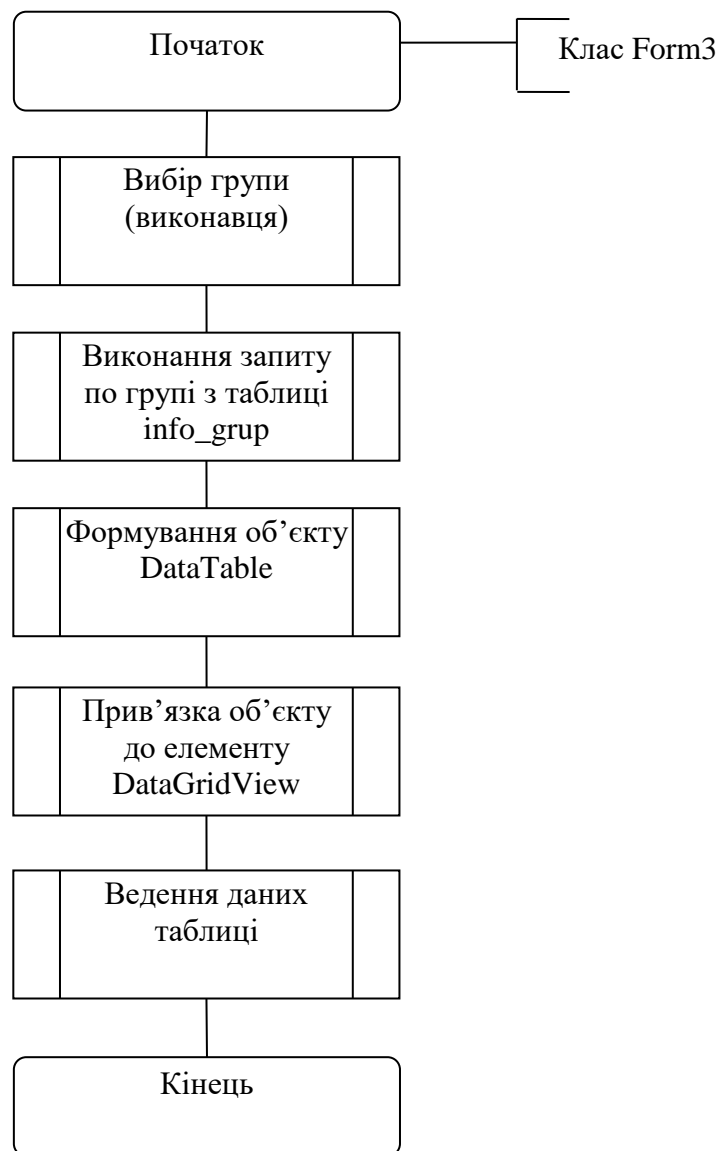


Рис. 2.3. Блок-схема алгоритму вирішення функціональної задачі «Ведення інформації по дискам»

У даному алгоритмі спочатку обирається певний виконавець або група виконавців зі списку, який в свою чергу формується з довідника виконавців. По цьому виконавцю вибираються диски, які до нього відносяться. Є можливість редагування, видалення, додавання інформації по дискам, а також вибір конкретної картинки, маршрут якої зберігається в базі даних.

На рис. 2.4. представимо блок-схему алгоритму роботи задачі «Оформлення продажу дисків» (Клас Form6).

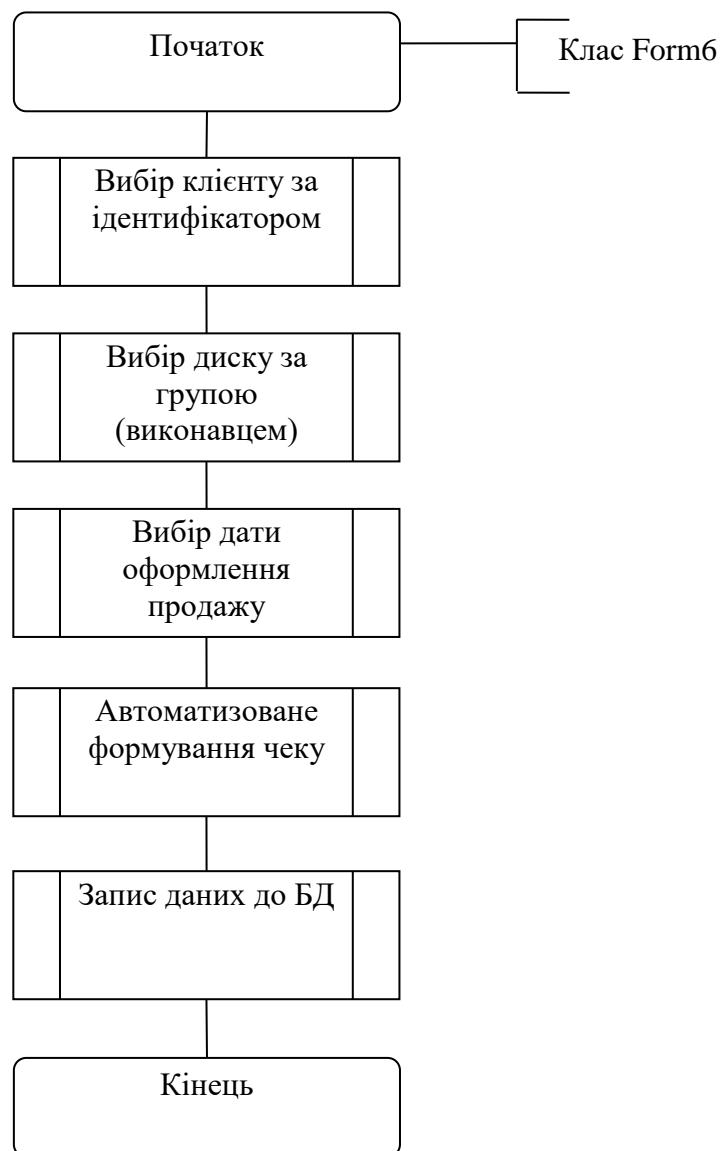


Рис. 2.4. Блок-схема алгоритму вирішення функціональної задачі «Оформлення продажу дисків»

У даному алгоритмі спочатку відбувається вибір клієнту за його номером ідентифікатору та вибір диску певної групи (виконавця), який хоче придбати клієнт.

Після цього відбувається вибір дати оформлення продажу, або встановлюється поточна дата. Наступним етапом здійснюється автоматизоване формування чеку у середовищі MS Excel, а також запис даних з оформлення продажу диску до бази даних.

### 2.3. Розробка алгоритму вирішення задачі «Diski\_3D»

Для реалізації задачі необхідно розробити алгоритм. Робота з програмою «Diski\_3D» здійснюється на алгоритму, який зображено на рис. 2.5.

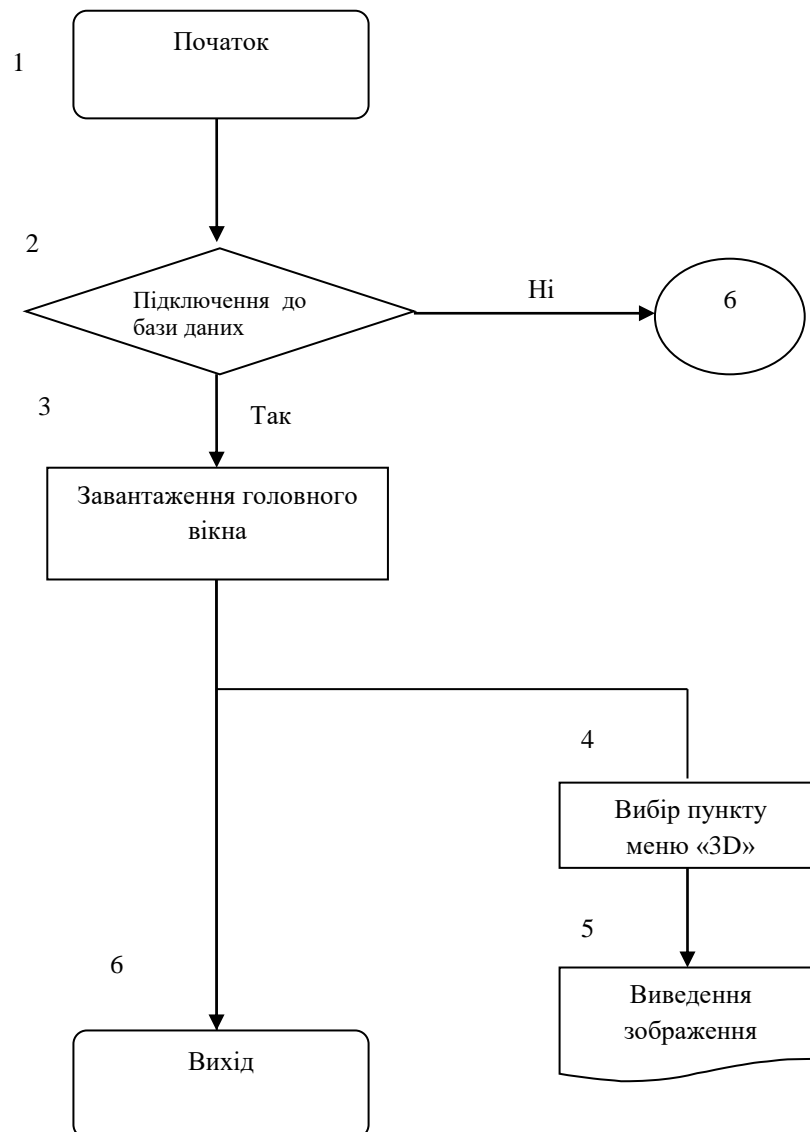


Рис. 2.5. Блок-схема алгоритму роботи програми «Diski\_3D»

## Висновки до розділу 2

Отже, у цьому розділі було успішно розроблено алгоритми вирішення задач «Diski», «Diski\_3D» шляхом побудови блок-схем алгоритмів цих задач. При створенні алгоритмів для вирішення цих задач було використано повний набір вхідних та вихідних даних задач.

## РОЗДІЛ 3

### ОРГАНІЗАЦІЯ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

Розглянемо інфологічну модель бази даних.

У якості середовища баз даних вибрано СУБД ACCESS. База даних має назву `diski.mdb` та зберігається у поточному каталозі. БД складається з чотирьох таблиць: `sprav_grup` – довідник груп, або виконавців; `info_grup` – інформація по дискам відповідного виконавця; `Klient` – довідник по клієнтам; `Prodaga` – інформація по продажам дисків.

У табл. 3.1 приведемо структуру таблиці `sprav_grup` (ID – ключове поле).

Таблиця 3.1

Структура таблиці `sprav_grup`

№ п/п	Назва	Тип поля	Значення
1	<b>ID</b>	Лічильник	Забезпечення нормалізації даних
2	Name	Текстове	Назва групи (виконавця)

У табл. 3.2 приведемо структуру таблиці `info_grup` (ID – ключове поле).

Таблиця 3.2

Структура таблиці `info_grup`

№ п/п	Назва	Тип поля	Значення
1	<b>ID</b>	Лічильник	Забезпечення нормалізації даних
2	ID_grup	Довге ціле	Зв'язка з полем ID таблиці <code>sprav_grup</code>
3	Info	Текстове	Інформація про диск
4	Years	Текстове	Роки (рік) випуску диску
5	Tip	Текстове	Тип музики
6	Sena	Грошовий	Ціна диску
7	Kart	Текстове	Маршрут картинки

У табл. 3.3 приведемо структуру таблиці Klient (Pasport – ключове поле).

Таблиця 3.3

## Структура таблиці Klient

№ п/п	Назва	Тип поля	Значення
1	<b>Pasport</b>	Текстове	Код ідентифікації клієнту
2	FIU	Текстове	Прізвище, ім'я, по-батькові клієнту
3	Pol	Текстове	Стать клієнту: «Ч» – чоловіча, «Ж» – жіноча
4	Voz	Довге ціле	Вік клієнту

У табл. 3.4 приведемо структуру таблиці Prodaga (Складний ключ, який складається з полів: Pasp\_Kl, ID\_disk, Data\_prod).

Таблиця 3.4

## Структура таблиці Prodaga

№ п/п	Назва	Тип поля	Значення
1	ID	Лічильник	Забезпечення нормалізації даних
2	<b>Pasp_Kl</b>	Текстове	Зв'язка з полем Pasport таблиці Klient
3	<b>ID_disk</b>	Довге ціле	Зв'язка з полем ID таблиці info_grup
4	<b>Data_prod</b>	Дата/Час	Дата продажу диску
5	Sena_prod	Грошовий	Ціна продажу диску

Дані таблиці пов'язані між собою відношеннями один до багатьох.

На рис. 3.1. наведемо схему даних бази даних «diski.mdb». З даної схеми видно реляційні зв'язки типу один-до-багатьох. Так, таблиця sprav\_grup пов'язана з таблицею info\_grup відношеннями один-до-багатьох, тобто у кожній групі (виконавця) може бути багато дисків. Безпосередньо таблиця, яка характеризує диск «info\_grup» пов'язана з таблицею продаж «Prodaga» відношенням один-до-багатьох, так як один і той самий диск може продаватися багато разів. Крім того, ведеться інформація про клієнтів (таблиця «Klient»), які також пов'язані з таблицею «Prodaga» відношенням

один-до-багатьох.

