# Міністерство освіти і науки України Харківський національний університет радіоелектроніки

Кафедра Програмної інженерії

КУРСОВА РОБОТА ПОЯСНЮВАЛЬНА ЗАПИСКА

з дисципліни “Об’єктно - орієнтоване програмування” “Картотека Интерпола”

Керівник , проф. Бондарєв В.М.

Студент гр. ПІ - 15 – 1 Слупський Д.О.

Комісія:

проф. Дудар З.В.

<посада> <прізвище, ініціали >

<посада> <прізвище, ініціали >

Харків 2016

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ**

Кафедра: ***Програмної інженерії***

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

Спеціальність: ***Програмна інженерія***

Курс 1 . Група *ПІ-15-1* . Семестр *2* .

***ЗАВДАННЯ***

***на курсовий проект студента Слупського Данили Олеговича***

# 1. Тема проекту: «Картотека Інтерполу»

2. Термін здачі студентом закінченого проекту: ***“01” - червня - 2016 р***.

# 3. Вихідні дані до проекту: база злочинців, які розшукуються Інтерполом: прізвище, ім’я, кодове ім’я, зріст, колір волосся, колір очей, громадянство, країна, місце і дата народження, останнє місце проживання, знання мов, злочини; кримінальні угруповання та інформація про їх членів. Вибірка за будь-якою множиною параметрів. Розподіл злочинців на три категорії: “розшукуються”, “впіймані”, “вбиті”. Видалення інформації за спеціальним дозволом.

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |
| --- | --- | --- |
| *№* | *Назва етапу* | *Термін виконання* |
| 1 | Видача теми, узгодження і затвердження теми | 1-03-2016 р. |
| 2 | Формулювання вимог до програми | 1-03-2016 – 4-03-2016 р. |
| 3 | Розробка моделі програмного продукту | 9-03-2016 – 25-03-2016 р. |
| 4 | Розробка підсистеми відображення, | 26-03-2016 – 5-04-2016 р. |
|  | редагування та фільтрації списку |  |
|  | злочинців |  |
| 5 | Розробка підсистеми відображення та | 6-04-2016 – 15-04-2016 р. |
|  | редагування списку кримінальних |  |
|  | угруповань |  |
| 6 | Розробка підсистеми зберігання /  зчитування інформації. | 15-04-2016 – 17-04-2016 р. |
| 7 | Оформлення пояснювальної записки, додатків, графічного матеріалу | 10-05-2016 – 24-05-2016 р. |
| 8 | Захист | 01-05-2016 р. |

# Студент Слупський Д.О.

Керівник Бондарєв В. М.

« 1 »\_березня 2016 р.

РЕФЕРАТ

Пояснювальна записка до курсової роботи: 91 с., 26 рис., 1 додаток, 6 джерел.

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

Методи розробки базуються на використанні середи розробки Microsoft Visual Studio 2015 Community, графічної підсистеми Windows Presentation Foundation, платформи .NET Framework 4.5, мови програмування C#.

Профіль на github: https://github.com/notdetninja47/InterpolDatabase

В результаті отримана програма під назвою “Картотека Інтерполу”, яка дозволяє зберігати інформацію про злочинців (ім’я, прізвище, кодове ім’я, зріст, вік, додаткова інформація, дата, країна та місце народження, останні країна та місце проживання, фото, причетність до кримінального угруповання, злочини, чинний статус), кримінальні угруповання (назва, члени, додаткова інформація). Є можливість робити вибірки за всіма зазначенними вище параметрами, редагувати інформацію. Видалення інформації можливе, але проводиться за попередньо отриманим дозволом.

ОБ’ЕКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ, ПРОГРАМА, КЛАС, МОВА ПРОГРАМУВАННЯ C#, ПЛАТФОРМА .NET, ЗЛОЧИНЕЦЬ, КРИМІНАЛЬНЕ УГРУПОВАННЯ, WPF.

