Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра Програмної інженерії

КУРСОВА РОБОТА

ПОЯСНЮВАЛЬНА ЗАПИСКА

з дисципліні “ Об’єктно -орієнтоване програмування”

КАРТОТЕКА ІНТЕРПОЛУ

Керівник, професор Бондарєв В. М.

Студент гр. ПЗПІ-19-4 Керецман І.А.

Комісія:

Старший викладач \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Черепанова Ю.Ю.

Профксор \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Бондарєв В. М.

Доцент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Побіженко І. О.

Харків 2020

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ**

Кафедра: ***Програмної інженерії***

Дисципліна: ***Об’єктно-орієнтоване програмування***

Спеціальність: ***121***  ***Інженерія програмного забезпечення***

Освітня програма: ***Програмна інженерія***

Курс 1 .Група *ПЗПІ-19 -4* . Семестр *2* .

***ЗАВДАННЯ***

***на курсовий проект студента***

*Керецмана Іллі Андрійовича \_*

(Прізвище, Ім'я, По батькові)

1 Тема проекту: Картотека інтерполу

2 Термін здачі студентом закінченого проекту: ***“19” - червня - 2020 р***.

3 Вихідні дані до проекту:

*Специфікація програми, методичні вказівки до виконання курсової роботи*

4 Зміст розрахунково-пояснювальної записки:

*Вступ,* *специфікація програми, проектна специфікація, інструкція користувача, висновки\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5 Перелік графічного матеріалу:

\_\_ *Схема об’єктної моделі, діаграма класів, приклади екранних форм\_\_\_\_\_* \_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |
| --- | --- | --- |
| *№* | *Назва етапу* | *Термін виконання* |
| 1 | Видача теми, узгодження і затвердження теми | 21-02-2020 р. |
| 2 | Формулювання вимог до програми | 25-02-2020 – 31-03-2020 р. |
| 3 | Розробка підсистеми зберігання та пошуку картотеки інтерполу. | 31-03-2020 – 10-04-2020 р. |
| 4 | Розробка підсистеми зберігання та пошуку злочинців. | 10-04-2020 – 21-04-2020 р. |
| 5 | Розробка підсистеми зберігання та пошуку злочинних угруповань. | 21-04-2020 – 02-05-2020 р. |
| 6 | Розробка функцій зберігання та завантаження даних | 02-05-2020 – 10-05-2020 р. |
| 7 | Тестування і доопрацювання розробленої програмної системи. | 10-05-2020 – 15-05-2020 р. |
| 8 | Оформлення пояснювальної записки, додатків, графічного матеріалу | 15-05-2020 – 05-06-2020 р. |
| 9 | Захист | 01-06-2020 – 19-06-2020 р. |

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Керівник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Бондарєв В. М.\_\_\_

(Прізвище, Ім'я, По батькові)

« 21 »\_лютого\_\_\_\_\_\_\_\_\_\_ 2020 р.

**РЕФЕРАТ**

Пояснювальна записка до курсової роботи: 48 с., 26 рис, 1 додаток, 5 джерел.

КЛАС, МОВА ПРОГРАМУВАННЯ C#, ЗЛОЧИНЕЦЬ, ОБ’ЕКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ, ПРОГРАМА, ПЛАТФОРМА .NET, СПИСОК, КОЛЕКЦІЯ, ЗЛОЧИННЕ УГРУПОВАННЯ, АРХІВ.

Метою роботи є розробка програми “Картотека інтерполу” на засадах об'єктно-орієнтованого програмування.

Методи розробки базуються на використанні середи розробки Microsoft Visual Studio 2015, Windows Forms, платформи .NET Framework 4.5, мови програмування C#.

В результаті отримана програма під назвою “Довідник покупця”, яка дозволяє зберігати список магазинів, характеристики кожного магазину, такі як: назва, спеціалізація, адреса, телефон, час роботи, вид власності. Є можливість утворювати нові списки магазинів, додавати, видаляти та редагувати магазини. Також, є функція формування звіту у Microsoft Excel.

**ЗМІСТ**

[ВСТУП 7](#_Toc41839878)

[1 СПЕЦИФІКАЦІЯ ПРОЕКТУ 8](#_Toc41839879)

[1.1 Користувачі програми 8](#_Toc41839880)

[1.2 Функції програми 8](#_Toc41839881)

[1.3 Функції адміністратора 9](#_Toc41839882)

[1.3.1 Функція «Вхід до програми» 9](#_Toc41839883)

[1.3.2 Функція «Додавання злочинців та злочинних угруповань до списку» 11](#_Toc41839884)

[1.3.3 Функція «Редагування списку злочинців та угруповань» 12](#_Toc41839885)

[1.3.4 Функція «Вихід з програми» 13](#_Toc41839886)

[1.4 Функції звичайного користувача 13](#_Toc41839887)

[1.4.1 Функція «Вхід до програми» 13](#_Toc41839888)

[1.4.2 Функція «Пошук злочинців» 14](#_Toc41839889)

[1.4.3 Функція «Вибірка з будь-якої підмножини ознак» 14](#_Toc41839890)

[2 ПРОЕКТНА СПЕЦИФІКАЦІЯ 16](#_Toc41839891)

[2.1 Архитектура 16](#_Toc41839892)

[2.2 Струкутра проекта 16](#_Toc41839893)

[2.3 Об’єктна модель 17](#_Toc41839894)

[2.3 Сховище і доступ до даних 20](#_Toc41839895)

[3 ІНСТРУКЦІЯ КОРИСТУВАЧА 21](#_Toc41839896)

[3.1 Призначення програми 21](#_Toc41839897)

[3.2 Робота з програмою 21](#_Toc41839898)

[3.2.1 Головне меню 21](#_Toc41839899)

[3.2.2 Авторизація 22](#_Toc41839900)

[3.2.3 Додавання нового злочинця 24](#_Toc41839901)

[3.2.4 Редагування даних злочинця 26](#_Toc41839902)

[3.2.5 Перенос злочинця до архіву 27](#_Toc41839903)

[3.2.6 Видалення злочинця із бази 28](#_Toc41839904)

[3.2.7 Додавання нового злочинного угруповання 28](#_Toc41839905)

[3.2.8 Редагування злочинного угруповання 31](#_Toc41839906)

[3.2.9 Пошук злочинців 33](#_Toc41839907)

[3.2.10 Фільтрація злочинців 33](#_Toc41839908)

[3.2.11 Перегляд інформації про злочинця 34](#_Toc41839909)

[3.2.12 Перегляд інформації про злочинне угруповання 35](#_Toc41839910)

[ВИСНОВКИ 37](#_Toc41839911)

[ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 38](#_Toc41839912)

[ДОДАТОК А 39](#_Toc41839913)

ВСТУП

Метою курсового проекту на тему «Картотека інтерполу» є вивчення основних принципів об’єктно орієнтованого програмування , оволодіння навичок розробки об’єктних програм, придбання навичок у розробці програних продуктів та розвинення навичок роботи в Windows Forms.