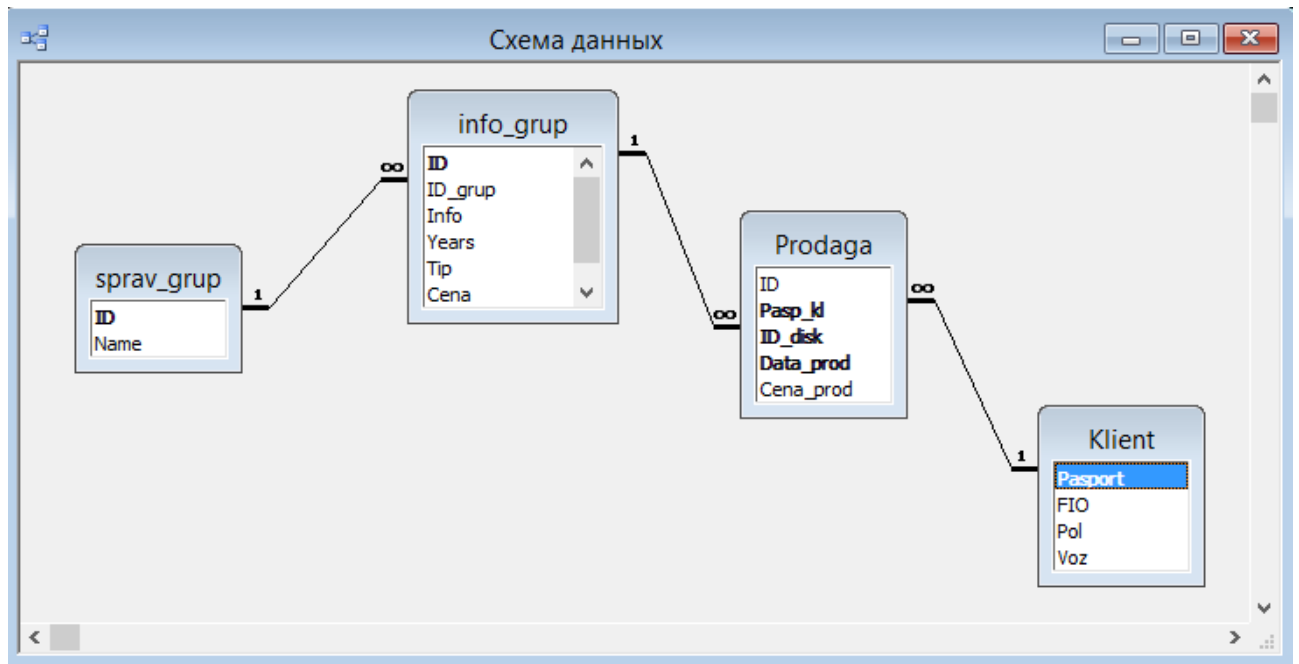


Рис. 3.1. Схема даних бази даних «diski.mdb»

Дана інфологічна модель бази даних дозволяє повністю автоматизувати процес продажу дисків для студії звукозапису.

В якості технології роботи з базою даних використано технологію ADO .NET.

### Висновки до розділу 3

В цьому розділі було розглянуто характеристику бази даних MS Access «diski.mdb», яка дозволяє ефективно організувати продаж дисків студії звукозапису. Наведена схема даних дозволяє ефективно вести довідники клієнтів, а також характеристику дисків, які призначено для продажу.

## РОЗДІЛ 4

### РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗАДАЧІ

#### 4.1. Розробка програмного забезпечення

##### *Головна форма програми*

Розробка програмного забезпечення реалізації задачі автоматизації продажу дисків студії звукозапису виконана на сучасній мові програмування C# у середовищі Visual Studio 2019. Дане програмне забезпечення виконано з використанням Windows Forms. Всі візуальні класи програми похідні від класу Form. Інтерфейс головної форми програми приведено на рис. 4.1.



Рис. 4.1. Головна форма програми

У головній формі передбачено п'ять опцій меню [додаток А]:

1. Довідники.
2. Продаж дисків.

3. Налаштування.
4. Діаграма в HTML.
5. Про програму.

Опція «Довідники» складається з трьох пунктів меню:

1. «Групи» – ведення довідника груп (виконавців).
2. «Інформація про диски» – вибір конкретного виконавця та ведення інформації по ньому.
3. «Клієнти» – ведення довідника клієнтів.

Опція «Продаж дисків» складається з двох пунктів меню:

1. «Оформлення продажу» – оформлення продажу вибраного диску певним клієнтом з автоматизованим формуванням чеку та записом в базу даних.
2. «Інформація про продажі» – виведення інформації про продаж дисків з фільтрацією за певний проміжок часу.

Опція «Налаштування» складається з двох пунктів меню:

1. «Вибір шрифту» – вибір шрифту для виведення текстової звітної інформації.
2. «Вибір кольору» – вибір кольору відповідно для заголовка таблиці, заголовків стовпців таблиці та тексту таблиці.

Всі вибрані налаштування відображаються у вигляді звітної інформації у форматах \*.xls та \*.htm, а також при закритті програми записуються до файлу налаштувань «disk.ini».

При виконанні опції «Про програму» на екран виводиться інформація про розробника програмного забезпечення (рис. 4.2.).

У головній формі програми відбувається підключення до бази даних «diski.mdb» з використанням об'єкту OleDbConnection. Потім даний об'єкт передається до інших форм програми.

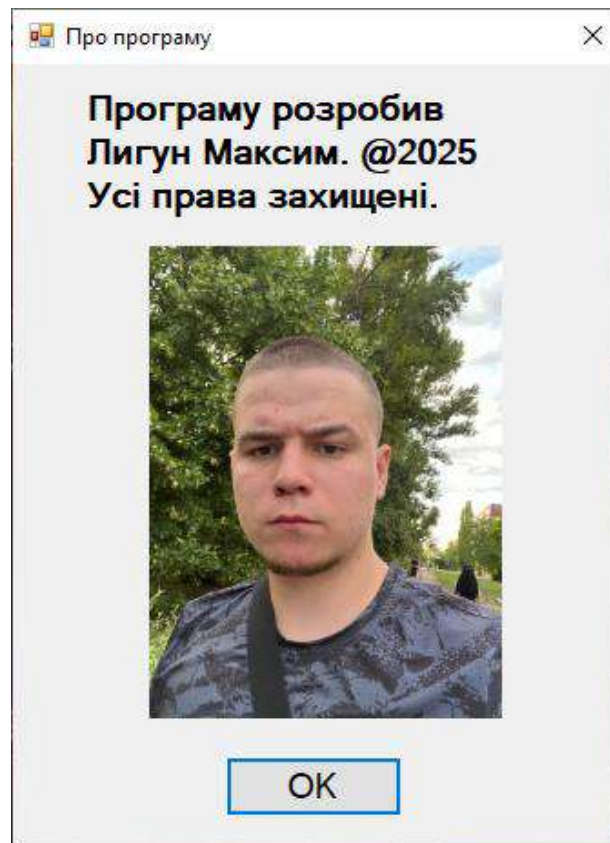


Рис. 4.2. Форма «Про програму»

### *Розробка програмного забезпечення та тестування задачі*

Розглянемо клас Form2 для ведення довідників (довідника груп (виконавців) та довідника клієнтів).

Проектування форми виглядає наступним чином (рис. 4.3).

Зверху форми знаходиться елемент напису Label, посередині елемент роботи з таблицями DataGridView, елемент знизу – рядок навігації даних (клас BindingNavigator). Слід зауважити, що до рядку навігації додано кнопку «Оновити» для оновлення даних в таблиці БД, а також кнопки із значком «xls» для експорту даних в MS Excel та кнопки із значком «html» для експорту даних в формат html.

У формі має місце цілочисельна змінна kod\_table, яка відповідає за завантаження конкретного довідника (SQL-запиту). Якщо змінна kod\_table дорівнює одиниці, завантажуються довідник з інформацією по групам (виконавцям) – «sprav\_grup», інакше – довідник по клієнтам – «Klient».

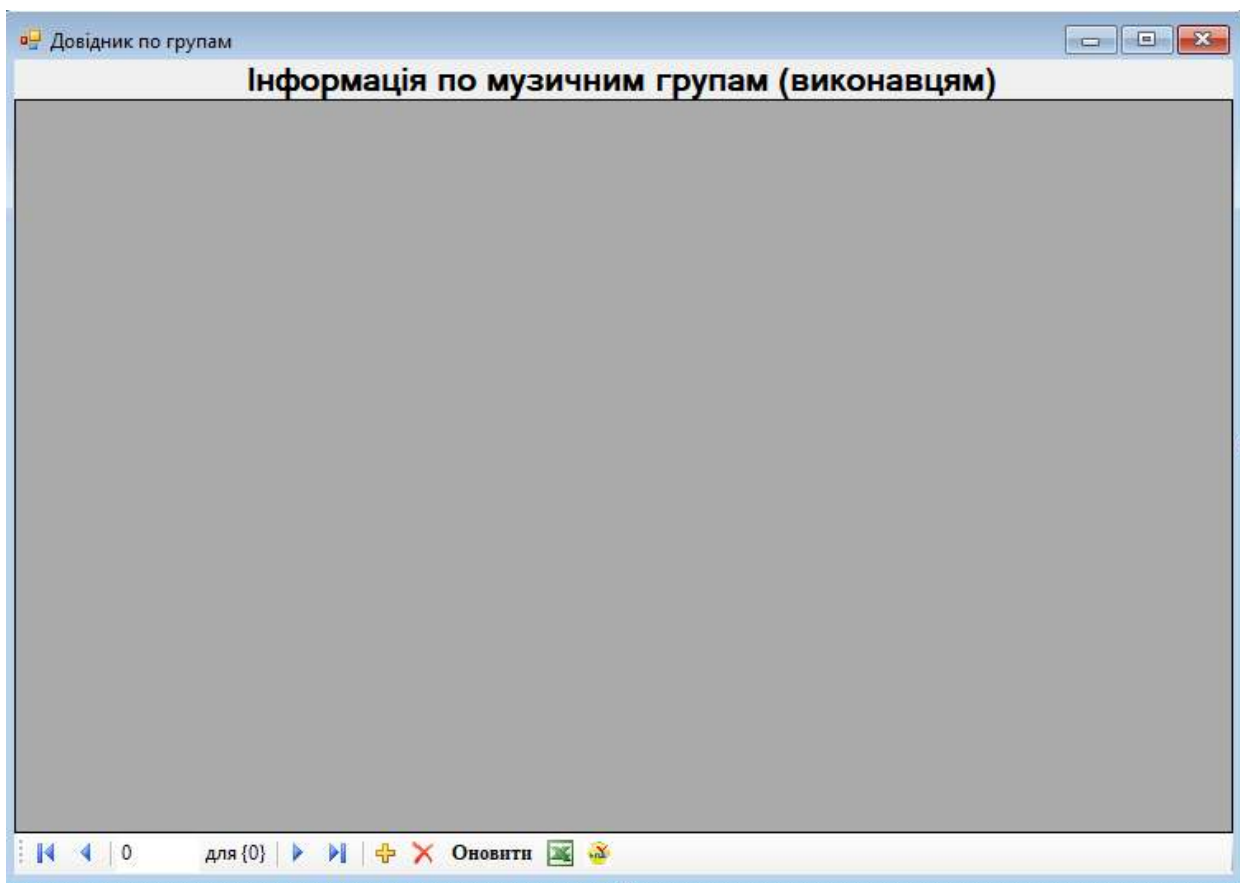


Рис. 4.3. Проектування форми Form2

Наведемо список та коротку характеристику основних функцій класу Form2.

1. Form2\_Load – функція завантаження форми.
2. toolStripButton1\_Click – функція оновлення даних.
3. toolStripButton2\_Click – функція експорту даних в MS Excel.
4. toolStripButton3\_Click – функція експорту даних в HTML.

Приведемо код функції Form2\_Load.

```
//Завантаження форми
private void Form2_Load(object sender, EventArgs e)
{
    try
    {
        if (kod_table == 1)
        {
            Text = "Довідник по групам";
            label1.Text = "Інформація по музичним групам (виконавцям)";

            da = new OleDbDataAdapter("Select ID, name from sprav_grup", conn);
            bulder = new OleDbCommandBuilder(da);
            ds = new DataSet();
        }
    }
}
```

```

        DataTableMapping datam = da.TableMappings.Add("Table", "Групи");
        datam.ColumnMappings.Add("ID", "№");
        datam.ColumnMappings.Add("name", "Група (виконавець)");
    }

    else
    {
        Text = "Довідник по клієнтам";
        label1.Text = "Довідник по клієнтам";

        da = new OleDbDataAdapter("Select Pasport, FIO, Pol, Voz from
Klient", conn);

        bulder = new OleDbCommandBuilder(da);
        ds = new DataSet();

        DataTableMapping datam = da.TableMappings.Add("Table", "Клієнти");
        datam.ColumnMappings.Add("Pasport", "Ідентифікатор");
        datam.ColumnMappings.Add("FIO", "ПІБ");
        datam.ColumnMappings.Add("Pol", "Стать");
        datam.ColumnMappings.Add("Voz", "Вік");

    }

    da.Fill(ds);

    bindingSource1.DataSource = ds.Tables[0];
    dataGridView1.DataSource = bindingSource1;
    bindingNavigator1.BindingSource = bindingSource1;
    if (kod_table == 1)
    {
        dataGridView1.Columns[0].Visible = false;
        dataGridView1.Columns[1].Width = 350;
    }
}

catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

```

У даній функції створюється об'єкт класу адаптеру da для взаємодії з таблицею «sprav\_grup» або «Klient» бази даних в залежності від змінної kod\_table. З використанням цього об'єкту створюється набір записів, який прив'язується до елементу DataGridView. Після цього можна у повному обсязі редагувати дані таблиці – додавати, видаляти, модифікувати записи. Слід зауважити, що всі зміни відбуваються тільки в елементі DataGridView. Щоб відповідні зміни відбулися у базі даних – необхідно натиснути кнопку оновлення даних. За формування запитів на оновлення (UPDATE), видалення (DELETE) та додавання (INSERT) даних відповідає клас

## OleDbCommandBuilder.

Наведемо код функції оновлення даних.

```
//Оновлення даних
private void toolStripButton1_Click(object sender, EventArgs e)
{
    try
    {
        da.Update(ds.Tables[0]);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

Після спрацювання даної функції буде внесено зміни до бази даних.

Наведемо код функцій відгуків експорту даних в MS Excel та HTML.