ЗМІСТ

Вступ ..................................................................................................... 6

1 Специфікація програми..................................................................... 7

1.1 Функції програми............................................................................ 8

1.2 Інтерфейс користувача ................................................................... 9

2 Проектна специфікація. .................................................................... 24

2.1 Архітектурний стиль ...................................................................... 24

2.2. Об’єктна структура програми....................................................... 24

2.3 Формат даних .................................................................................. 27

3 Інструкція користувача ..................................................................... 28

ВИСНОВКИ .......................................................................................... 37

ПЕРЕЛІК ПОСИЛАНЬ.. ...................................................................... 38

Додаток А – Код програми.. ................................................................ 39

# ВСТУП

У роботі Інтерполу одним із найвідповідальніших завдань є коректне ведення картотеки кримінальних елементів, яка відіграє важливу роль під час розшуку злочинців та при розслідуванні справ.

Ведення подібної системи “на папері” є доволі складною та неефективною справою, оскільки інформація може знадобитися у будь-якому місці у будь-який час, також потрібно цю інформацію постійно оновлювати.

Переведення картотеки Інтерполу у електронний вигляд може сильно полегшити життя працівників архіву.

Розроблена програма має на меті прискорення доступу до інформації, постійне підтримання інформації в актуальному стані та зменшення необхідних на підтримку теперішньої системи коштів.

1 СПЕЦИФІКАЦІЯ ПРОГРАМИ

1.1 Функції програми

Програма, що буде розроблена, повинна мати наступні функції:

а) відображення списку злочинців з короткою інформацією про кожного з них;

б) можливість фільтрації списку злочинців за будь-якою множиною параметрів;

в) відображення повної інформації про кожного окремого злочинця; г) додавання нових злочинців до картотеки;

д) видалення злочинців із попереднім запитом дозволу; е) можливість змінити інформацію про злочинця;

ж) відображення списку кримінальних угруповань;

з) відображення повної інформації про кожне окреме угруповання; и) додавання нових кримінальних угруповань;

к) видалення кримінальних угруповань із попереднім запитом дозволу; л) можливість редагувати інформацію про кримінальне угруповання; м) зберігання інформації картотеки до бінарного файлу;

н) зчитування інформації картотеки з бінарного файлу.

1.2 Інтерфейс користувача

Головне вікно програми відображається одразу ж після її запуску

(рисунок 1.1).



Рисунок 1.1 – Головне вікно програми

Інтерфейс программи побудовано таким чином, щоб користувачу було максимально легко та комфортно знаходити необхідну інформацію та оперувати нею. Головне вікно відображає список злочинців, елементи для фільтрації списку, що тематично розділені.

Натискання на будь-який підрозділ фільтру відображає тематично згруповані поля для введення інформації. Можна комбінувати будь-які параметри фільтру (Рисунки 1.2 – 1.8).