У практиці боротьби зі злочинністю вже давно велике значення надається збору і зберіганню кримінально-розшукової та іншої криміналістичної інформації, за допомогою якої можна провести індивідуальне ототожнення або встановити групову приналежність об'єкта, що перевіряється в процесі розкриття і розслідування злочину (опису ознак зовнішності злочинців, їх фотографії і ін.). Саме для цього створене це програмне забезпечення. Програмний продукт має забезпечувати перегляд та пошук інформації про злочинців, які знаходяться в розшуку, та про діючі злочинні угруповання. Програма буде корисною при виконанні службових обов’язків працівниками правоохоронних органів.

Розроблене програмне забезпечення має відповідати таким загальним вимогам:

* стійкість програми;
* забезпечення цілісності даних;
* функціональна повнота;
* наявність роботи з користувачем;
* наявність тестування.

1 СПЕЦИФІКАЦІЯ ПРОЕКТУ

## Користувачі програми

У проекті є дві категорії користувачів, яким необхідні різноманітні функції програми – це адміністратор та звичайний користувач.

Звичайний користувач може шукати злочинців та переглядати інформацію про них та про угруповання, яким вони належать.

Адміністратор підтримує список злочинців та угруповань, редагує інформацію, переносе до архіву, видаляє.

В момент входу в програму користувач програми автоматично стає звичайним користувачем. Здобути права адміністратора він може тільки при аутентифікації даних.

## 1.2 Функції програми

Програма надає звичайному користувачу такі можливості:

1. Вхід до програми
2. Пошук злочинців та злочинних угруповань
3. Вибірка із будь-якої підмножини ознак
4. Вихід з програми

Програма надає адміністратору наступні можливості

1. Вхід до програми
2. Додавання злочинців
3. Додавання злочинних угруповань
4. Редагування списку злочинців
5. Редагування злочинних угруповань
6. Завантаження даних із постійної пам’яті в оперативну
7. Збереження поточного стану даних в постійній пам’яті
8. Вихід з програми

Крім того адміністратору надаються усі функції звичайного користувача, для контролю цих функцій

## 1.3 Функції адміністратора

### 1.3.1 Функція «Вхід до програми»

Адміністратор заходить до програми за допомогою імені та пароля. Імена та паролі адміністраторів зберігаються в даних програми. Функції адміністратора сховані від звичайного користувача. Тому вхід адміністратора починається с натискання певної комбінації клавіш.

Основний сценарій:

1. Адміністратор натискає комбінацію клавіш Ctrl + F12 і відкривається вікно входу адміністратора.(див. рис.1.1)
2. Адміністратор вводить пароль та логін і нажимає кнопку «ОК».
3. Введенні данні перевіряються, і ,якщо данні вірні , на головній формі з’являється панель адміністратора у вигляді випадаючого меню(див. рис. 1.2)
4. Якщо данні не пройшли перевірку, з’являється повідомлення про це і надається спроба увійти ще раз. Кількість спроб не обмежена.

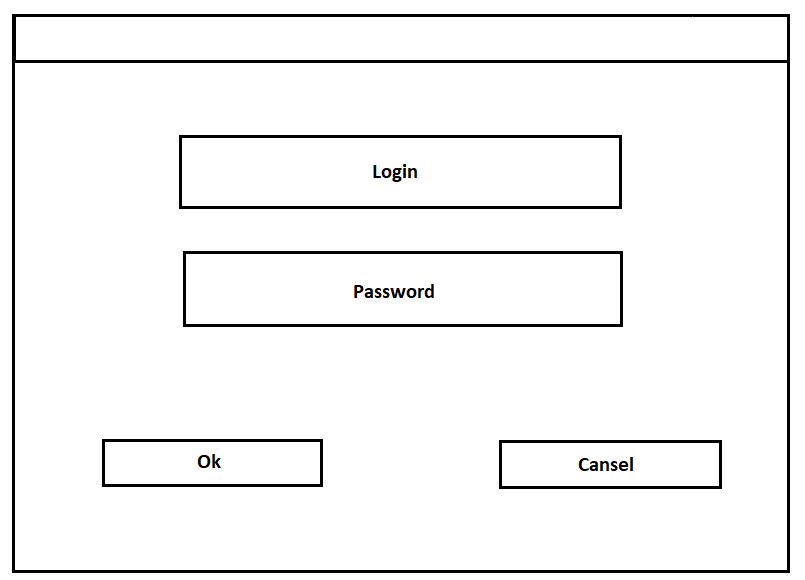


Рисунок 1.1 – Панель входу адміністратора

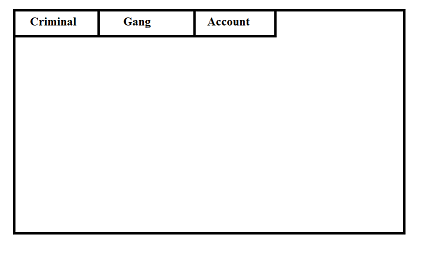


Рисунок 1.2 – Панель адміністратора

### 1.3.2 Функція «Додавання злочинців та злочинних угруповань до списку»

У програмі буде декілька списків злочинців. В одному будуть усі зареєстровані в системі злочинці, у другому – архівні злочинці, тобто ті , які уже виправилися. Ці списки використовуються при пошуку. У кожного угруповання є список членів угруповання.

Сценарій «Додавання злочинця»:

1. На панелі адміністратора адмін «Criminals» -> «Add».
2. Відкривається форма додавання злочинця. Усі поля порожні.(див. рис. 1.3)
3. Заповнює поля форми, обирає належність до угруповання, обирає фото.
4. Натискає кнопку «Зберегти»

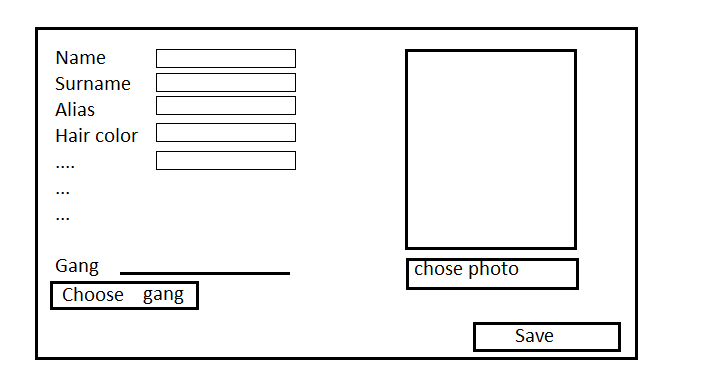


Рисунок 1.3 – Форма додавання злочинця

Сценарій «Додавання угруповання»:

1. На панелі адміністратора адмін обирає «Gang» -> «Add».
2. Відкривається форма додавання угруповання(рис 1.4). Усі поля порожні
3. Адмін заповнює поля форми, обирає злочинців, які належать до цього угруповання, обирає емблему.
4. Натискає кнопку «Зберегти»