```
//Експорт даних в Excel
private void toolStripButton2_Click(object sender, EventArgs e)
{
    if (!Form1.ExportXLS(ds.Tables[0].DefaultView))
    {
        MessageBox.Show("Неможливий експорт в Excel!",
            "Увага!", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

//Експорт даних в HTML
private void toolStripButton3_Click(object sender, EventArgs e)
{
    if (!Form1.Export_Html(ds.Tables[0].TableName + ".htm",
        ds.Tables[0].DefaultView, ds.Tables[0].TableName,
        Form1.font_text, Form1.cvet_zag, Form1.cvet_hapka, Form1.cvet_text))
    {
        MessageBox.Show("Неможливий експорт в HTML!",
            "Увага!", MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }

    Form5 ff = new Form5();
    ff.webBrowser1.Navigate(Application.StartupPath + "\\\" +
        ds.Tables[0].TableName + ".htm");
    ff.ShowDialog();
}
```

На рис. 4.4. наведено результат виконання класу Form2 для ведення інформації по групам (виконавцям).

На рис. 4.5. наведено результат виконання класу Form2 для ведення інформації клієнтам.

Довідник по групам

### Інформація по музичним групам (виконавцям)

Група (виконавець)
ABBA
AC/DC
ACCEPT
ADRIANO CELENTANO
AEROSMITH
AIRBOURNE
ALESTORM
ALICE COOPER
ALICE IN CHAINS
ALLMAN BROTHERS BAND
ANNHILATOR
ARCH ENEMY

of 12 | Оновити

Рис. 4.4. Ведення довідника груп (виконавців)

Довідник по клієнтам

### Довідник по клієнтам

Ідентифік.	ПІБ	Стать	Вік
AB043223	Лапшин	Ч	20
AB234790	Рамзаєв	Ч	23
AB928340	Чертов	Ч	27
AB234980	Нечаєв	Ч	43
AB324789	Алієв	Ч	34
AB349028	Іванов	М	30
AK092384	Лозова	Ж	18
AK218739	Романов	Ч	22
AK438030	Богун	Ч	29
AK938204	Берген	Ч	18
AM230432	Петров	Ч	34
AM230984	Сідоров	Ч	39
AM232984	Бочкова	Ж	45
AM234938	Андрєєв	Ч	37
AT320948	Томашева	Ж	22
BP201323	Істван	Ч	21
BP214092	Ткачук	Ч	31
BP230943	Федоров	Ч	30
BP234238	Ференс	Ч	32

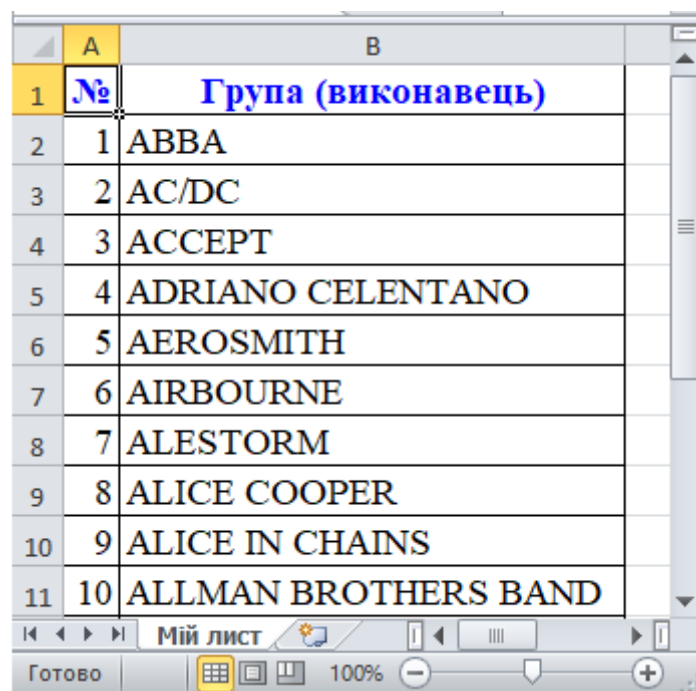
of 29 | Оновити

Рис. 4.5. Ведення довідника клієнтів

При натисненні на кнопку «xls» – інформацію буде виведено у MS Excel, звідки її можна роздрукувати у зручному для користувача вигляді (рис. 4.6).

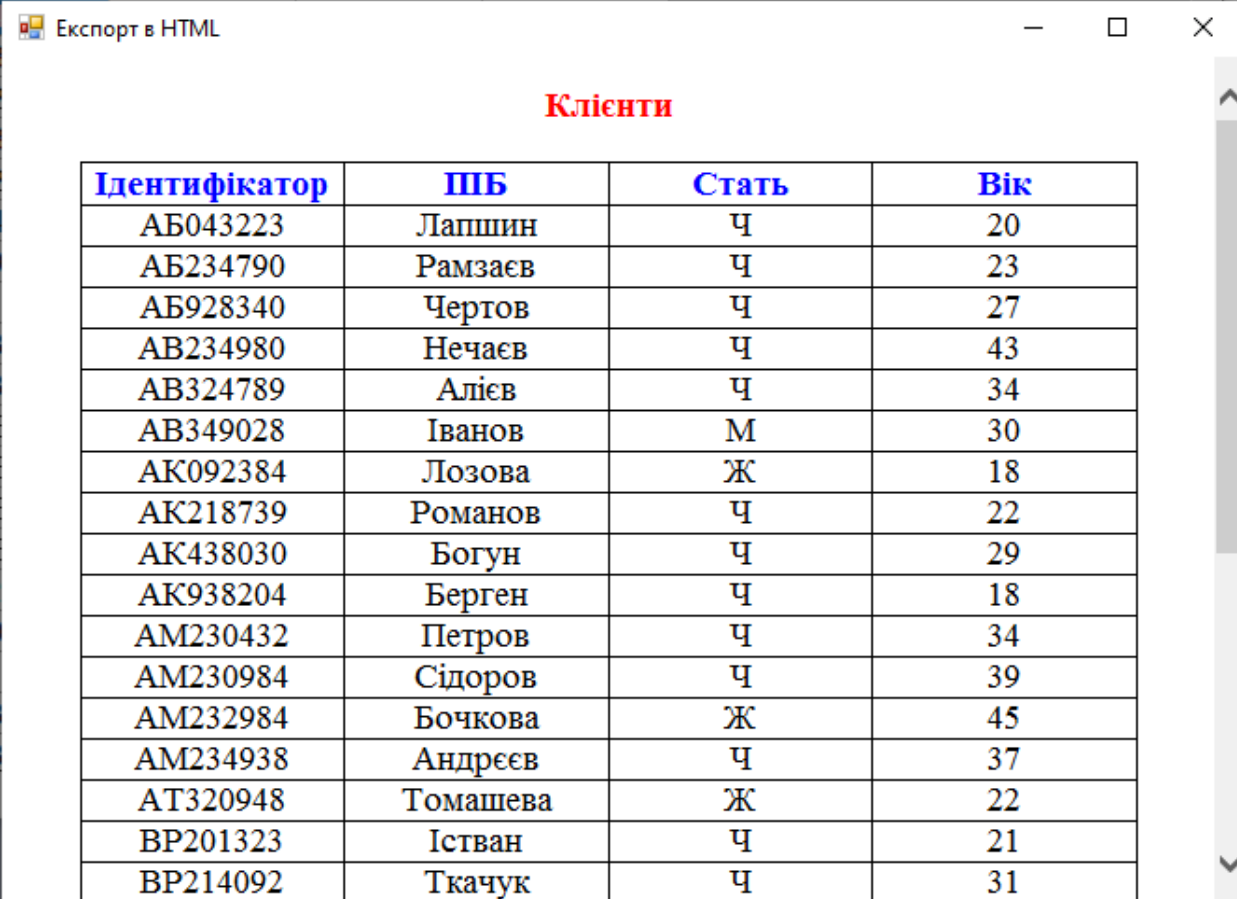
При натисненні на кнопку «html» – інформацію буде виведено у формат html (рис. 4.7). Код html-сторінки виводиться в розробленій програмі – у формі Form5. У якості елемента керування для перегляду web-сторінок використано елемент WebBrowser.

Наступним етапом є проектування класу Form3 де відбувається ведення інформації по дискам конкретного виконавця, або групи виконавців. Так як таблиці бази даних пов'язані відношенням один до багатьох, спочатку необхідно вибрати конкретну групу, а потім по ній сформулювати SQL-запит. З цією метою спроектовано клас Name\_List, який вміщує базову інформацію для будь-якого списку – строкове значення та значення ключового ідентифікатору – ItemData.



	A	B
1	№	Група (виконавець)
2	1	ABBA
3	2	AC/DC
4	3	ACCEPT
5	4	ADRIANO CELENTANO
6	5	AEROSMITH
7	6	AIRBOURNE
8	7	ALESTORM
9	8	ALICE COOPER
10	9	ALICE IN CHAINS
11	10	ALLMAN BROTHERS BAND

Рис. 4.6. Експорт даних до MS Excel



**Клієнти**

Ідентифікатор	ІПБ	Стать	Вік
АВ043223	Лапшин	Ч	20
АВ234790	Рамзаєв	Ч	23
АВ928340	Чертов	Ч	27
АВ234980	Нечаєв	Ч	43
АВ324789	Алієв	Ч	34
АВ349028	Іванов	М	30
АК092384	Лозова	Ж	18
АК218739	Романов	Ч	22
АК438030	Богун	Ч	29
АК938204	Берген	Ч	18
АМ230432	Петров	Ч	34
АМ230984	Сідоров	Ч	39
АМ232984	Бочкова	Ж	45
АМ234938	Андрєєв	Ч	37
АТ320948	Томашева	Ж	22
ВР201323	Істван	Ч	21
ВР214092	Ткачук	Ч	31

Рис. 4.7. Експорт даних в HTML

Наведемо код класу Name\_List.

```
//Клас для заповнення списку
public class Name_List
{
    public string str;
    public int ItemData;

    public Name_List(string str, int ItemData)
    {
        this.str = str;
        this.ItemData = ItemData;
    }

    public override string ToString()
    {
        return str;
    }
}
```

Спроекуємо форму Form3 для ведення даних інформації по дискам (рис. 4.8.).

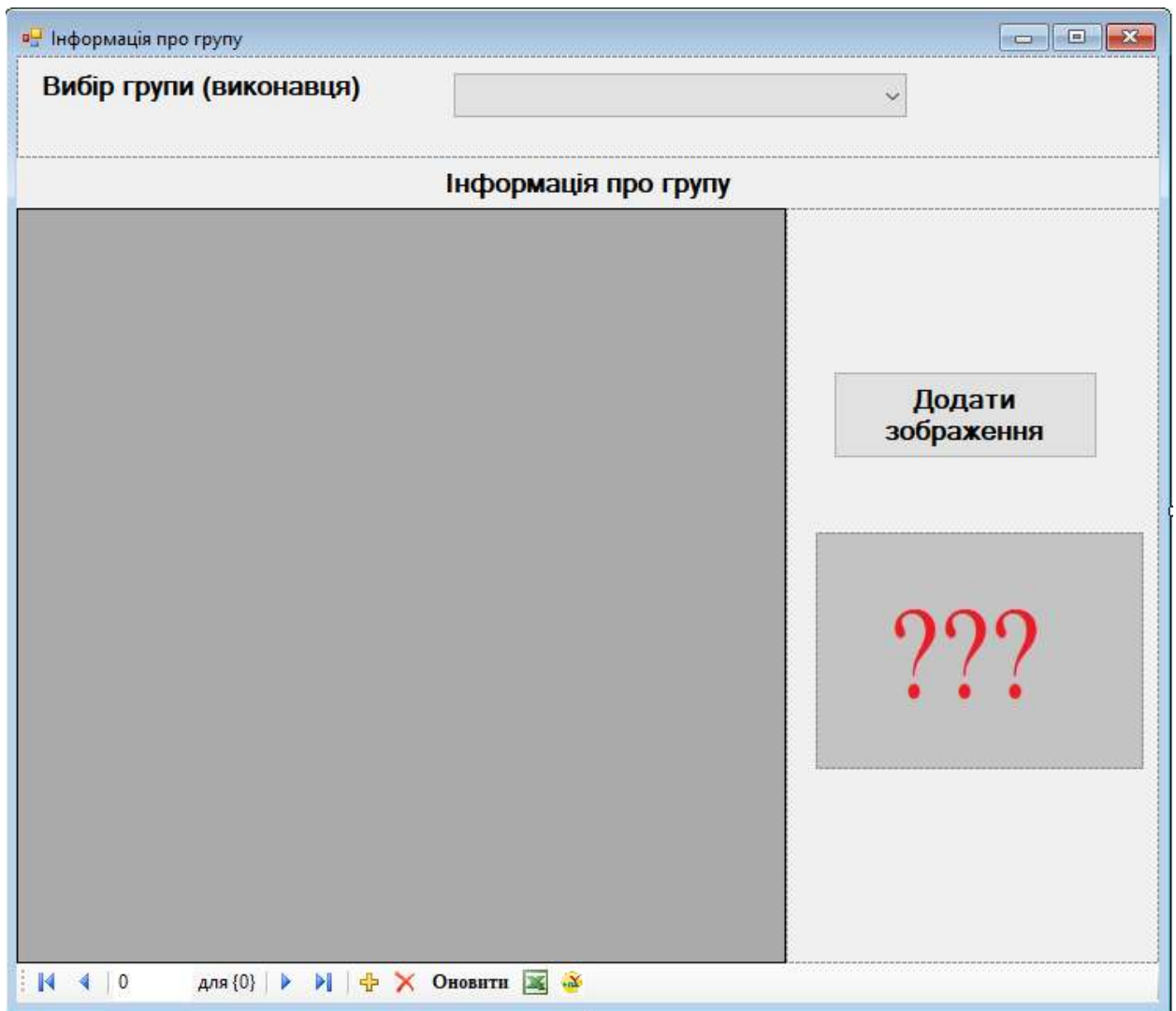


Рис. 4.8. Проектування форми Form3

Наведемо список та коротку характеристику основних функцій класу Form3.

1. Form3\_Load – функція завантаження форми.
2. comboBox1\_SelectedIndexChanged – вибір групи виконавця зі списку.
3. toolStripButton1\_Click – функція оновлення даних.
4. toolStripButton2\_Click – функція експорту даних в MS Excel.
5. toolStripButton3\_Click – функція експорту даних в HTML.
6. button1\_Click – функція додавання картинки.

Приведемо код функції Form3\_Load.