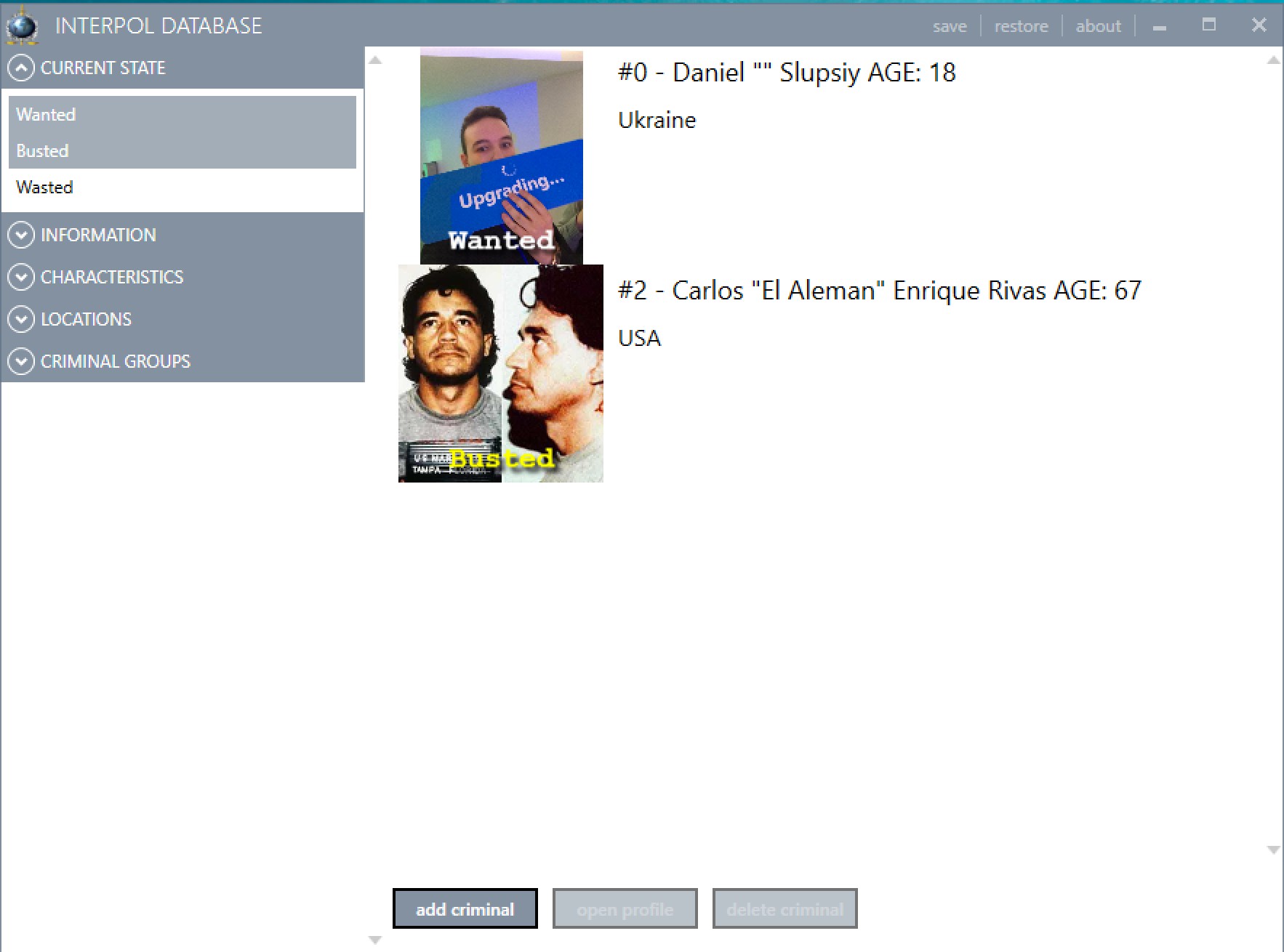


Рисунок 1.2 – Фільтр за статусом злочинця з вибраними параметрами

“Розшукуються” та “Заарештовані”



Рисунок 1.3 – Фільтр за головною інформацією з вказаним полем “Прізвище”