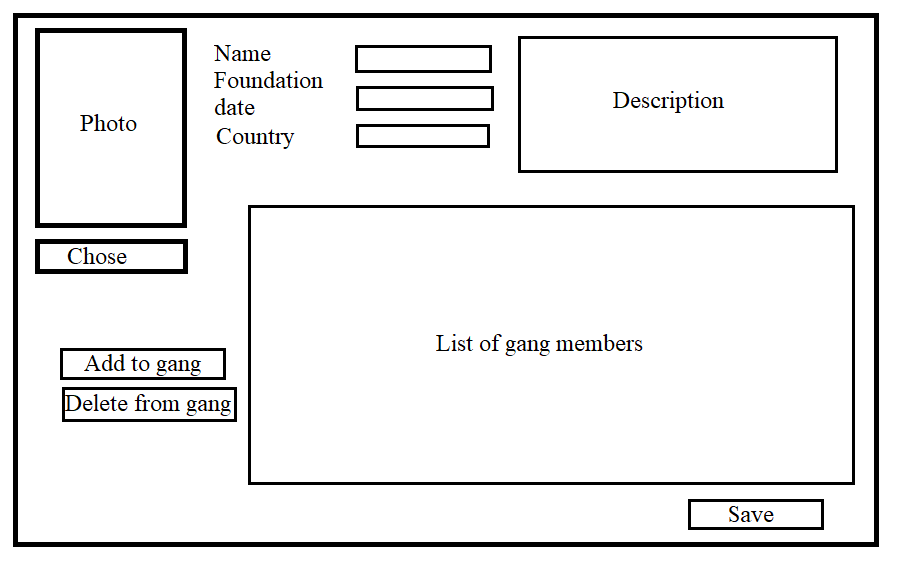


Рисунок 1.4 – Форма додавання злочинців

### 1.3.3 Функція «Редагування списку злочинців та угруповань»

Неможливо завжди діяти без помилок, тому у програмі повинна бути можливість не тільки додавати до списку, а й оновлювати інформації, виправляти помилки, заносити до архіву .

Сценарій «Редагування анкет злочинців»:

1. Адміністратор стає на запис потрібного злочинця та на панелі адміністратора обирає «Criminals» -> «Edit».
2. Адміну надається його анкета с правом на редагування .
3. Там він може змінити усю інформацію про злочинця, додати його до якоїсь злочинної організації, зареєстрованої в програмі.
4. Натискає кнопку «Зберегти» .

Сценарій «Редагування угруповань» нічим особливим не відрізняється від «Редагування анкет злочинців», окрім того , що до угруповання можна додавати злочинців .

Сценарій «Видалення анкет злочинців» :

1. Адмін стає на запис потрібного злочинця та на панелі адміністратора адмін обирає «Criminals» -> «Delete».
2. Адміністратору потрібно підтвердити свою дію.
3. Після підтвердження злочинець видаляється без змоги відновлення.

1.3.4 Функція «Вихід з програми»

У програмі зберігання відбувається автоматично. Усі дії, які ведуть до втрати інформації потребують підтвердження.

## 1.4 Функції звичайного користувача

1.4.1 Функція «Вхід до програми»

Оскільки звичайний користувач не має функції реєстрації, то вхід до програми здійснюється автоматично з відкриттям програми.

1.4.2 Функція «Пошук злочинців»

Звичайний користувач може користуватися пошуком по прізвищу, імені, прізьвиську.

Сценарій :

1. У головному меню користувач обирає критерії пошуку (прізвищу, імені, прізьвиську) та вводить дані у поле.
2. Автоматично змінюється список злочинців. Показуються лише ті, які підпадають під критерії пошуку.

1.4.3 Функція «Вибірка з будь-якої підмножини ознак»

У вікні списку злочинців/угруповань(рис. 1.5) є можливість вибірка з будь-якої множини ознак. Це зроблено для покращення зручності пошуку.

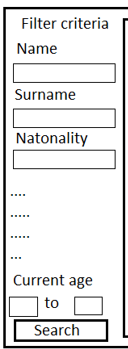


Рисунок 1.5 – Приклад фільтру за будь-якою множиною ознак

2 ПРОЕКТНА СПЕЦИФІКАЦІЯ

## 2.1 Архитектура

Програма буде розроблятися на мові С# с використаням вереди розробки Microsoft Visual Studio 2019 на платформі .Net Framework 4.7.2 з використанням інтерфейсу клієнтських програм Windows Forms.

Вибір мови обумовлений обьєкно-орієнтованою орієнтацією. На данний момент мова С# є однією з найпопулярніших, зручних і повноцінних серед мов програмування.

У процесі розробки розглядалися два фреймворки юзер інтерфейсу: Windows Forms та WPF(Windows Presentation Foundation). Вибір пав на перший, бо WPF значно складніша в освоєнні, а Windows Forms задовільняв усім задачам , поставленими перед проектом. Також була обрана платформа .Net Framework, бо перед проектом не стояла задача кроссплатформеності.

## 2.2 Струкутра проекта

У программі присутні два типа кроистувачів: адміністратор та звичайний користувач. З початку роботи програми кожний користувач це звичайний користувач. У процессі роботи користувач за допомогою авторизації може здобути права адміністратора. Крім того для тестування програми на етапі розробки слід зробити кілька тестових данних для перевірки правильності роботи програми.

Слідуючи з цього можна скласти наступну структуру проекту (див. рис. 2.1)

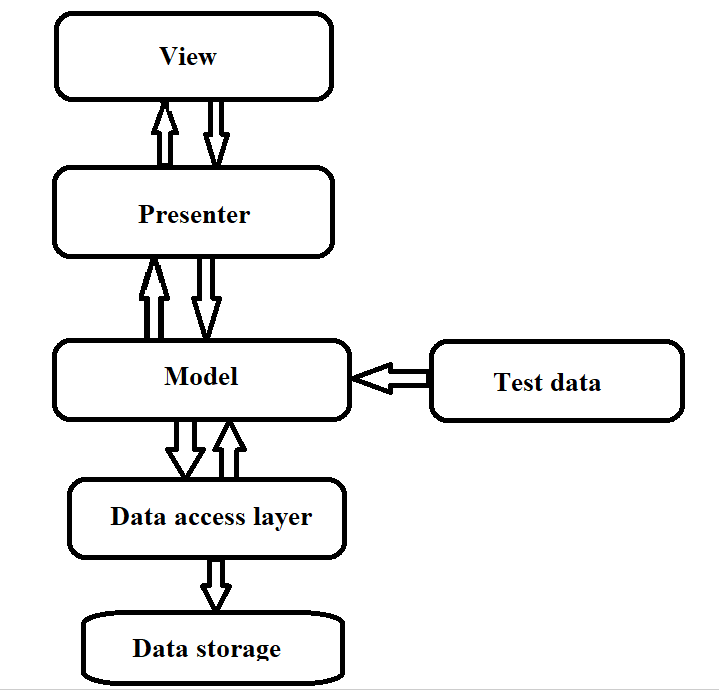


Рисунок 2.1 – Структура проекту

## 2.3 Об’єктна модель

Програма складається з 6 класів, 1 статичного класу, 1 перечислення та 6 форм (див. рис . 2.2)

У класі FileCabinet (див. додаток А) зберігаються три колекції: колекція діючих злочинців, колекція архівних злочинців, колекція злочинних угруповань. Є такі методи:

* GetInstance() – повертає екземпляр даного класу.
* FileCabinet() – створює екземпляр класу FileCabinet
* GenerateMembers(int n) – заповнює колекції тестовими даними
* MoveToArchive(Criminal criminal) – переносить даного злочинця до архіву
* GetGanglessCriminals() – повертає усіх злочинців, які не состоять у злочинних угрупованнях
* Search(string value, string criteria, List<Criminal> list) – повертає колекцію, яка відповідає певним критеріям
* Save() – зберігає дані
* Load() – загружає дані

У класі Criminal зберігаються усі дані про злочинця. Про злочинця відомо: ім’я, прізвище, прозьвисько, зріст, колір волосся та очей, вік, стать, національність, дата народження, адресса, прописка, короткий опис, країна де розшукується, злочинне угруповання до якого належить та унікальний ідентифікатор.