```
//Завантаження форми
private void Form3_Load(object sender, EventArgs e)
{
```

```

try
{
    // Заповнення списку
    cmd = new OleDbCommand("Select ID, name From sprav_grup", conn);
    datar = cmd.ExecuteReader();
    while (datar.Read())
        comboBox1.Items.Add(new Name_List(datar[1].ToString(),
Convert.ToInt32(datar[0])));

    comboBox1.SelectedIndex = 0;
    datar.Close();
    cmd.Dispose();
    //////////////////////////////////////

}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

```

У даній функції відбувається заповнення списку combobox1 інформацією по виконавцям з використанням класу Name\_List. Для виконання запиту використовується об'єкт OleDbCommand. Результат виконання записується в об'єкт для читання OleDbDataReader, який застосовується для заповнення елементів керування та призначений тільки для читання записів.

Далі наведемо код функції comboBox1\_SelectedIndexChanged, яка спрацьовує при виборі відповідного виконавця.

```

//Вибір групи(виконавця)
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    try
    {
        string str = "Select ID,Info,Years,Tip,Kart,Cena from info_grup where ";
        str += "ID_grup = " + ((Name_List)comboBox1.SelectedItem).ItemData;
        da = new OleDbDataAdapter(str, conn);
        bulder = new OleDbCommandBuilder(da);
        ds = new DataSet();

        DataTableMapping datam = da.TableMappings.Add("Table", "Диск");
        datam.ColumnMappings.Add("ID", "№");
        datam.ColumnMappings.Add("Info", "Інформація про диск");
        datam.ColumnMappings.Add("Years", "Рік випуску");
        datam.ColumnMappings.Add("Tip", "Тип музики");
        datam.ColumnMappings.Add("Cena", "Ціна");
        datam.ColumnMappings.Add("Kart", "Зображення");

        da.Fill(ds);

        bindingSource1.DataSource = ds.Tables[0];
        dataGridView1.DataSource = bindingSource1;
        bindingNavigator1.BindingSource = bindingSource1;
    }
}

```

```

dataGridView1.Columns[0].Visible = false;
dataGridView1.Columns[4].Visible = false;
dataGridView1.Columns[1].Width = 350;
bindingSource1.Position = 0;
pictureBox1.Image = Properties.Resources.No;
bindingSource1_PositionChanged(null, null);
}

catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

```

При виборі виконавця зі списку формується запит по конкретному ідентифікатору. Результат відповідного запиту по виконавцю прив'язується до елементу DataGridView з подальшими можливостями редагування.

Слід звернути увагу на функцію додавання зображення, яка відповідає конкретному диску. Картинка завантажується у форматі «\*.jpg». Вона вибирається з папки “Music”, яка знаходиться у поточному каталозі. Щоб не перевантажувати базу даних, до неї записується тільки маршрут зображення. У випадку відсутності зображення, або помилці при її виборі на екрані буде відображатись зображення з трьома знаками питання «???».

Наведемо код даної функції.

```

//Додавання зображення
private void button1_Click(object sender, EventArgs e)
{
    if (bindingSource1.Position < 0) return;

    openFileDialog1.Title = "Виберіть зображення у папці Music";
    string path = Path.GetDirectoryName(Application.ExecutablePath) +
@"\Music";
    openFileDialog1.InitialDirectory = path;

    // початкова назва файлу
    openFileDialog1.FileName = "";
    // фільтр
    openFileDialog1.Filter = "Зображення (*.jpg)|*.jpg";
    if (openFileDialog1.ShowDialog() != DialogResult.OK) return;
    pictureBox1.Image = Image.FromFile (openFileDialog1.FileName);

    ds.Tables[0].Rows[bindingSource1.Position][4] =
openFileDialog1.SafeFileName;
    toolStripButton1_Click(null, null);
}

```

На рис. 4.9. наведемо результат виконання класу Form3.

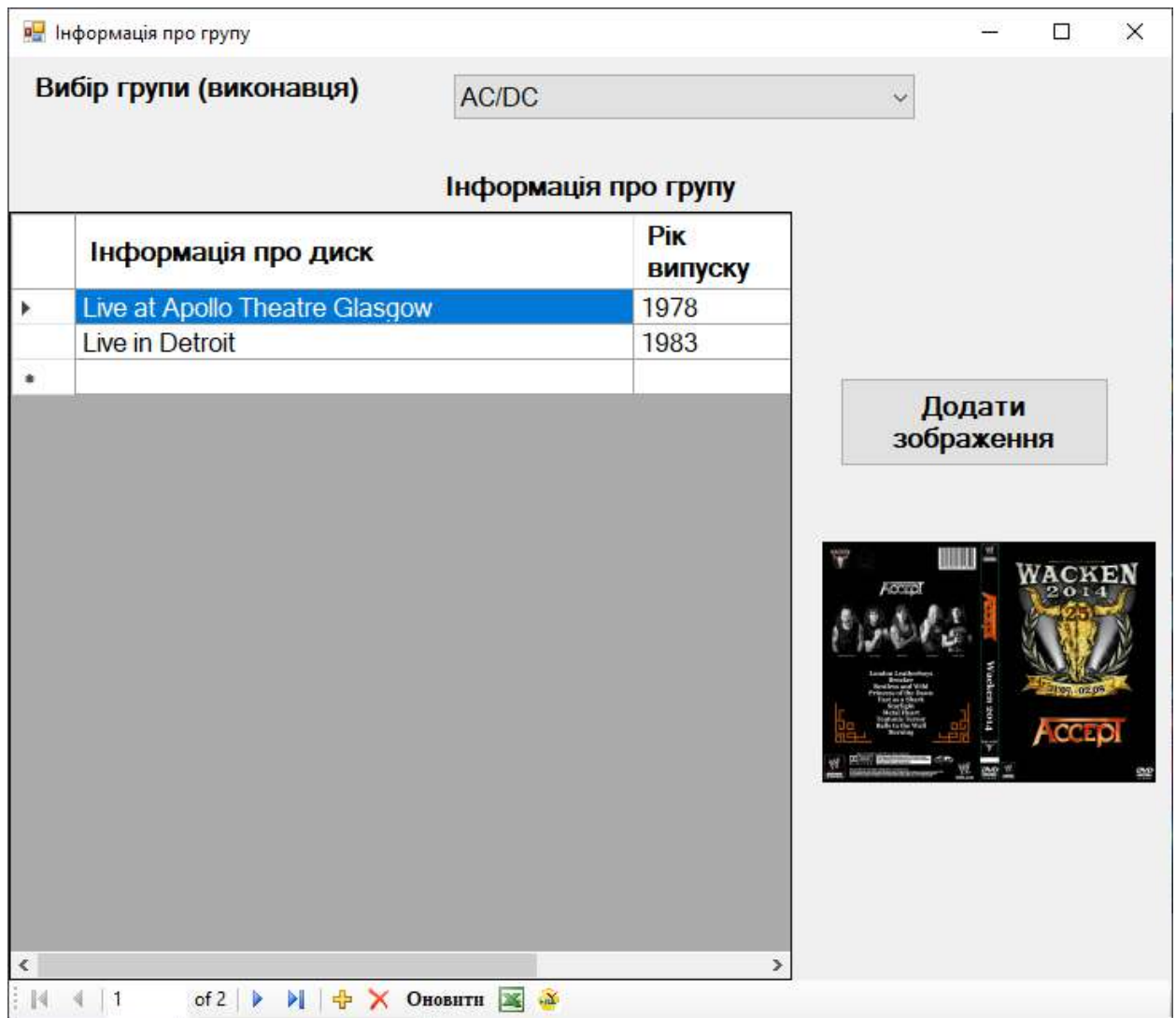


Рис. 4.9. Ведення інформації по дискам

При натисненні кнопки «Додати зображення» на екран виводиться діалог вибору картинки, де по умовчанням вибирається папка «Music» (рис. 4.10).

У випадку, якщо вибране не те зображення, або відбувається помилка – картинка з виконавцем не завантажується, а замість картинки буде встановлене зображення по умовчанням з трьома знаками питання «???».

Автоматизація ведення бази даних по дискам студії звукозапису є початковим етапом для оформлення продажу дисків з формуванням відповідної інформації.

Слід сказати, що окрім розроблених форм мають місце універсальні функції роботи з даними.

Приведемо код функції експорту даних ExportXls довільної таблиці БД.

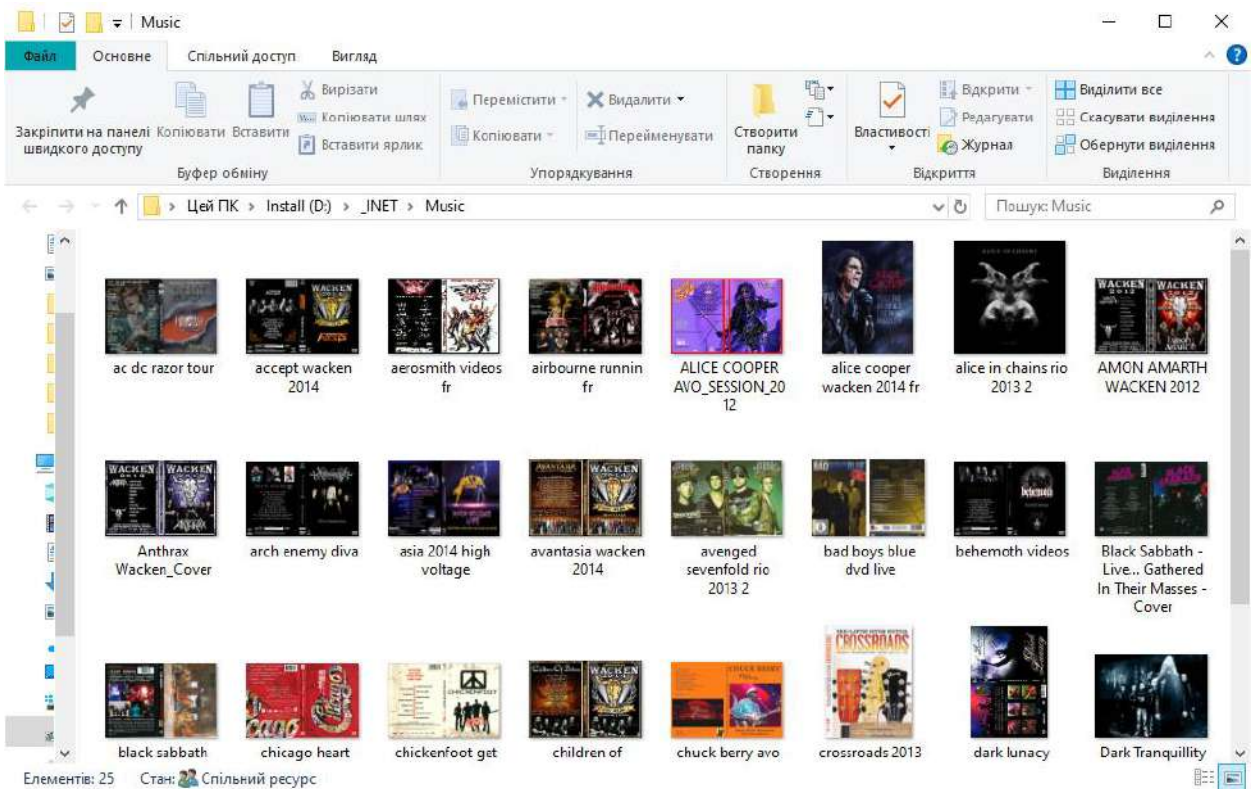


Рис. 4.10. Вибір картинки по диску виконавця

Код функції ExportXls.

```
//Універсальна функція експорту в Excel довільних даних
public static bool ExportXLS(DataView dt)
{
    try
    {
        Excel.Application ExcelApp = new Excel.Application();

        Excel.Workbook book = ExcelApp.Workbooks.Add(Type.Missing);
        ExcelApp.SheetsInNewWorkbook = 1;
        Excel.Worksheet sheet = (Excel.Worksheet)ExcelApp.Sheets.get_Item(1);
        sheet.Name = "Мій лист";

        Excel.Range range, range1;

        range = sheet.Cells;

        string str = (char)('A' + dt.Table.Columns.Count - 1) + "";

        //A1 - D1
        range1 = range.get_Range("A1", ((char)('A' + dt.Table.Columns.Count - 1))
+ "1");

        range1.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
        range1.Font.Bold = true;

        int row = 1, col;
        string v;

        Excel.Border border;
        border = range1.Borders.get_Item(Excel.XlBordersIndex.xlEdgeTop);
```

```

//Малювання заголовків стовпців таблиці
for (col = 0; col < dt.Table.Columns.Count; col++)
{
    v = dt.Table.Columns[col].ColumnName;
    if (v == "") continue;
    range.set_Item(row, col + 1, v);
}

row++;

foreach (DataRowView rr in dt)
{
    for (col = 0; col < dt.Table.Columns.Count; col++)
    {
        v = rr[col].ToString();
        if (v == "") continue;
        double dd;

        if (dt.Table.Columns[col].DataType.ToString() ==
"System.DateTime")
            v = Convert.ToDateTime(rr[col]).ToShortDateString();

        if (double.TryParse(v, out dd) == true)
            range.set_Item(row, col + 1, dd);
        else
            range.set_Item(row, col + 1, v);
    }

    row++;
}

range1 = range.get_Range("A1", ((char)('A' + dt.Table.Columns.Count - 1))
+ (dt.Count + 1).ToString());
range1.VerticalAlignment = Excel.XlVAlign.xlVAlignCenter;
range1.Borders.get_Item(Excel.XlBordersIndex.xlEdgeBottom).LineStyle = 1;
range1.Borders.get_Item(Excel.XlBordersIndex.xlEdgeLeft).LineStyle = 1;
range1.Borders.get_Item(Excel.XlBordersIndex.xlEdgeRight).LineStyle = 1;
range1.Borders.get_Item(Excel.XlBordersIndex.xlEdgeTop).LineStyle = 1;

range1.Borders.get_Item(Excel.XlBordersIndex.xlInsideHorizontal).LineStyle = 1;
range1.Borders.get_Item(Excel.XlBordersIndex.xlInsideVertical).LineStyle
= 1;

range1.Font.Name = Form1.font_text.Name;
range1.Font.Size = Form1.font_text.SizeInPoints;
range1.Font.Color = Form1.cvet_text;

range1 = range.get_Range("A1", ((char)('A' + dt.Table.Columns.Count - 1))
+ "1");
range1.Font.Color = Form1.cvet_hapka;
sheet.Columns.AutoFit();
ExcelApp.Visible = true;
}

catch (Exception)
{
    return false;
}

return true;
}

```

Як видно з функції єдиним параметром для передачі є об'єкт уявлення

таблиці, який має тип **DataView** та представлений властивістю таблиці **DefaultView**.

На наступному кроці продемонструємо універсальну функцію експорту до формату HTML, яка має назву `Export_Html`. Приведемо код даної функції.