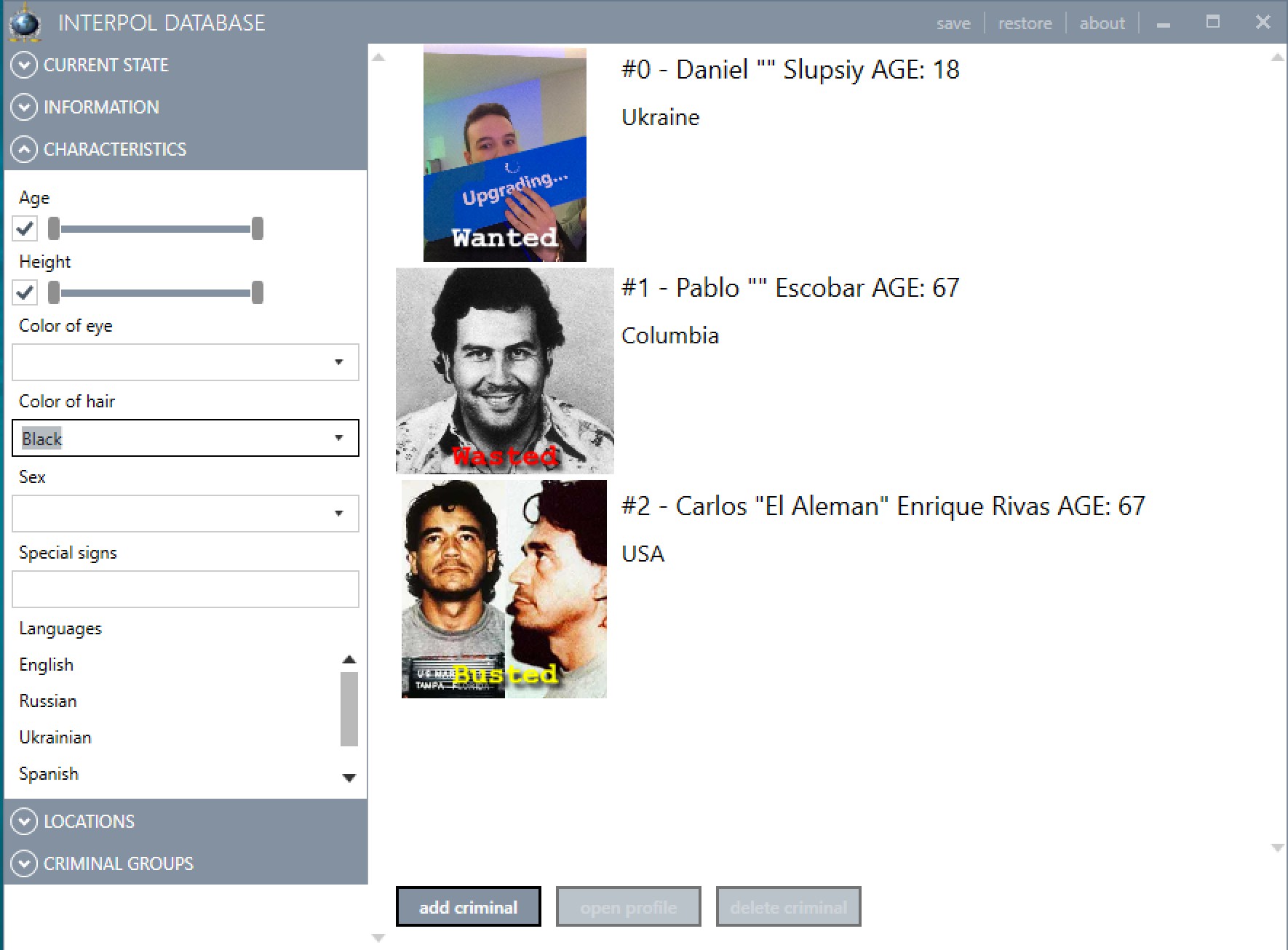


Рисунок 1.4 – Фільтр за характеристикою злочинця з обраним параметром

“Колір волосся”