У класі CriminalGang зберігаються усі дані про злочинне угруповання. Про угруповання відомо: назва, члени угруповання, ворогуючі угруповання, союзні угруповання, особливості, дата заснування, лідер, країна.

Класс Adress призначений для зручного зберігання адреси. Про адресу відомо: місто, вулиця, номер вулиці.

У класі Admin зберігається логін та пароль адміністратора.

Клас User показує, яка роль у користувача програми на даний момент.

Перечислення UserRole зберігає логічні константи які відповідають за роль користувача у програмі наданий момент.

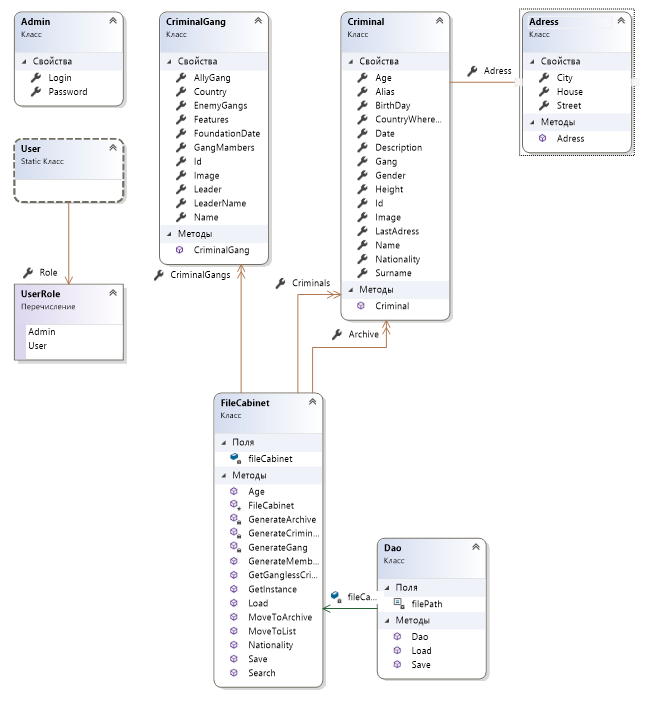


Рисунок 2.2 –Діаграма класів

## 2.3 Сховище і доступ до даних

Зберігатися усі дані будуть у бінарному фалі «fileCabinet.bin». Зберігання буде відбуватися за допомогою класса серіалізації та десеріалізації вбудованного класу BinaryFormatter. Захист данних у рамках данного проекту не розглядається.

3 ІНСТРУКЦІЯ КОРИСТУВАЧА

## 3.1 Призначення програми

«Картотека інтерполу» призначена для користування та управління базою даних злочинців.

Основні функції програми:

* додавання нового злочинця у базу;
* додавання нового злочинного угруповання;
* редагування злочинця;
* редагування злочинного угруповання;
* перенос злочинців до архіву;
* видалення злочинців;
* фільтрування злочинців;
* пошук злочинців;
* сортування злочинців;
* додавання злочинців до злочинного угруповання.

## 3.2 Робота з програмою

### 3.2.1 Головне меню

При вході в програму користувача очкує головне меню програми (див. рис.3.1)

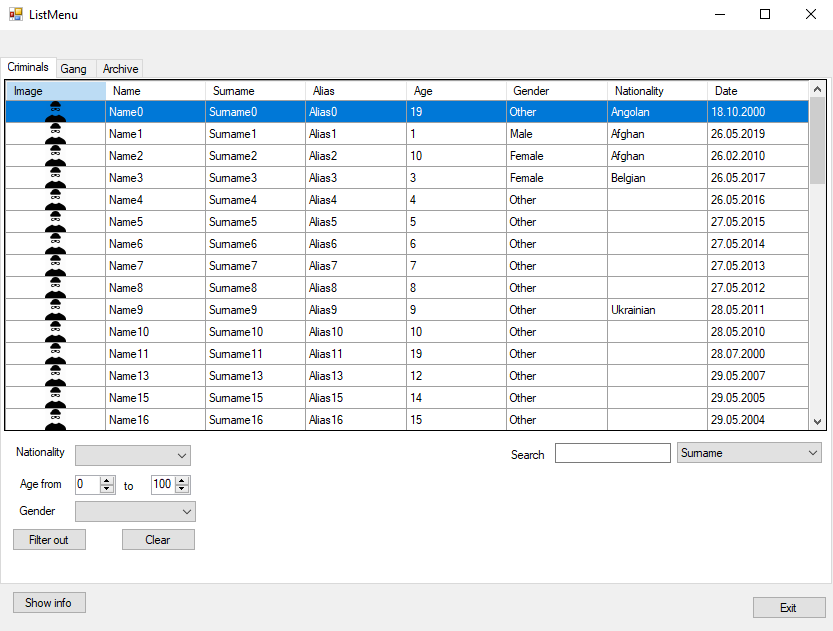


Рисунок 3.1 – Головне меню

З початку роботи програми користувач працює у режимі звичайного користувача.

### 3.2.2 Авторизація

Для того, щоб користуватися правами адміністратора користувач має натиснути комбінацію клавіш «Ctrl» + «Insert». Відкриється вікно авторизації:

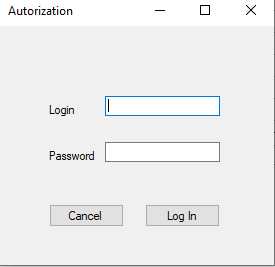


Рисунок 3.2 – Форма авторизації

Після успішної валідації даних форма на головній формі з’явиться панель адміністратора (див. рис. 3.3).



Рисунок 3.3 – Панель адміністратора

При роботі в режимі адміністратора користувач має можливість виконувати усі функції, описані вище.

Вибір дії виконується за допомогою меню, зображеному на рисунку 3.4, або кнопок, зображених на рисунку 3.5.

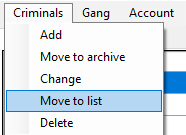


Рисунок 3.4 – Меню



Рисунок 3.5

### 3.2.3 Додавання нового злочинця

Додавання нового злочинця адміністратор може здійснити вибравши певні пункти з меню (рис 3.2) «Criminals» -> «Add».