```
//Універсальна функція експорту у HTML довільних даних
// ф-я експорту довільної таблиці у файл html.
// Вхід:
// filename - ім'я html-файлу в який відбувається експорт
// dt - представлення записів DataView
// zaglav - заголовок таблиці
// ff - шрифт
// cvetz - колір заголовку таблиці
// cveth - колір заголовків стовпців таблиці
// cvett - колір тексту комірок таблиці

public static bool Export_Html(string filename, DataView dt, string zaglav, Font
ff, Color cvetz, Color cveth, Color cvett)
{
    int i, kol_p;
    string vt;
    try
    {
        StreamWriter writer = new StreamWriter(filename, false, Encoding.UTF8);
        writer.Write("<html>\n");
        writer.Write("<head>\n");
        writer.Write("<title>");
        writer.Write("Експорт таблиці");
        writer.Write("</title>\n");
        writer.Write("<style>\n");
        writer.Write("table{border-style:solid; ");
        writer.Write("border-top-style:none;font-family:{0};font-size:{1}pt;",
ff.Name, (int)ff.SizeInPoints);
        //"{0:x2}{1:x2}{2:x2}"
        writer.Write("background-color:white;}\n");
        writer.Write("p.zag{color:");
        writer.Write("#{0:x2}{1:x2}{2:x2}; text-align:center; font-weight:bold;
font-family:{3}; font-size:{4}pt; color:#{0:x2}{1:x2}{2:x2};",
        cvetz.R, cvetz.G, cvetz.B, ff.Name, (int)ff.SizeInPoints);
        writer.Write("}\n");

        writer.Write("td{border-style:none;border-right-style:solid;border-top-
style:solid;border: solid windowtext 1px;}\n");
        writer.Write("td.text{color:");
        writer.Write("#{0:x2}{1:x2}{2:x2};", cvett.R, cvett.G, cvett.B);
        writer.Write("}\n");

        writer.Write("td.shapka{color:");
        writer.Write("#{0:x2}{1:x2}{2:x2}; font-weight:bold;", cveth.R, cveth.G,
cveth.B);
        writer.Write("}\n");

        //////////////////////////////////////////////////

        writer.Write("</style>\n");
        writer.Write("</head>\n");
        writer.Write("<body>\n");
        writer.Write("<center>\n");
        writer.Write("<p class=zag align=center>\n{0}\n</p>\n", zaglav);
```

```

        writer.Write("<table width=90% border=1 bordercolor=#000000 cellpadding=0
cellpadding=0 valign=center style='width:90.0%;border-
collapse:collapse;border:none;'>\n");
        kol_p = dt.Table.Columns.Count;
        writer.Write("<tr>\n");
        //Експорт заголовків стовпців таблиці
        for (i = 0; i < kol_p; i++)
        {
            vt = string.Format("    <td class=shapka width={0:F2}%
align=center>", 100.0 / kol_p);
            vt = vt.Replace(',', '.');
            writer.Write(vt);
            writer.Write(dt.Table.Columns[i].ColumnName);
            writer.Write("</td>\n");
        }
        writer.Write("</tr>\n");
        //Експорт даних
        foreach (DataRowView row in dt)
        {
            writer.Write("<tr>\n");
            for (i = 0; i < kol_p; i++)
            {
                writer.Write("    <td class=text align=center>");
                writer.Write("&nbsp;");
                vt = row[i].ToString();
                if (dt.Table.Columns[i].DataType.ToString() == "System.DateTime")
                    vt = Convert.ToDateTime(row[i]).ToShortDateString();
                if (vt != "")
                {
                    writer.Write(vt);
                    writer.Write("&nbsp;");
                }
                writer.Write("</td>\n");
            }
            writer.Write("</tr>\n");
        }
        writer.Write("</table>\n<br>\n");
        writer.Write("\n");
        writer.Write("</body>\n");
        writer.Write("</html>\n");
        writer.Close();
        return true;
    }
    catch (Exception)
    {
        return false;
    }
}

```

До даної функції передається ім'я файлу, в якому буде збережено дану web-сторінку (string filename), клас уявлення таблиці, яку треба експортувати (DataView dt), заголовок таблиці (string zaglav), а також настройки шрифту та кольорів виведення інформації (Font ff, Color cvetz, Color cveth, Color cvett). При цьому передбачено колір заголовку, колір шапки та колір виведення тексту таблиці.

Перевагою експорту в HTML є те, що сформований html-файл

відображається у власному середовищі за допомогою елементу керування WebBrowser.

На наступному етапі розглянемо безпосередньо автоматизацію продажу дисків студії звукозапису.

Спроекуємо форму для оформлення продажу Form6 (рис. 4.11).

Як видно з рис. 4.11. для оформлення продажу треба вибрати клієнта за його ідентифікатором, та відповідний диск, який він хоче придбати. За допомогою елементу MonthCalendar можна вибрати дату оформлення продажу. За умовчанням буде відображатись поточна дата. При натисканні на кнопку «Прослухати диск» можна прослухати відповідну музику, при натисканні на кнопку «Друк» – роздрукувати обкладинку для диску.

Оформлення продажу

Клієнт

Ідентифікатор  ПІБ

Диск

Прослухати диск Друк

Група  Диск

Рік випуску  Тип музики

Ціна:

Вибрати дату

березень 2025 р.						
Пн	Вт	Ср	Чт	Пт	Сб	Нд
24	25	26	27	28	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Сьогодні: 29.03.2025

Запис до БД Чек

Рис. 4.11. Проектування форми Form6 «Оформлення продажу дисків»

При натисканні на кнопку «Чек» у середовищі MS Excel у автоматизованому режимі буде сформовано чек, який можна роздрукувати. При натисканні кнопки «Запис до БД» інформація щодо продажу вибраного диску буде записана до бази даних до таблиці «Prodaga».

На рис. 4.12. наведемо автоматизоване формування чеку при натисканні на кнопку «Чек».

	A	B
1		
2	<b>Чек на купівлю диску</b>	
3		
4	<b>ПІБ клієнта</b>	<b>Лапшин</b>
5	<b>Ідентифікатор</b>	<b>АБ043223</b>
6	<b>Група</b>	<b>АВВА</b>
7	<b>Диск</b>	<b>The movie in concert (с переводом)</b>
8	<b>Тип музики</b>	<b>pop</b>
9	<b>Рік випуску</b>	<b>1980</b>
10	<b>Дата купівлі</b>	<b>29.03.2025</b>
11	<b>Ціна</b>	<b>25 грн.</b>
12	<b>Підпис</b>	
13	<b>Продавець</b>	<b>Вознюк Н.М.</b>
14		

Рис. 4.12. Формування чеку на покупку диску

Результат виконання форми «Оформлення продажу» приведено на рис. 4.13.

Оформлення продажу

Клієнт

Ідентифікатор АВ043223 ПІБ Лапшин

Диск

Прослухати диск Друк

Група АВВА Диск The movie in concert (с переводом)

Рік випуску 1980 Тип музики роп

Ціна: 25

Вибрати дату

березень 2025 р.

Пн	Вт	Ср	Чт	Пт	Сб	Нд
24	25	26	27	28	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Сьогодні: 29.03.2025

Запис до БД Чек

ALICE IN CHAINS

Рис. 4.13. Результат виконання форми «Оформлення продажу по дискам»

Так, у візуальному режимі клієнту надається повна інформація про диск та його виконавця з графічною ілюстрацією у вигляді файлу \*.jpg (елемент керування PictureBox).

Вся інформація про продаж дисків накопичується у базі даних, та при натисканні опції меню «Інформація про продажі» її можна переглянути із засобами застосування фільтрації по початковій та кінцевій датах.

За візуальне представлення інформації про продажі дисків відповідає клас Form7, результат виконання якого наведено на рис. 4.14.

Ідентифік.	ПІБ	Дата продажу	Ціна продажу	Група	Диск	Рік випуску	Тип музики
АБ043223	Лапшин	15.03.2024	25	ABBA	The movie...	1980	pop
АБ043223	Лапшин	27.05.2024	25	ABBA	The movie...	1980	pop
АБ043223	Лапшин	16.02.2025	54	ALICE IN ...	MTV Unpl...	1996	grunge,alt...
АБ043223	Лапшин	16.02.2025	38	ADRIANO ...	Uni popo- ...	1980	pop
АБ234790	Рамзаєв	05.03.2025	25	ABBA	The movie...	1980	pop
АБ234790	Рамзаєв	07.03.2025	27	AC/DC	Live at Ap...	1978	hard rock

Рис. 4.14. Результат виконання форми «Інформація про продажі дисків»

Як видно з форми, за допомогою елементів керування DateTimePicker можна вибрати діапазон дат та отримати необхідну інформацію про продажі та покупку дисків відповідними клієнтами.

Дану інформацію можна вивести до додатку MS Excel, а також до формату HTML, де потім роздрукувати у зручному режимі.

Важливим етапом при розробці будь-якого програмного забезпечення є робота з налаштуваннями. У нашій програмі за це відповідає меню «Налаштування». Через дане меню можна змінити кольори та шрифт звітної інформації.

Змінити шрифт можливо натиснувши пункт меню «Вибір шрифту». У відкритому діалозі зміни можливо здійснити всі необхідні зміни, що спростили б використання запропонованої програми (рис. 4.15).

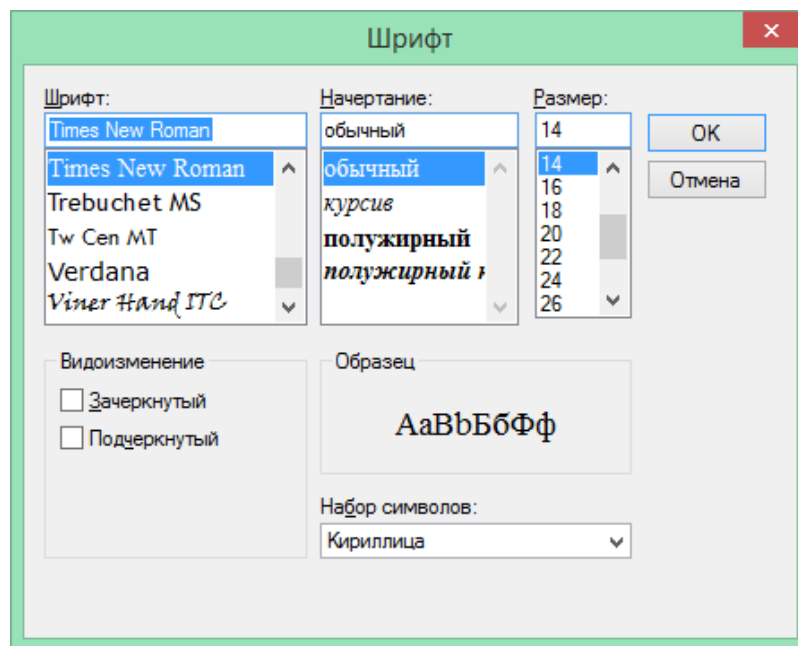


Рис. 4.15. Вікно вибору шрифту

Як видно з наданого малюнка, можливо змінити текст використовуючи курсив або виділити його жирним.

Для обрання необхідного кольору тексту, необхідно натиснути пункт меню «Вибір кольору» (рис. 4.16). Змінювати текст можливо також, використовуючи палітру, а не стандартні кольори.

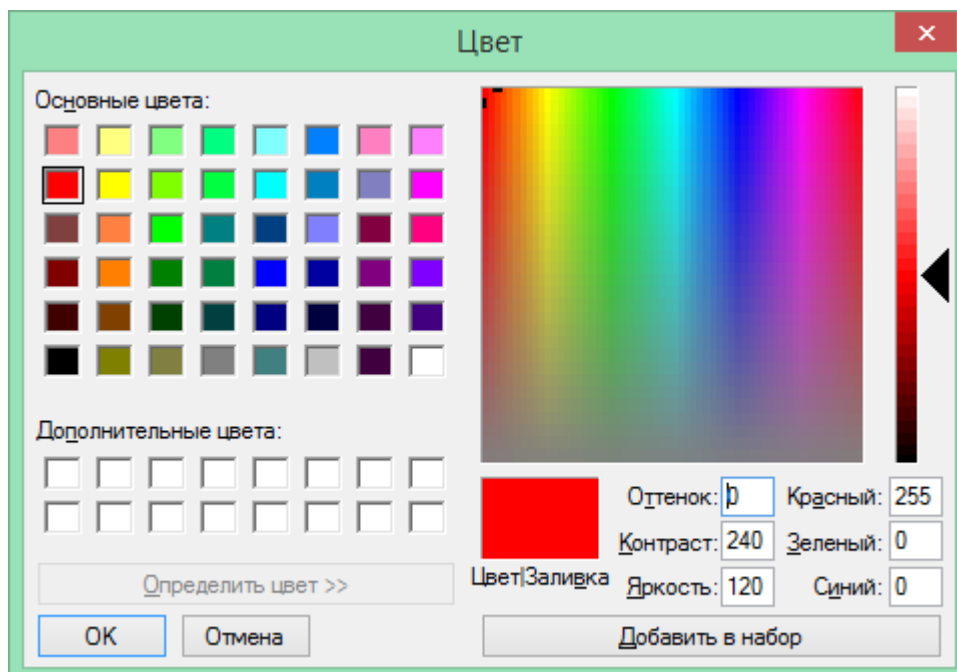


Рис. 4.16. Вікно вибору кольору

## 4.2. Опис програми організації продажу дисків «Diski\_3D»

За допомогою мови Visual C++ та використання бібліотек OpenGL було створене 3D зображення полігону та гістограми, для того щоб наочно проаналізувати ціни на альбоми тих чи інших виконавців. Була передбачена робота зі світлом, виведення тіні стовпчиків, виведення тексту, накладення текстур на землю, градієнтна заливка.

Наведемо фрагмент коду зчитування інформації з бази даних, використовуючи технологію ADO.