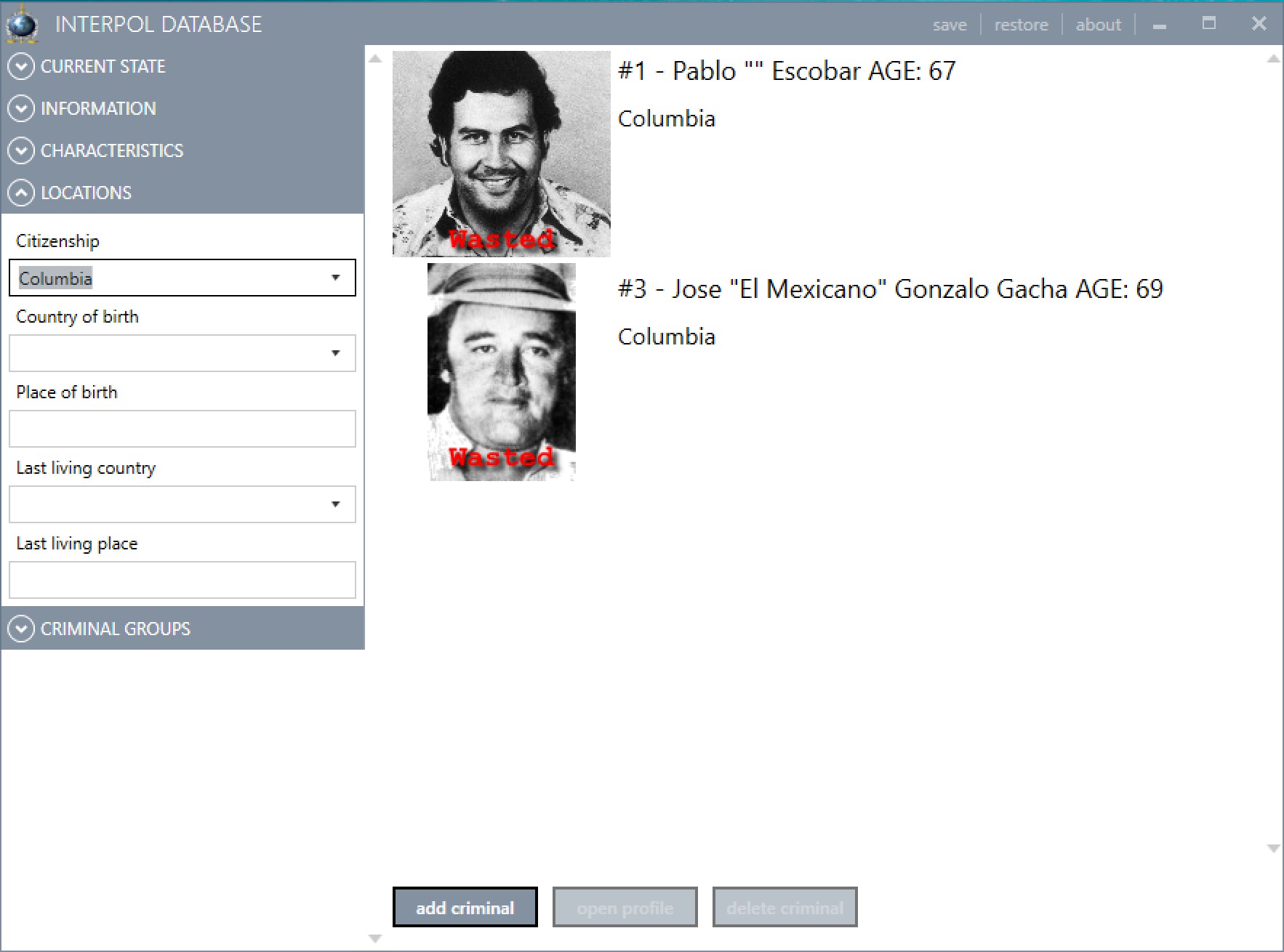


Рисунок 1.5 – Фільтр за локаціями злочинця з обраним параметром

“Громадянство”