Екран введення даних має вигляд(див. рис. 3.6)

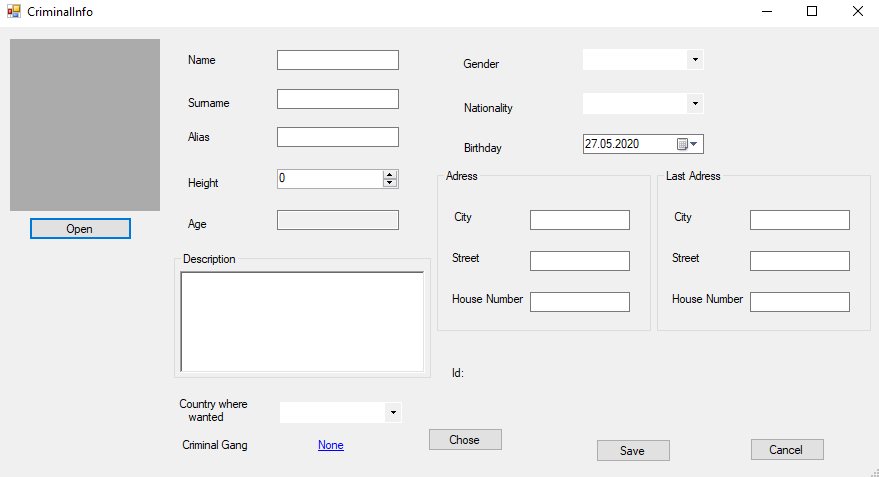


Рисунок 3.6 – Екран введення даних злочинця

Адміністратор має заповнити усі обов’язкові поля. Якщо цього не відбудеться, то обов’язкові поля стануть червоними і спеціальне вікно проінформує адміністратора, що не всі поля заповнені.

Вибір фотографії злочинця виконується за допомогою натискання клавіші «Open». Відкриється діалогове вікно вибору фотографії.

Натиснувши на кнопку «Chose» у адміністратора з’являється можливість додати даного злочинця до списку злочинців банди, яка уже зареєстрована у системі. Відкривається вікно вибору злочинного угруповання(див. рис. 3.7)

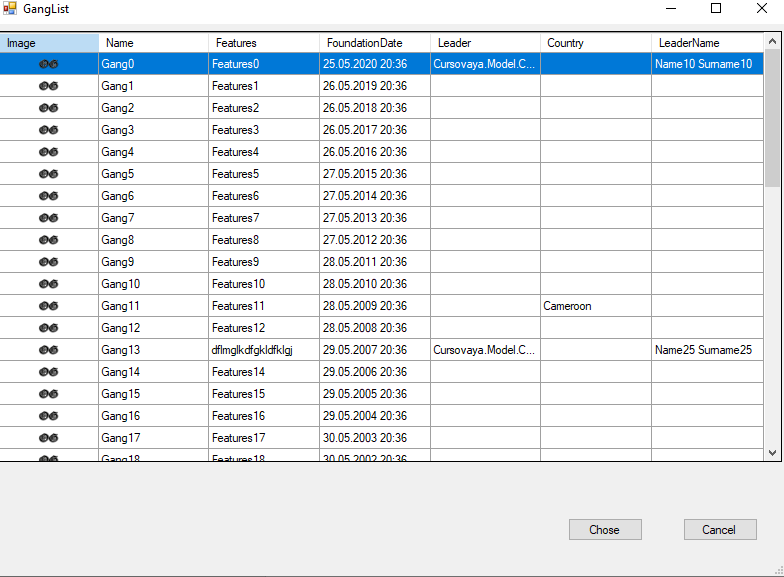


Рисунок 3.7 – Вікно вибору злочинного угруповання

Вибравши потрібне угруповання, адміністратор натискає кнопку «Chose».

Натиснувши кнопку «Save» адміністратор підтверджує своії дії. Вікно закриється, а злочинець додається до бази.

Натиснувши кнопку «Cancel» адміністратор відхиляє свої дії. Вікно закриється без додавання.

### 3.2.4 Редагування даних злочинця

Для редагування інформації адміністратору треба за допомогою «миші» або клавіатури стати на запис с даними необхідного злочинця та визвати форму редагування, вибравши з меню (див. рис 3.4) «Criminals» -> «Edit», натиснувши кнопку , або виконати подвійний натиск на ліву кнопку миші.

Форма для редагування має вигляд :

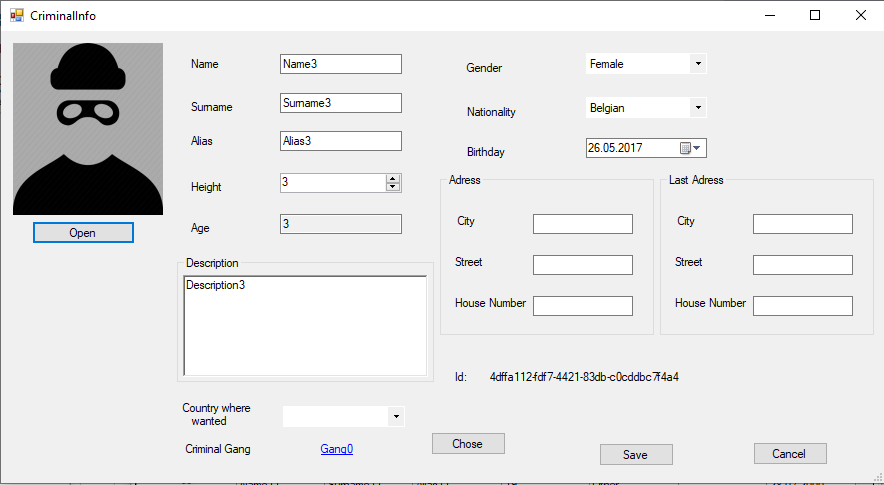


Рисунок 3.8 – Форма редагування злочинців

Адміністратор може змінити усі дані злочинця окрім ідентифікаційного номеру.

Для підтвердження змін даних злочинця треба натиснути на кнопку «Save».

Для відмови змін даних злочинця треба натиснути кнопку «Cancel».

### 3.2.5 Перенос злочинця до архіву

Для переносу злочинця до архіву адміністратору необхідно за допомогою «миші» або клавіатури стати на запис з даними злочинця та вибрати з меню(рис 3.4) «Criminals» -> «Move to archive» . У адміністратора попросить підтвердження його дії (див. рис. 3.9)

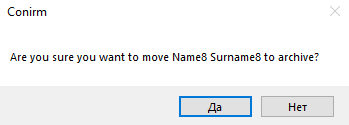


Рисунок 3.9 – Підтвердження переносу до архіву

Натиснувши кнопку «Yes», злочинець заноситься до архіву та видаляється із основного списку.

Натиснувши кнопку «No», адміністратор відхиляє свої дії.

### 3.2.6 Видалення злочинця із бази

Для видалення злочинця адміністратору необхідно за допомогою «миші» або клавіатури стати на запис с даними злочинця та вибрати з меню (рис 3.4) «Criminals» -> «Delete». У адміністратора попросить підтвердження його дії (див. рис. 3.10)

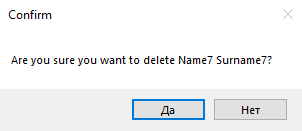


Рисунок 3.10 – Підтвердження видалення злочинця

Натиснувши кнопку «Yes», злочинець заноситься до архіву та видаляється із основного списку.

Натиснувши кнопку «No», адміністратор відхиляє свої дії.