```
try
{
    pConn->Open((_bstr_t)str_con, L"", L"", 0);
    //Заповнення масивів та закриття набору даних
    pRecordset = pConn->Execute("Select Info, Cena From info_grup", 0,
adCmdText);
    FieldsPtr fields = pRecordset->Fields;
    pRecordset->MoveFirst();
    long i = 0;
    while (!pRecordset->ADO_EOF)
    {
        strcpy_s(name[i], (char*)(_bstr_t)fields->GetItem(0L)->GetValue());
        cena[i] = fields->GetItem(1L)->GetValue();
        pRecordset->MoveNext();
        i++;
    }

    count = i;

    pRecordset->Close();
    pRecordset.Release();

    //////////////////////////////////////
    //////////////////////////////////////

    pConn->Close();
    pConn.Release();
}
catch (_com_error &ce)
{
    ErrMessage(ce);
    if (pConn->GetState())pConn->Close();
    return;
}
```

На рис. 4.17. виведено полігон по ціні альбомів студії звукозапису.

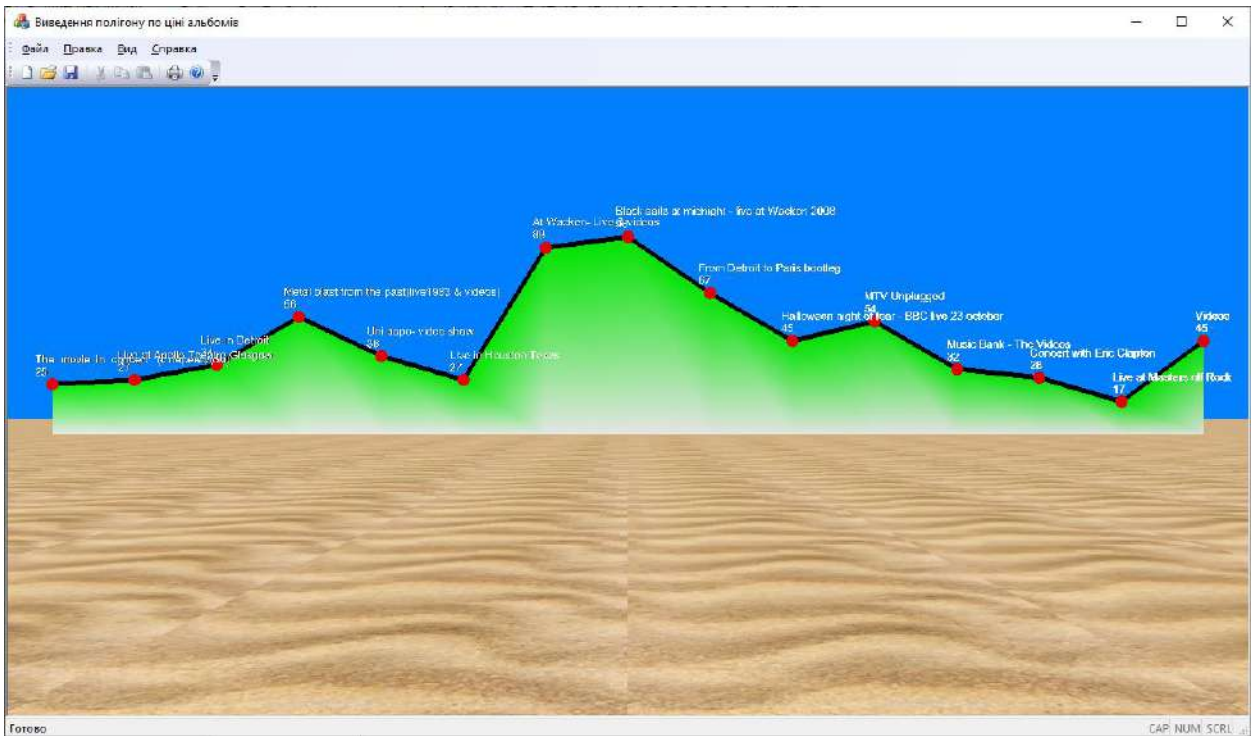


Рис. 4.17. Виведення полігону цін альбомів студії звукозапису

На рис. 4.18. виведено гістограму по ціні альбомів студії звукозапису.

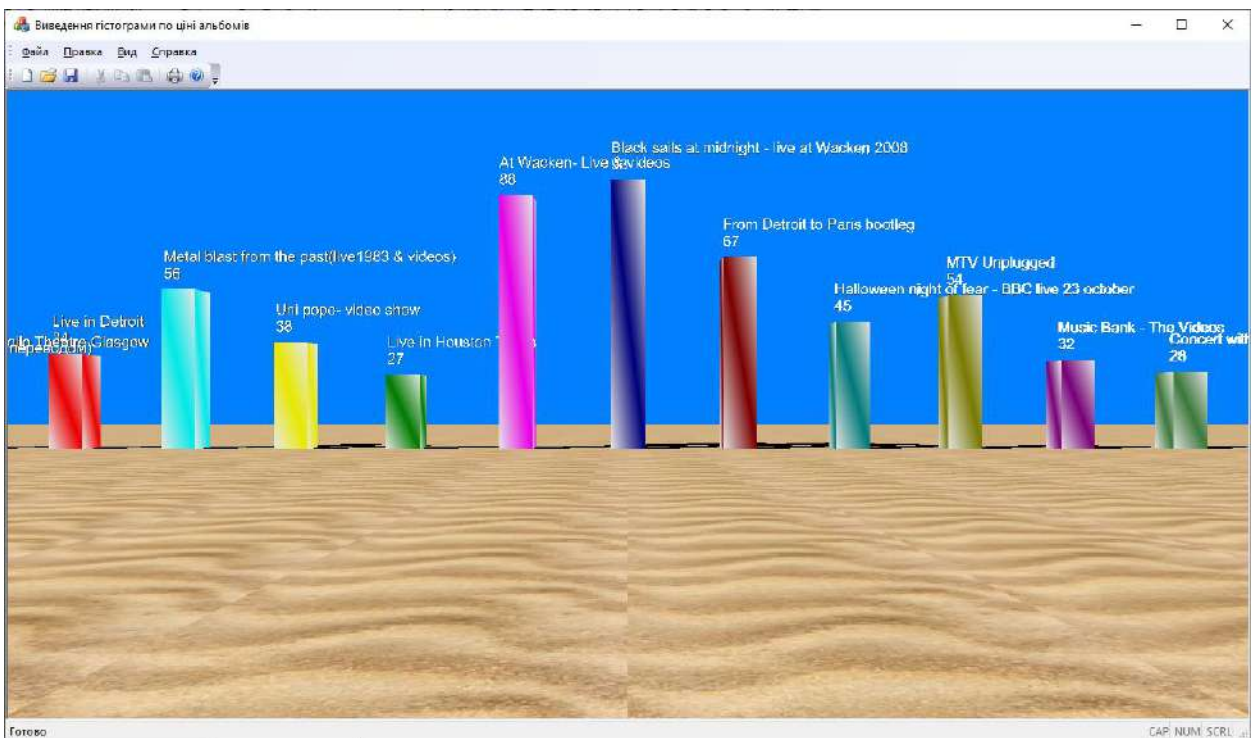


Рис. 4.18. Виведення гістограми цін альбомів студії звукозапису

### 4.3. Опис скриптів для формування діаграм у форматі HTML.

Як web-складову цього програмного забезпечення було обрано автоматичне генерування скриптів з використанням компонента керування chart.js. Для цього реалізовано універсальну функцію, яка приймає два масиви вхідних даних: числові значення, що визначають побудову діаграми, та рядкові значення, які використовуються для відображення підказок і легенди до стовпчиків.

Наведемо код даної функції:

```
//Універсальна функція скрипта для побудови діаграм в Web
public static void SaveHtmlFile(string fileName, string title, string chartLabel,
string[] countries, double[] population)
{
    // Формуємо HTML-контент
    string htmlContent = $"
<!DOCTYPE html>
<html lang='uk'>
<head>
    <meta charset='UTF-8'>
    <meta name='viewport' content='width=device-width, initial-scale=1.0'>
    <title>{title}</title>
    <script src='https://cdn.jsdelivr.net/npm/chart.js'></script>
    <style>
        body {{
            font-family: Arial, sans-serif;
            text-align: center;
            margin: 0;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            background-color: #f4f4f4;
        }}
        canvas {{
            width: 100vw !important;
            height: 100vh !important;
        }}
    </style>
</head>
<body>
    <canvas id='populationChart'></canvas>
    <script>
        const ctx = document.getElementById('populationChart').getContext('2d');
        new Chart(ctx, {{
            type: 'bar',
            data: {{
                labels: {GenerateArrayForJS(countries)},
                datasets: [{{
                    label: '{chartLabel}',
                    data: {GenerateArrayForJS(population)},
                    backgroundColor: 'rgba(75, 192, 192, 0.6)',
                    borderColor: 'rgba(75, 192, 192, 1)',
                    borderWidth: 1
                }}]
            }}
        }}
    </script>
</body>
</html>";
    File.WriteAllText(fileName, htmlContent);
}
```

```

    }},
    options: {{
        responsive: true,
        maintainAspectRatio: false,
        scales: {{
            y: {{
                beginAtZero: true
            }}
        }}
    }}
    }});
</script>
</body>
</html>";

// Записуємо HTML в файл
File.WriteAllText(fileName, htmlContent);
}

private static string GenerateArrayForJS(string[] array)
{
    return $"['{string.Join(", ", array)}']";
}