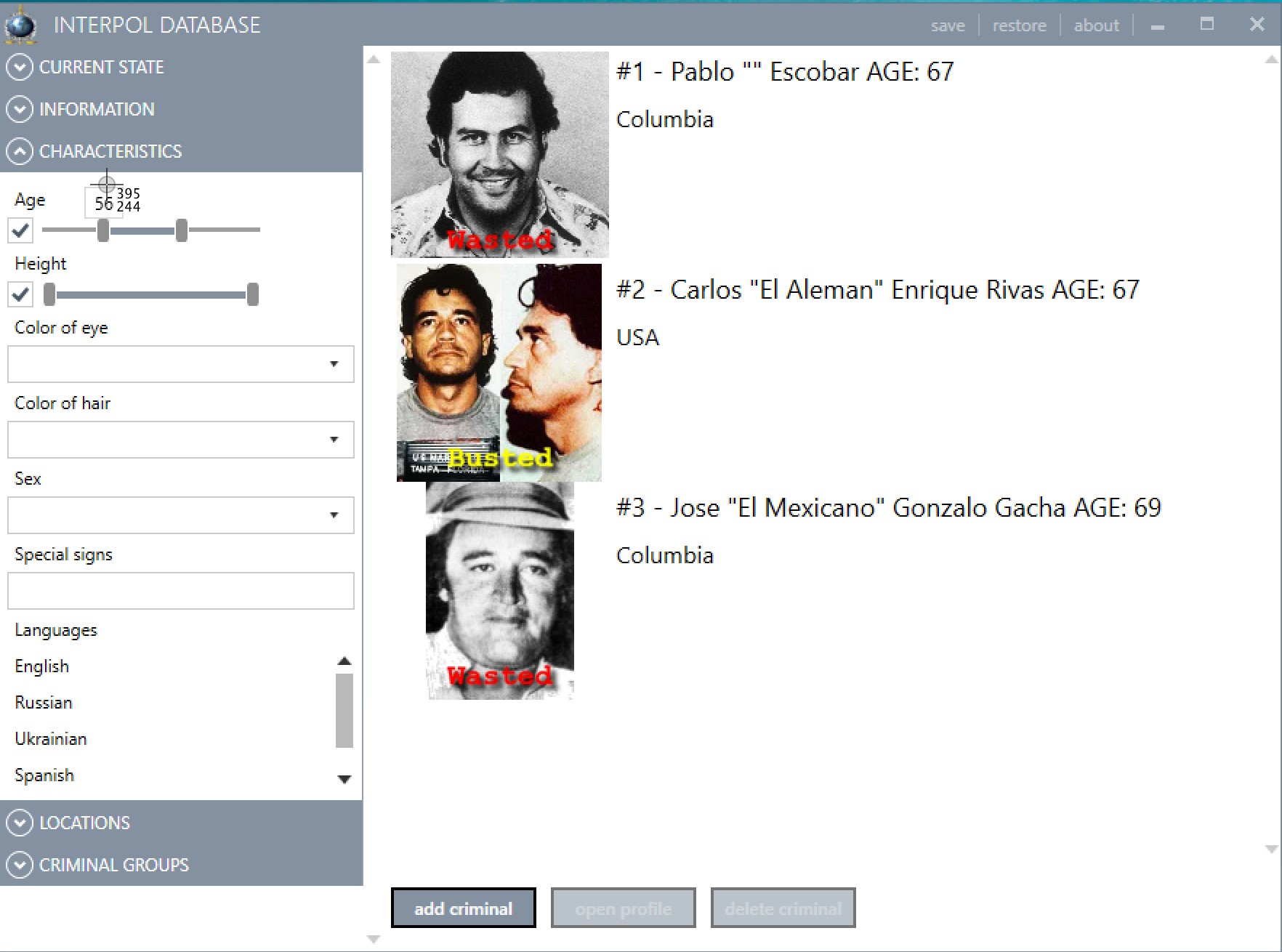


Рисунок 1.6 – Фільтр за характеристиками злочинця з вказаним діапазоном віку