### 3.2.7 Додавання нового злочинного угруповання

Додавання нового злочинного угруповання адміністратор може здійснити вибравши в меню «Gang» -> «Add»

Екран введення даних має вигляд(див. рис. 3.11)

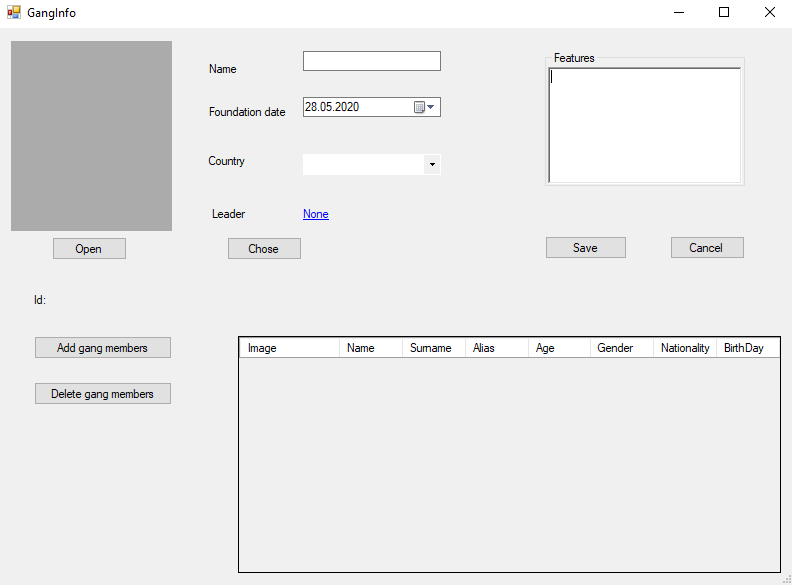


Рисунок 3.11 – Екран введення даних злочинного угруповання

Вибір фотографії злочинного угруповання виконується за допомогою натискання клавіші «Open». Відкриється діалогове вікно вибору фотографії.

Натиснувши кнопку «Add gang members», відкривається діалогове вікно вибору членів угруповання (див. рис 3.12).

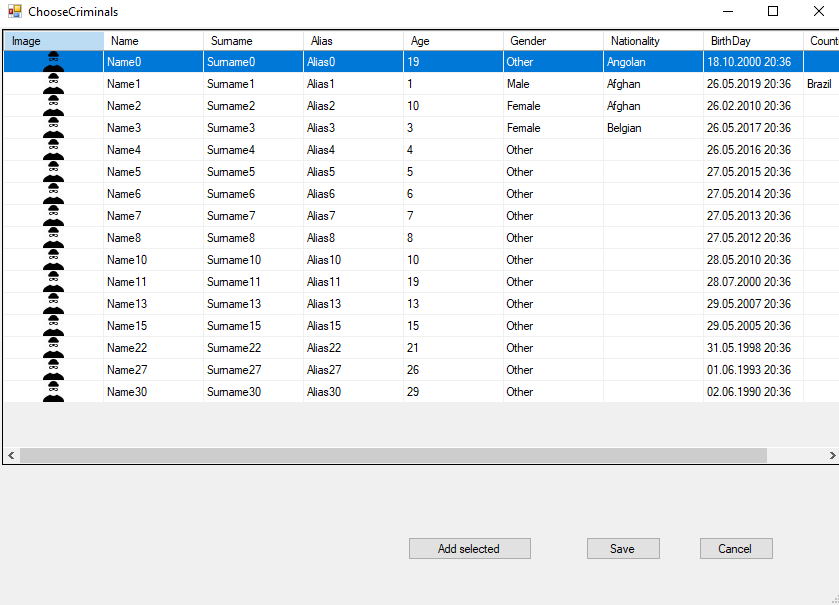


Рисунок 3.12 – Вікно вибору членів угруповання

Адміністратор може виділити усіх необхідних злочинців і натиснути кнопку «Add selected». Для підтвердження дії необхідно натиснути на кнопку «Save». Для відхилення – кнопку «Cancel».

Для того щоб видалити злочинця, адміністратору необхідно натиснути стати на записі злочинця, якого потрібно видалити, і натиснути на кнопку «Delete gang member». Адміністратор має підтвердити свою дію (див рис. 3.13)

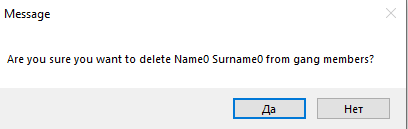


Рисунок 3.13 – Підтвердження видалення члена угруповання

Для того, щоб зробити члена банди лідером, необхідно встати на записі цього злочинця та натиснути кнопку «Chose». Адміністратор має підтвердити свою дію (див. рис. 3.14)

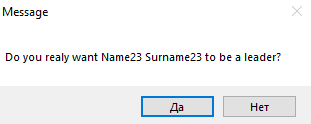


Рисунок 3.14 – Підтвердження вибору лідера угруповання

Натиснувши на кнопку «Save» адміністратор підтверджує свої дії. Злочинне угруповання додається до бази.

Натиснувши на кнопку «Cancel» адміністратор відхиляє свої дії.

### 3.2.8 Редагування злочинного угруповання

Для редагування інформації адміністратору треба за допомогою «миші» або клавіатури стати на запис с даними необхідного злочинного угруповання та визвати форму редагування, вибравши з меню (рис 3.4) «Gang» -> «Edit», натиснувши кнопку «Show info», або виконати подвійний натиск на ліву кнопку миші.

Відкриється вікно редагування (див. рис. 3.15).

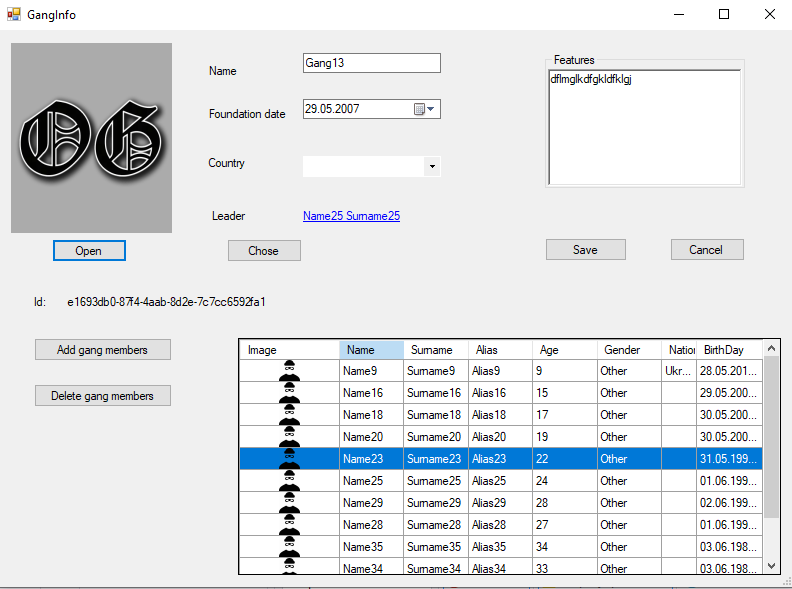


Рисунок 3.15 – Вікно редагування злочинного угруповання

Адміністратор може змінити усі доступну інформацію, окрім ідентифікаційного номеру.

Для підтвердження своїх дій він має натиснути на кнопку «Save».

Для відхилення своїх дій адміністратор має натиснути на кнопку «Cancel».

### 3.2.9 Пошук злочинців

Для пошуку необхідного злочинця, користувач на головній формі(рис 3.1) може вибрати критерій пошуку(за замовченням пошук проводиться за прізвищем) та почати пошук за цим критерієм. Інтерфейс пошуку зображений на малюнку 3.16:



Рисунок 3.16 – Інтерфейс пошуку злочинців

### 3.2.10 Фільтрація злочинців

Для фільтрації злочинців, користувач має обрати на панелі фільтрування (див. рис. 3.17) необхідні йому критерії та натиснути кнопку «Filter out». Для сбросу критеріїв треба натиснути кнопку «Clear».