private static string GenerateArrayForJS(double[] array)
{
    return $"[{string.Join(", ", array)}]";
}

```

Дана функція є універсальною і дозволяє автоматично формувати JavaScript по даним які беруться з бази даних та відповідна інформація відразу відкривається з браузера. Це дозволяє користуватися статистикою проданих дисків не тільки з дек стоп-додатку, а і з мобільних пристроїв.

Гістограма в HTML статистики ціни за дисками студії звукозапису продемонстрована на рис. 4.19.

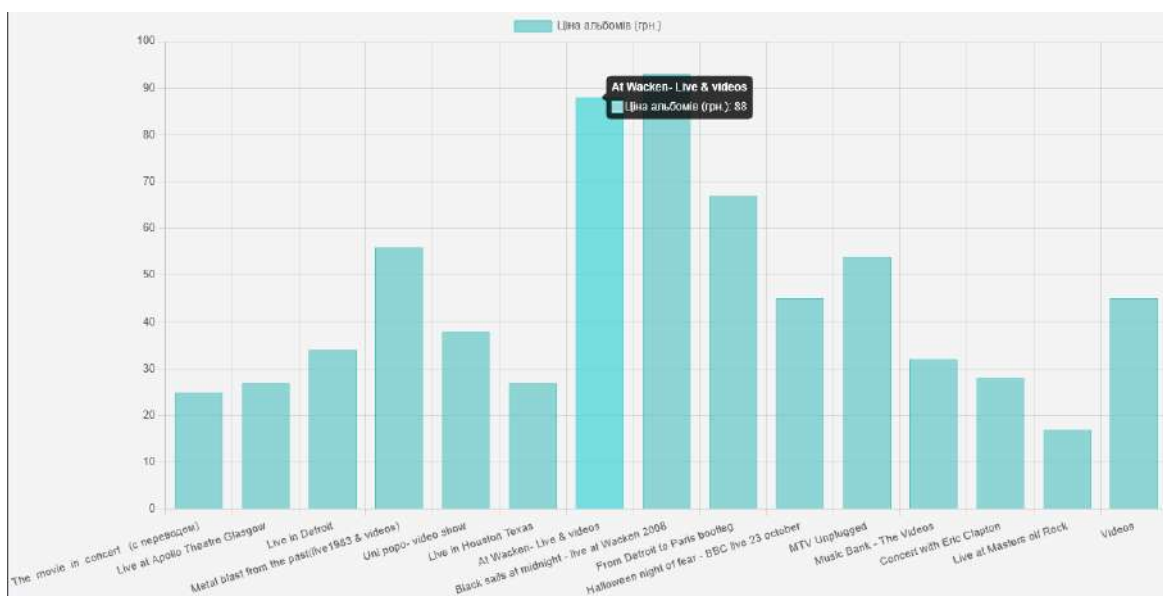


Рис. 4.19. Гістограма в HTML зміни цін по альбомам студії звукозапису

## Висновки до розділу 4

Отже у четвертому розділі продемонстроване розроблене програмне забезпечення продажу дисків студії звукозапису, яке складається з трьох частин: програма «Diski» розроблена на C#; програма «Diski\_3D» виводить статистику цін по альбомах дисків, використовуючи полігон та гістограму; у якості web-частини застосовується автоматичне формування Java-скриптів для формування статистичної інформації по дискам.

## ВИСНОВКИ

У ході виконання дипломної роботи були виконані наступні задачі:

1. Розроблено програмне забезпечення автоматизації продажу дисків студії звукозапису.
2. Використано основні класи універсальної технології ADO.Net для обробки баз даних.
3. Розроблено та створено візуальні класи Form1, Form2, Form3, Form4, Form5, Form6, Form7.
4. Розроблено універсальні функції експорту даних у MS Excel та HTML.

Відповідне програмне забезпечення дозволяє автоматизувати процес продажу дисків студії звукозапису та вирішити наступні задачі:

- ведення даних по дискам;
- ведення даних по клієнтам;
- організація продажу дисків, запис їх до бази даних та автоматичне формування чеків;
- автоматизоване формування звітів та їх експорт в MS Excel та HTML.

Програмний комплекс «Diski\_3D», розроблений на мові C++ з використанням відкритої графічної бібліотеки OpenGL, дає можливість наочного подання інформації, у вигляді графіків гістограми та полігону. За допомогою цих графіків надається можливість наочно аналізувати інформацію про зміни цін на альбоми тих чи інших виконавців.

Застосування універсальних Java-скриптів на основі елементу керування Chart.js дають можливість гнучко передавати дані з бази до створюваного html-файлу та користуватися ним в будь-яких браузерах. Це дозволило зробити гістограми в html для цін дисків різних виконавців.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Агуров, Павел С#. Сборник рецептов / Павел Агуров. - М.: "БХВ-Петербург", 2012. - 432 с.
2. Албахари, Джозеф С# 3.0. Справочник / Джозеф Албахари, Бен Албахари. - М.: БХВ-Петербург, 2012. - 944 с.
3. Албахари, Джозеф С# 3.0. Справочник / Джозеф Албахари, Бен Албахари. - М.: БХВ-Петербург, 2013. - 944 с.
4. Альфред, В. Ахо Компиляторы. Принципы, технологии и инструментарий / Альфред В. Ахо и др. - М.: Вильямс, 2015. - 266 с.
5. Бишоп, Дж. С# в кратком изложении / Дж. Бишоп, Н. Хорспул. - М.: Бином. Лаборатория знаний, 2013. - 472 с.
6. Вагнер, Билл С# Эффективное программирование / Билл Вагнер. - М.: ЛОРИ, 2013. - 320 с.
7. Зиборов, В.В. Visual С# 2012 на примерах / В.В. Зиборов. - М.: БХВ-Петербург, 2013. - 480 с.
8. Зиборов, Виктор Visual С# 2010 на примерах / Виктор Зиборов. - М.: "БХВ-Петербург", 2011. - 432 с.
9. Ишкова, Э. А. Самоучитель С#. Начала программирования / Э.А. Ишкова. - М.: Наука и техника, 2013. - 496 с.
10. Касаткин, А. И. Профессиональное программирование на языке си. Управление ресурсами / А.И. Касаткин. - М.: Высшая школа, 2012. - 432 с.
11. Лотка, Рокфорд С# и CSLA .NET Framework. Разработка бизнес-объектов / Рокфорд Лотка. - М.: Вильямс, 2010. - 816 с.
12. Мак-Дональд, Мэтью Silverlight 5 с примерами на С# для профессионалов / Мэтью Мак-Дональд. - М.: Вильямс, 2013. - 848 с.
13. Марченко, А. Л. Основы программирования на С# 2.0 / А.Л. Марченко. - М.: Интернет-университет информационных технологий, Бином. Лаборатория знаний, 2011. - 552 с.
14. Подбельский, В. В. Язык С#. Базовый курс / В.В. Подбельский. - М.:

Финансы и статистика, Инфра-М, 2011. - 384 с.

15. Прайс, Джейсон Visual C# 2.0. Полное руководство / Джейсон Прайс, Майк Гандэрлой. - М.: Век +, Корона-Век, Энтроп, 2010. - 736 с.

16. Рихтер, Джеффри CLR via C#. Программирование на платформе Microsoft .NET Framework 4.0 на языке C# / Джеффри Рихтер. - М.: Питер, 2013. - 928 с.

17. Смоленцев, Н. К. MATLAB. Программирование на Visual C#, Borland JBuilder, VBA (+ CD-ROM) / Н.К. Смоленцев. - М.: ДМК Пресс, 2011. - 456 с.

18. Троелсен, Эндрю Язык программирования C# 5.0 и платформа .NET 4.5 / Эндрю Троелсен. - М.: Вильямс, 2015. - 486 с.

19. Троелсен, Эндрю Язык программирования C# 2008 и платформа .NET 3.5 / Эндрю Троелсен. - М.: Вильямс, 2010. - 370 с.

20. Фримен, Адам ASP.NET MVC 3 Framework с примерами на C# для профессионалов / Адам Фримен , Стивен Сандерсон. - М.: Вильямс, 2011. - 672 с.

21. Дж.Рихтер. CLR via C# . Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4-е изд./Дж.Рихтер – Питер, 2013. - 896с.

22. Дж.Рихтер. CLR via C#. Программирование на платформе Microsoft .NET Framework 2.0 на языке C#, 2-е изд/ Дж.Рихтер - Питер, 2007. - 656 с.

23. Зеленков Ю. Введение в базы данных / Ю. Зеленков – СПб.: Питер, 2003

24. Кузнецов С.Д. Основы современных баз данных. 2-е издание испр. / С.Д.Кузнецов,– М.:Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний– 2007. – 484с.

# ДОДАТКИ

## Текст класу головної форми програми «Form1»

```

using System;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.Data.OleDb;
using System.Drawing;
using System.Text;
using System.IO;
using System.Windows.Forms;
using System.Collections.Generic;
using Excel = Microsoft.Office.Interop.Excel;

namespace Diski
{
    public partial class Form1 : Form
    {
        //Підключення до БД
        OleDbConnection conn;

        //Налаштування
        public static Color cvet_zag;
        public static Color cvet_hapka;
        public static Color cvet_text;
        public static Font font_text;

        int kod_zap = 0;

        public Form1()
        {
            InitializeComponent();
            //Формування об'єкту Connection
            try
            {
                string source = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=diski.mdb";
                conn = new OleDbConnection(source);
                conn.Open();
            }

            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
                return;
            }

            //Зчитування налаштувань
            try
            {
                BinaryReader bb = new BinaryReader(new FileStream("disk.ini",
                FileMode.Open));
                cvet_zag = Color.FromArgb(bb.ReadByte(), bb.ReadByte(), bb.ReadByte());
                cvet_hapka = Color.FromArgb(bb.ReadByte(), bb.ReadByte(), bb.ReadByte());
                cvet_text = Color.FromArgb(bb.ReadByte(), bb.ReadByte(), bb.ReadByte());
                font_text = new Font(bb.ReadString(), bb.ReadSingle(),
                (FontStyle)bb.ReadInt32());
                bb.Close();
                return;
            }
            catch

```

## Продовження додатку А

```

    {
        MessageBox.Show("Файл disk.ini не знайдено!\nБудуть прийняті значення за
умовчанням",
            "Увага!", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }

    cvet_zag = Color.Red;
    cvet_hapka = Color.Blue;
    cvet_text = Color.Black;
    font_text = new Font("Times New Roman", 14, FontStyle.Regular);
    kod_zap = 1;
}

//Довідник по групам
private void групиToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form2 ff = new Form2();
    ff.conn = conn;
    ff.kod_table = 1;
    ff.ShowDialog();
}

//Інформація по дискам
private void информацияОдискахToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form3 ff = new Form3();
    ff.conn = conn;
    ff.ShowDialog();
}

//Інформація по клієнтах (довідник)
private void клиентыToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form2 ff = new Form2();
    ff.conn = conn;
    ff.kod_table = 2;
    ff.ShowDialog();
}

// Оформлення продажу
private void оформлениеПродажиToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form6 ff = new Form6();
    ff.conn = conn;
    ff.ShowDialog();
}

// Інформація про продажі
private void информацияОПродажахToolStripMenuItem_Click(object sender, EventArgs
e)
{
    Form7 ff = new Form7();
    ff.conn = conn;
    ff.ShowDialog();
}

//Вибір шрифту
private void выборШрифтаToolStripMenuItem_Click(object sender, EventArgs e)
{

```

## Продовження додатку А

```

fontDialog1.Font = font_text;
if (fontDialog1.ShowDialog() == DialogResult.OK)
{
    font_text = fontDialog1.Font;
    kod_zap = 1;
}
}

//Вибір кольорів
private void цветЗаголовкаТаблицыToolStripMenuItem_Click(object sender, EventArgs
e)
{
    colorDialog1.Color = cvet_zag;
    if (colorDialog1.ShowDialog() == DialogResult.OK)
    {
        cvet_zag = colorDialog1.Color;
        kod_zap = 1;
    }
}

private void цветШапкиТаблицыToolStripMenuItem_Click(object sender, EventArgs e)
{
    colorDialog1.Color = cvet_hapka;
    if (colorDialog1.ShowDialog() == DialogResult.OK)
    {
        cvet_hapka = colorDialog1.Color;
        kod_zap = 1;
    }
}

private void цветТекстаТаблицыToolStripMenuItem_Click(object sender, EventArgs e)
{
    colorDialog1.Color = cvet_text;
    if (colorDialog1.ShowDialog() == DialogResult.OK)
    {
        cvet_text = colorDialog1.Color;
        kod_zap = 1;
    }
}

//Формування діаграми в HTML
private void діаграмаВHTMLToolStripMenuItem_Click(object sender, EventArgs e)
{
    //Формування масивів для побудови діаграми
    List<string> infoList = new List<string>();
    List<double> cenaList = new List<double>();

    // SQL запит
    string query = "SELECT Info, Cena FROM info_grup";
    OleDbCommand command = new OleDbCommand(query, conn);

    // Виконання запиту
    OleDbDataReader reader = command.ExecuteReader();

    while (reader.Read())
    {
        // Додаємо дані до списків
        infoList.Add(reader["Info"].ToString());
        cenaList.Add(Convert.ToDouble(reader["Cena"]));
    }
    reader.Close();
}

```

## Продовження додатку А

```

command.Dispose();

//Виклик скрипту для формування HTML
SaveHtmlFile("Diagram.htm", "Гістограма ціни альбомів",
    "Ціна альбомів (грн.)", infoList.ToArray(), cenaList.ToArray());

// Запускаємо браузер за умовчанням для відкриття HTML файлу
Process.Start(new ProcessStartInfo("Diagram.htm") { UseShellExecute = true
});
}

//Інформація про програму
private void опПрограммеToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form4 ff = new Form4();
    ff.ShowDialog();
}

//Закриття форми
protected override void OnClosing(CancelEventArgs e)
{
    if (kod_zap == 1)
    {
        DialogResult dr = MessageBox.Show("Налаштування були змінені!\nЗберегти
зміни на диск?", "Зберігання!",
        MessageBoxButtons.YesNoCancel, MessageBoxIcon.Question);
        if (dr == DialogResult.Yes)
        {
            BinaryWriter bb = new BinaryWriter(new FileStream("disk.ini",
FileMode.OpenOrCreate));
            bb.Write(cvet_zag.R);
            bb.Write(cvet_zag.G);
            bb.Write(cvet_zag.B);

            bb.Write(cvet_hapka.R);
            bb.Write(cvet_hapka.G);
            bb.Write(cvet_hapka.B);

            bb.Write(cvet_text.R);
            bb.Write(cvet_text.G);
            bb.Write(cvet_text.B);

            bb.Write(font_text.Name);
            bb.Write(font_text.SizeInPoints);
            bb.Write((int)font_text.Style);
            bb.Close();
            return;
        }
        else if (dr == DialogResult.Cancel)
            e.Cancel = true;
    }

    base.OnClosing(e);
}

//Універсальна функція скрипта для побудови діаграм в Web

```

## Продовження додатку А

```

    public static void SaveHtmlFile(string fileName, string title, string chartLabel,
string[] countries, double[] population)
    {
        // Формуємо HTML-контент
        string htmlContent = $"
<!DOCTYPE html>
<html lang='uk'>
<head>
    <meta charset='UTF-8'>
    <meta name='viewport' content='width=device-width, initial-scale=1.0'>
    <title>{title}</title>
    <script src='https://cdn.jsdelivr.net/npm/chart.js'></script>
    <style>
        body {{
            font-family: Arial, sans-serif;
            text-align: center;
            margin: 0;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            background-color: #f4f4f4;
        }}
        canvas {{
            width: 100vw !important;
            height: 100vh !important;
        }}
    </style>
</head>
<body>
    <canvas id='populationChart'></canvas>
    <script>
        const ctx = document.getElementById('populationChart').getContext('2d');
        new Chart(ctx, {{
            type: 'bar',
            data: {{
                labels: {GenerateArrayForJS(countries)},
                datasets: [{{
                    label: '{chartLabel}',
                    data: {GenerateArrayForJS(population)},
                    backgroundColor: 'rgba(75, 192, 192, 0.6)',
                    borderColor: 'rgba(75, 192, 192, 1)',
                    borderWidth: 1
                }}]
            }},
            options: {{
                responsive: true,
                maintainAspectRatio: false,
                scales: {{
                    y: {{
                        beginAtZero: true
                    }}
                }}
            }}
        }});
    </script>
</body>
</html>";

        // Записуємо HTML в файл
        File.WriteAllText(fileName, htmlContent);
    }

```

## Продовження додатку А

```

private static string GenerateArrayForJS(string[] array)
{
    return $"[{string.Join(", ", array)}]";
}

private static string GenerateArrayForJS(double[] array)
{
    return $"[{string.Join(", ", array)}]";
}

//Універсальна функція експорту в Excel довільних даних
public static bool ExportXLS(DataView dt)
{
    try
    {
        Excel.Application ExcelApp = new Excel.Application();

        Excel.Workbook book = ExcelApp.Workbooks.Add(Type.Missing);
        ExcelApp.SheetsInNewWorkbook = 1;
        Excel.Worksheet sheet = (Excel.Worksheet)ExcelApp.Sheets.get_Item(1);
        sheet.Name = "Мій лист";

        Excel.Range range, range1;

        range = sheet.Cells;

        string str = (char)('A' + dt.Table.Columns.Count - 1) + "";

        //A1 - D1
        range1 = range.get_Range("A1", ((char)('A' + dt.Table.Columns.Count - 1))
+ "1");

        range1.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
        range1.Font.Bold = true;

        int row = 1, col;
        string v;

        Excel.Border border;
        border = range1.Borders.get_Item(Excel.XlBordersIndex.xlEdgeTop);

        //Малювання заголовків стовпців таблиці
        for (col = 0; col < dt.Table.Columns.Count; col++)
        {
            v = dt.Table.Columns[col].ColumnName;
            if (v == "") continue;
            range.set_Item(row, col + 1, v);
        }

        row++;

        foreach (DataRowView rr in dt)
        {
            for (col = 0; col < dt.Table.Columns.Count; col++)
            {
                v = rr[col].ToString();
                if (v == "") continue;
                double dd;

                if (dt.Table.Columns[col].DataType.ToString() ==
"System.DateTime")
                    v = Convert.ToDateTime(rr[col]).ToShortDateString();
            }
        }
    }
}

```

## Продовження додатку А