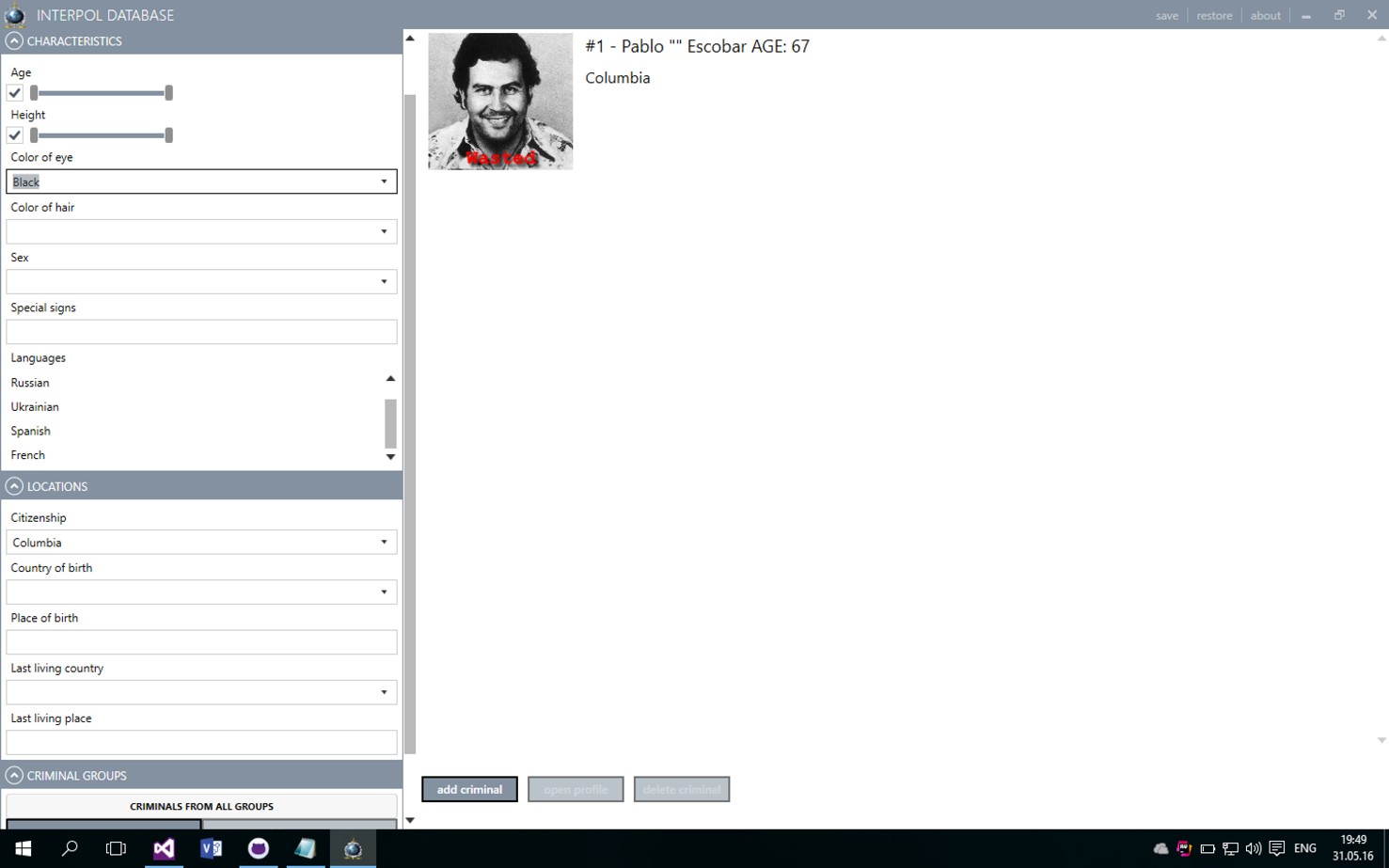


Рисунок 1.7 – Фільтр за обраними параметрами “Колір очей” та “Громадянство”