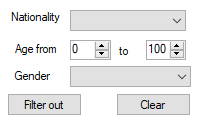


Рисунок 3.17 – Панель фільтрування

### 3.2.11 Перегляд інформації про злочинця

Переглянути інформації про злочинця звичайний користувач може ставши на запис відповідного злочинця та натиснувши клавішу «Show info» або виконавши подвійний натиск на ліву кнопку миші.

Відкриється вікно інформації про злочинця (див. рис. 3.18)

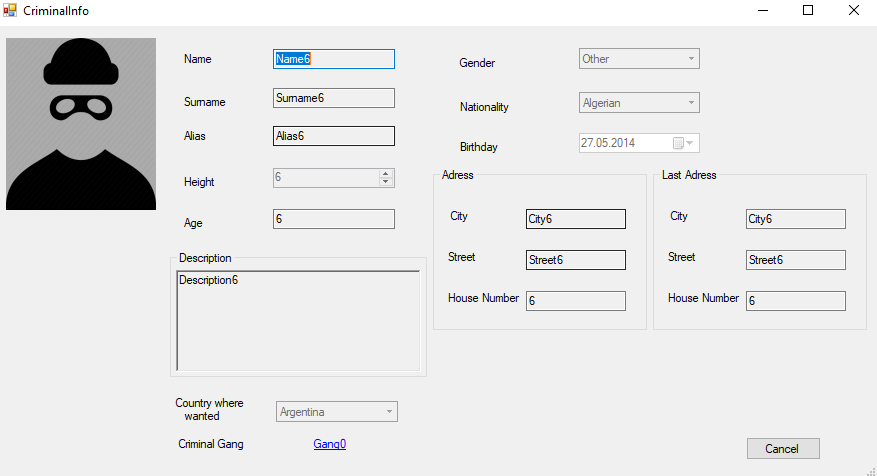


Рисунок 3.18 – Вікно інформації про злочинця

Звичайний користувач не має права змінювати інформацію, лише переглядати.

Закрити це вікно користувач може за допомогою клавіши «Cancel» або «хрестика».

### 3.2.12 Перегляд інформації про злочинне угруповання

Переглянути інформації про злочинне угруповання звичайний користувач може ставши на запис відповідного злочинного угруповання та натиснувши клавішу «Show info» або виконавши подвійний натиск на ліву кнопку миші.

Відкриється вікно інформації про злочинне угруповання (див. рис. 3.19)

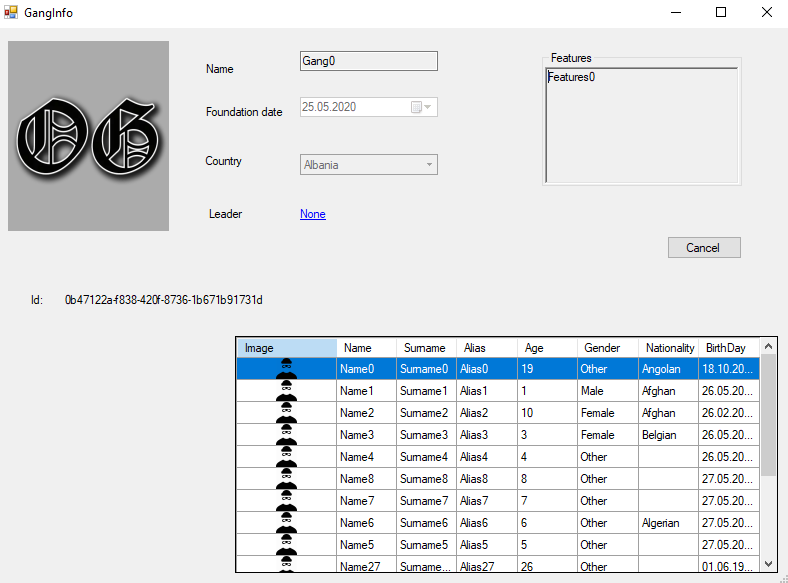


Рис 3.19 – Вікно інформації про угруповання

3.2.13 Вихід з програми та вихід з акаунту адміністратора

Для того щоб вийти з акаунту, адміністратор повинен вибрати з меню (рис 3.4) «Account» -> «Exit».

Для того щоб вийти з програми, користувач повинен натиснути на кнопку «Exit» на головній формі.

ВИСНОВКИ

Працюючи над курсовою роботою, були вивчені окремі фази розробки програмного забезпечення та поєднані в одне ціле – в проект. У ході виконання проекту було використано принципи об’єктно – орієнтованого програмування та реалізовано об’єктні моделі мови програмування.

У курсовому проекті була розроблена програма під назвою «Картотека інтерполу». Програмне забезпечення буде корисним при виконанні службових обов’язків працівниками правоохоронних органів. За допомогою програми можна переглядати інформацію про злочинців, які розшукуються, та про діючі злочинні угруповання. Для зручності використання є функція пошуку та фільтрації злочинців. Адміністратор програми має доступ до всіх функцій звичайного користувача, а також може редагувати та додавати нову інформацію про злочинців та угруповання.

Програмне забезпечення реалізовано за допомогою мови програмування C# на платформі .NET Framework 4.7.2 з використанням технологій Windows Forms у середі розробки Microsoft Visual Studio 2019.

Усі дані програми зберігаються локально за допомогою серіалізації даних у файлі формату «BIN». Програму можна покращити, додавши до неї базу даних, та удосконалити пошук злочинців та злочинних угруповань.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Бондарев В. М. Объектно-ориентированное программирование на С# [Текст] : учеб. Пособие / В. М. Бондарев. – Харьков : СМИТ, 2009. –224 с. – Библиогр. : ISBN 978-966-2028-34-8.
2. Microsoft Developer Network. Библиотека MSDN / MSDN – сеть разработчиков Microsoft [Електронний ресурс] – Режим доступу до ресурсу: <https://msdn.microsoft.com/ru-ru/>.
3. Руководство по программированию в Windows Forms [Електронний ресурс] // METANIT.COM. – 2016. – Режим доступу до ресурсу: <https://metanit.com/sharp/windowsforms/>.
4. Полное руководство по языку программирования С# 8.0 и платформе .NET Core [Електронний ресурс] // METANIT.COM. – 2019. – Режим доступу до ресурсу: <https://metanit.com/sharp/tutorial/>.
5. Методичні вказівки до виконання курсової роботи з дисципліни «Об’єктно-орієнтоване програмування» для студентів першого (бакалаврського) рівня вищої освіти усіх форм навчання спеціальності 121 «Інженерія програмного забезпечення», освітньо-професійна програма «Програмна інженерія» / Упоряд.: В.М. Бондарєв, Ю.Ю. Черепанова – Харків: ХНУРЕ, 2020. – 45 с.