```

        if (double.TryParse(v, out dd) == true)
            range.set_Item(row, col + 1, dd);
        else
            range.set_Item(row, col + 1, v);
    }

    row++;
}

range1 = range.get_Range("A1", ((char)('A' + dt.Table.Columns.Count - 1))
+ (dt.Count + 1).ToString());
range1.VerticalAlignment = Excel.XlVAlign.xlVAlignCenter;
range1.Borders.get_Item(Excel.XlBordersIndex.xlEdgeBottom).LineStyle = 1;
range1.Borders.get_Item(Excel.XlBordersIndex.xlEdgeLeft).LineStyle = 1;
range1.Borders.get_Item(Excel.XlBordersIndex.xlEdgeRight).LineStyle = 1;
range1.Borders.get_Item(Excel.XlBordersIndex.xlEdgeTop).LineStyle = 1;

range1.Borders.get_Item(Excel.XlBordersIndex.xlInsideHorizontal).LineStyle = 1;
range1.Borders.get_Item(Excel.XlBordersIndex.xlInsideVertical).LineStyle
= 1;

range1.Font.Name = Form1.font_text.Name;
range1.Font.Size = Form1.font_text.SizeInPoints;
range1.Font.Color = Form1.cvet_text;

range1 = range.get_Range("A1", ((char)('A' + dt.Table.Columns.Count - 1))
+ "1");

range1.Font.Color = Form1.cvet_hapka;

sheet.Columns.AutoFit();
ExcelApp.Visible = true;

}

catch (Exception)
{
    return false;
}

return true;
}

//Універсальна функція експорту у HTML довільних даних
// ф-я експорту довільної таблиці у файл html.
// Вхід:
// filename - ім'я html-файлу в який відбувається експорт
// dt - представлення записів DataView
// zaglav - заголовки таблиці
// ff - шрифт
// cvetz - колір заголовку таблиці
// cveth - колір заголовків стовпців таблиці
// cvett - колір тексту комірок таблиці

public static bool Export_Html(string filename, DataView dt, string zaglav, Font
ff, Color cvetz, Color cveth, Color cvett)
{

    int i, kol_p;
    string vt;

    try
    {

```

## Продовження додатку А

```

StreamWriter writer = new StreamWriter(filename, false, Encoding.UTF8);
writer.Write("<html>\n");
writer.Write("<head>\n");
writer.Write("<title>");
writer.Write("Експорт таблиці");
writer.Write("</title>\n");
writer.Write("<style>\n");

        writer.Write("table{border-style:solid; ");
        writer.Write("border-top-style:none;font-family:{0};font-size:{1}pt;",
ff.Name, (int)ff.SizeInPoints);

        //"{0:x2}{1:x2}{2:x2}"
        writer.Write("background-color:white;}\n.p.zag{color:");
        writer.Write("#{0:x2}{1:x2}{2:x2}; text-align:center; font-weight:bold;
font-family:{3}; font-size:{4}pt; color:#{0:x2}{1:x2}{2:x2};",
        cvetz.R, cvetz.G, cvetz.B, ff.Name, (int)ff.SizeInPoints);
        writer.Write("}\n");

        writer.Write("td{border-style:none;border-right-style:solid;border-top-
style:solid;border: solid windowtext 1px;}\n");
        writer.Write("td.text{color:");
        writer.Write("#{0:x2}{1:x2}{2:x2};", cvett.R, cvett.G, cvett.B);
        writer.Write("}\n");

        writer.Write("td.shapka{color:");
        writer.Write("#{0:x2}{1:x2}{2:x2}; font-weight:bold;", cveth.R, cveth.G,
cveth.B);
        writer.Write("}\n");

        //////////////////////////////////

        writer.Write("</style>\n");
        writer.Write("</head>\n");
        writer.Write("<body>\n");

        writer.Write("<center>\n");
        writer.Write("<p class=zag align=center>\n{0}\n</p>\n", zaglav);

        writer.Write("<table width=90% border=1 bordercolor=#000000 cellspacing=0
cellpadding=0 valign=center style='width:90.0%;border-
collapse:collapse;border:none;'>\n");

        kol_p = dt.Table.Columns.Count;

        writer.Write("<tr>\n");

        //Експорт заголовків стовпців таблиці
        for (i = 0; i < kol_p; i++)
        {
            vt = string.Format("    <td class=shapka width={0:F2}%
align=center>", 100.0 / kol_p);
            vt = vt.Replace(',', '.');
            writer.Write(vt);
            writer.Write(dt.Table.Columns[i].ColumnName);
            writer.Write("</td>\n");
        }

        writer.Write("</tr>\n");

        //Експорт даних
        foreach (DataRowView row in dt)

```

## Продовження додатку А

```

    {
        writer.Write("<tr>\n");
        for (i = 0; i < kol_p; i++)
        {
            writer.Write("    <td class=text align=center>");
            writer.Write("&nbsp;");

            vt = row[i].ToString();

            if (dt.Table.Columns[i].DataType.ToString() == "System.DateTime")
                vt = Convert.ToDateTime(row[i]).ToShortDateString();

            if (vt != "")
            {
                writer.Write(vt);
                writer.Write("&nbsp;");
            }
            writer.Write("</td>\n");
        }

        writer.Write("</tr>\n");
    }

    writer.Write("</table>\n<br>\n");

    writer.Write("\n");
    writer.Write("</body>\n");
    writer.Write("</html>\n");

    writer.Close();
    return true;
}
catch (Exception)
{
    return false;
}
}
}

//Клас для заповнення списку
public class Name_List
{
    public string str;
    public int ItemData;

    public Name_List(string str, int ItemData)
    {
        this.str = str;
        this.ItemData = ItemData;
    }

    public override string ToString()
    {
        return str;
    }
}
}
}

```

Текст класів роботи з довідниками «Form2»  
та ведення інформації по групам «Form3»

```

using System;
using System.Data;
using System.Data.Common;
using System.Data.OleDb;
using System.Windows.Forms;

namespace Diski
{
    public partial class Form2 : Form
    {
        public OleDbConnection conn;
        OleDbDataAdapter da;
        OleDbCommandBuilder bulder;
        DataSet ds;

        public int kod_table=1;

        public Form2()
        {
            InitializeComponent();
        }

        //Завантаження форми
        private void Form2_Load(object sender, EventArgs e)
        {
            try
            {
                if (kod_table == 1)
                {
                    Text = "Довідник по групам";
                    label1.Text = "Інформація по музичним групам (виконавцям)";

                    da = new OleDbDataAdapter("Select ID, name from sprav_grup", conn);
                    bulder = new OleDbCommandBuilder(da);
                    ds = new DataSet();

                    DataTableMapping datam = da.TableMappings.Add("Table", "Групи");
                    datam.ColumnMappings.Add("ID", "№");
                    datam.ColumnMappings.Add("name", "Група (виконавець)");
                }

                else
                {
                    Text = "Довідник по клієнтам";
                    label1.Text = "Довідник по клієнтам";

                    da = new OleDbDataAdapter("Select Pasport, FIO, Pol, Voz from
Klient", conn);
                    bulder = new OleDbCommandBuilder(da);
                    ds = new DataSet();

                    DataTableMapping datam = da.TableMappings.Add("Table", "Клієнти");
                    datam.ColumnMappings.Add("Pasport", "Ідентифікатор");
                    datam.ColumnMappings.Add("FIO", "ПІБ");
                }
            }
            catch { }
        }
    }
}

```

## Продовження додатку Б

```

        datam.ColumnMappings.Add("Pol", "Стать");
        datam.ColumnMappings.Add("Voz", "Вік");

    }

    da.Fill(ds);

    bindingSource1.DataSource = ds.Tables[0];
    dataGridView1.DataSource = bindingSource1;
    bindingNavigator1.BindingSource = bindingSource1;
    if (kod_table == 1)
    {
        dataGridView1.Columns[0].Visible = false;
        dataGridView1.Columns[1].Width = 350;
    }
}

catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}

}

//Оновлення даних
private void toolStripButton1_Click(object sender, EventArgs e)
{
    try
    {
        da.Update(ds.Tables[0]);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

//Експорт даних в Excel
private void toolStripButton2_Click(object sender, EventArgs e)
{
    if (!Form1.ExportXLS(ds.Tables[0].DefaultView))
    {
        MessageBox.Show("Неможливий експорт в Excel!",
            "Увага!", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

//Експорт даних в HTML
private void toolStripButton3_Click(object sender, EventArgs e)
{
    if (!Form1.Export_Html(ds.Tables[0].TableName + ".htm",
        ds.Tables[0].DefaultView, ds.Tables[0].TableName,
        Form1.font_text, Form1.cvet_zag, Form1.cvet_hapka, Form1.cvet_text))
    {
        MessageBox.Show("Неможливий експорт в HTML!",
            "Увага!", MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
    Form5 ff = new Form5();
    ff.webBrowser1.Navigate(Application.StartupPath + "\\\" +
        ds.Tables[0].TableName + ".htm");
    ff.ShowDialog();
}
}
}
}

```

## Продовження додатку Б

```

using System;
using System.Data;
using System.Data.Common;
using System.Data.OleDb;
using System.Drawing;
using System.IO;
using System.Windows.Forms;

namespace Diski
{
    public partial class Form3 : Form
    {
        public OleDbConnection conn;
        OleDbCommand cmd;
        OleDbDataReader datar;
        OleDbDataAdapter da;
        OleDbCommandBuilder bulder;
        DataSet ds;

        public Form3()
        {
            InitializeComponent();
        }

        //Завантаження форми
        private void Form3_Load(object sender, EventArgs e)
        {
            try
            {
                // Заповнення списку
                cmd = new OleDbCommand("Select ID, name From sprav_grup", conn);
                datar = cmd.ExecuteReader();
                while (datar.Read())
                    comboBox1.Items.Add(new Name_List(datar[1].ToString(),
Convert.ToInt32(datar[0])));

                comboBox1.SelectedIndex = 0;
                datar.Close();
                cmd.Dispose();
                ///////////////////////////////////
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

        //Вибір групи(виконавця)
        private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
        {
            try
            {
                string str = "Select ID,Info,Years,Tip,Kart,Cena from info_grup where ";
                str += "ID_grup = " + ((Name_List)comboBox1.SelectedItem).ItemData;
                da = new OleDbDataAdapter(str, conn);
                bulder = new OleDbCommandBuilder(da);
                ds = new DataSet();
            }
        }
    }
}

```

## Продовження додатку Б

```

DataTableMapping datam = da.TableMappings.Add("Table", "Диск");
datam.ColumnMappings.Add("ID", "№");
datam.ColumnMappings.Add("Info", "Інформація про диск");
datam.ColumnMappings.Add("Years", "Рік випуску");
datam.ColumnMappings.Add("Tip", "Тип музики");
datam.ColumnMappings.Add("Cena", "Ціна");
datam.ColumnMappings.Add("Kart", "Зображення");

da.Fill(ds);

bindingSource1.DataSource = ds.Tables[0];
dataGridView1.DataSource = bindingSource1;
bindingNavigator1.BindingSource = bindingSource1;
dataGridView1.Columns[0].Visible = false;
dataGridView1.Columns[4].Visible = false;
dataGridView1.Columns[1].Width = 350;
bindingSource1.Position = 0;
pictureBox1.Image = Properties.Resources.No;
bindingSource1_PositionChanged(null, null);

}

catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}

}

//Додавання зображення
private void button1_Click(object sender, EventArgs e)
{
    if (bindingSource1.Position < 0) return;

    openFileDialog1.Title = "Виберіть зображення у папці Music";
    string path = Path.GetDirectoryName(Application.ExecutablePath) +
@"\Music";
    openFileDialog1.InitialDirectory = path;

    // початкова назва файлу
    openFileDialog1.FileName = "";
    // фільтр
    openFileDialog1.Filter = "Зображення (*.jpg)|*.jpg";
    if (openFileDialog1.ShowDialog() != DialogResult.OK) return;

    pictureBox1.Image = Image.FromFile (openFileDialog1.FileName);

    //MessageBox.Show(openFileDialog1.SafeFileName);

    ds.Tables[0].Rows[bindingSource1.Position][4] =
openFileDialog1.SafeFileName;
    toolStripButton1_Click(null, null);

}

//Оновлення даних
private void toolStripButton1_Click(object sender, EventArgs e)
{
    try
    {
        da.Update(ds.Tables[0]);
    }
}

```

## Продовження додатку Б

```

    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

//Зміна позиції
private void bindingSource1_PositionChanged(object sender, EventArgs e)
{
    if(bindingSource1.Position<0)return;

    if (ds.Tables[0].Rows[bindingSource1.Position][4].ToString() != "")
    {
        try
        {
            string path = Path.GetDirectoryName(Application.ExecutablePath) +
@"\Music";
            pictureBox1.Image = Image.FromFile(path + "\\\" +
ds.Tables[0].Rows[bindingSource1.Position][4].ToString());
        }
        catch (Exception)
        {
            pictureBox1.Image = Properties.Resources.No;
        }

    }
    else
        pictureBox1.Image = Properties.Resources.No;
}

//Експорт в XLS
private void toolStripButton2_Click(object sender, EventArgs e)
{
    if (!Form1.ExportXLS(ds.Tables[0].DefaultView))
    {
        MessageBox.Show("Неможливий експорт в Excel!",
            "Увага!", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

//Експорт в HTML
private void toolStripButton3_Click(object sender, EventArgs e)
{
    if (!Form1.Export_Html(ds.Tables[0].TableName + ".htm",
ds.Tables[0].DefaultView, ds.Tables[0].TableName,
    Form1.font_text, Form1.cvet_zag, Form1.cvet_hapka, Form1.cvet_text))
    {
        MessageBox.Show("Неможливий експорт в HTML!",
            "Увага!", MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }

    Form5 ff = new Form5();
    ff.webBrowser1.Navigate(Application.StartupPath + "\\\" +
ds.Tables[0].TableName + ".htm");
    ff.ShowDialog();
}
}
}
}

```