ДОДАТОК А

Клас Dao

//Клас, який відповідає за сериалізацію даных

public class Dao

{

private FileCabinet fileCabinet;

const string filePath = "fileCabinet.bin";

//Конструктор, який приймає на вхід екззкмпляр класу FileCabinet

public Dao(FileCabinet store)

{

fileCabinet = store;

}

//Метод сериалізації

public void Save()

{

using (Stream stream = File.Create(filePath))

{

var serializer = new BinaryFormatter();

serializer.Serialize(stream, fileCabinet);

}

}

//Метод десериалізації

public void Load()

{

using (Stream stream = File.OpenRead(filePath))

{

var serializer = new BinaryFormatter();

FileCabinet st = (FileCabinet)serializer.Deserialize(stream);

Copy(st.Criminals, fileCabinet.Criminals);

Copy(st.CriminalGangs, fileCabinet.CriminalGangs);

Copy(st.Archive, fileCabinet.Archive);

}

void Copy<T>(List<T> from, List<T> to)

{

to.Clear();

to.AddRange(from);

}

}

}

Клас Admin

//Класс Admin в якому зберігається логін та пароль адміна

public class Admin

{

public string Login { get; } = "admin";

public string Password { get; } = "123456";

}

Клас Adress

//Класс адресових даних

public class Adress

{

public string City{ get; set; }

public string Street { get; set; }

public string House { get; set; }

public Adress(string city, string street, string house)

{

City = city;

Street = street;

House = house;

}

}

Клас Criminal

//Клас злочинця в якому зберігається вся відома інформація про злочинця

public class Criminal

{

public Criminal()

{

Id = Guid.NewGuid();

}

public Bitmap Image { get; set; }

public string Name { get; set; }

public string Surname { get; set; }

public string Alias { get; set; }

public int Height { get; set; }

public int Age

{

get

{

var i = DateTime.Now.Year - BirthDay.Year;

if (DateTime.Now.DayOfYear - BirthDay.DayOfYear < 0)

{

i--;

}

return i;

}

}

public string Gender { get; set; }

public string Nationality { get; set; }

public DateTime BirthDay { get ; set ; }

public string Date { get => BirthDay.ToShortDateString(); }

public Adress Adress { get; set; }

public Adress LastAdress { get; set; }

public string Description { get; set; }

public string CountryWhereWanted { get; set; }

public CriminalGang Gang{ get; set; }

public Guid Id { get;private set; }

}

Клас CriminalGang

//Клас у якому зберігається уся інформація про злочинне угруповання

public class CriminalGang

{

public CriminalGang()

{

GangMambers = new List<Criminal>();

EnemyGangs = new List<CriminalGang>();

AllyGang = new List<CriminalGang>();

Id = Guid.NewGuid();

}

public Bitmap Image { get; set; }

public string Name { get; set; }

public List<Criminal> GangMambers { get; set; }

public List<CriminalGang> EnemyGangs { get; set; }

public List<CriminalGang> AllyGang { get; set; }

public string Features { get; set; }

public DateTime FoundationDate { get; set; }

public Criminal Leader { get; set; }

public string Country { get; set; }

public Guid Id { get; private set; }

public string LeaderName

{

get

{

return Leader == null ? " " : Leader.Name + " " + Leader.Surname;

}

}

}

Клас FileCabinet

//Клас у якому зберігаються колекції злочинців, архівних злочинців та злочинних угруповань

public class FileCabinet

{

static private FileCabinet fileCabinet;

//Возвращает текущий екземпляр класса

static public FileCabinet GetInstance()

{

if (fileCabinet == null) // если текущий екзепляр еще не создан, то создаем его

{

fileCabinet = new FileCabinet();

}

return fileCabinet;

}

//Конструктор

protected FileCabinet()

{

Criminals = new List<Criminal>();

CriminalGangs = new List<CriminalGang>();

Archive = new List<Criminal>();

}

//Картотека = коллекция преступников + колекция банд

public List<Criminal> Archive { get; set; }

public List<Criminal> Criminals { get; set; }

public List<CriminalGang> CriminalGangs { get; set; }

//Метод для генерации тестовых данных

public void GenerateMembers(int n)

{

GenerateCriminals(n);

GenerateArchive(n);

GenerateGang(n);

}

private void GenerateGang(int n)

{

CriminalGangs = new List<CriminalGang>();

for(int i = 0; i < n;i++)

{

var c = new CriminalGang()

{

Image = new Bitmap("D:/Project/C#/OOP/Cursova/Cursovaya/Images/1.png"),

Name = $"Gang{i}",

Features = $"Features{i}",

FoundationDate = DateTime.Now - TimeSpan.FromDays(i \* 365),

Country = ""

};

CriminalGangs.Add(c);

}

}

private void GenerateCriminals(int n)

{

Criminals = new List<Criminal>();

for (int i = 0; i < n;i++)

{

var c = new Criminal()

{

Image = new Bitmap("D:/Project/C#/OOP/Cursova/Cursovaya/Images/criminal.png"),

Name = $"Name{i}",

Surname = $"Surname{i}",

Alias = $"Alias{i}",

Height = i,

Gender = "Other",

Nationality = null,

BirthDay = DateTime.Now - TimeSpan.FromDays(i \* 365),

Description = $"Description{i}"

};

Criminals.Add(c);

}

}

private void GenerateArchive(int n)

{

Archive = new List<Criminal>();

for (int i = 50; i < n + 50; i++)

{

var c = new Criminal()

{

Image = new Bitmap("D:/Project/C#/OOP/Cursova/Cursovaya/Images/criminal.png"),

Name = $"Name{i}",

Surname = $"Surname{i}",

Alias = $"Alias{i}",

Height = i,

Gender = "Other",

Nationality = null,

BirthDay = DateTime.Now - TimeSpan.FromDays(i \* 365),

Description = $"Description{i}"

};

Archive.Add(c);

}

}

//Метод перемещания преступника в архив

public void MoveToArchive(Criminal criminal)

{

Criminals.Remove(criminal);

Archive.Add(criminal);

}

//Метод перемещения преступника в коллекцию текущих из архива

public void MoveToList(Criminal criminal)

{

Criminals.Add(criminal);

Archive.Remove(criminal);

}

//Метод возвращает всех преступников у которые не состоят в преступных групировках

public List<Criminal> GetGanglessCriminals()

{

List<Criminal> res = new List<Criminal>();

foreach(Criminal c in Criminals)

{

if (c.Gang == null)

{

res.Add(c);

}

}

return res;

}

//Производит поиск преступников из данного списка по определенным критериям(фамилии, имени, кличке)

public List<Criminal> Search(string value, string criteria, List<Criminal> list)

{

value = value.ToLower();

List<Criminal> res = new List<Criminal>();

switch (criteria)

{

case "Surname":

foreach(Criminal i in list)

{

if(i.Surname.ToLower().StartsWith(value))

{

res.Add(i);

}

}

break;

case "Name":

foreach (Criminal i in list)

{

if (i.Name.ToLower().StartsWith(value))

{

res.Add(i);

}

}

break;

case "Alias":

foreach (Criminal i in list)

{

if (i.Alias.ToLower().StartsWith(value))

{

res.Add(i);

}

}

break;

}

return res;

}

//Сохраняет данные

public void Save()

{

new Dao(fileCabinet).Save();

}

//Загружает данные

public void Load()

{

new Dao(fileCabinet).Load();

}

}

Клас User

//Класс отвечает за идентификацию пользователя

static public class User

{

static public UserRole Role { get; set; } = UserRole.Admin;

}

Перечислення UserRole

//Перечесление которое кранит два состояния пользователя

public enum UserRole

{

Admin,

User

}

Посилання на GitHub: <https://github.com/hlv-z/Cursovaya>