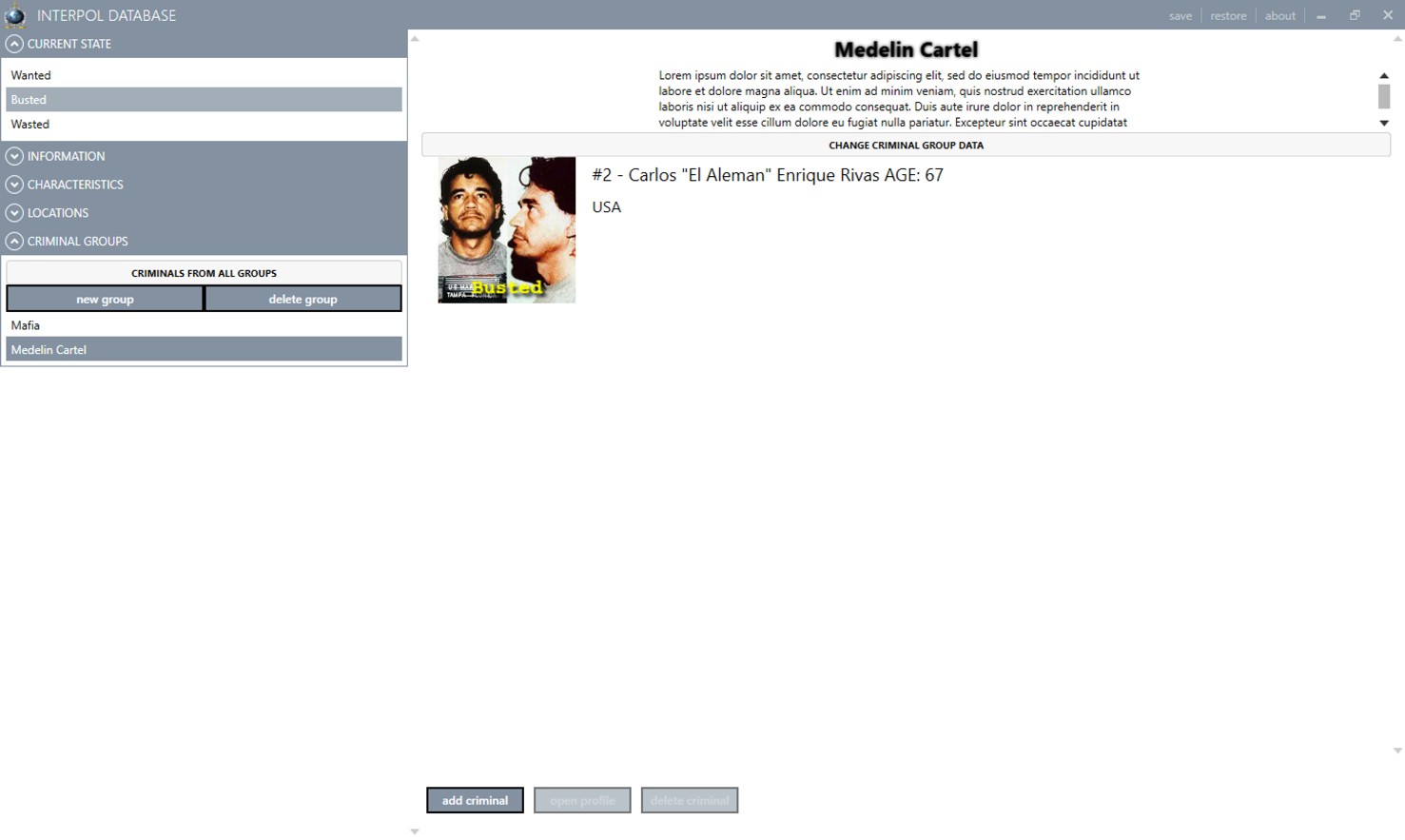


Рисунок 1.8 – Фільтр за обраними параметрами “Заарештований”

та “Член Меделінського картелю”

Як видно з Рисунку 1.8, список угруповань відображається як частина фільтру. При виборі угруповання над списком злочинців відображається назва та додаткова інформація. Над списком угруповань є кнопка скидання вибраного угруповання. Під списком є кнопки для додавання та видалення, остання активна лише тоді, коли користувач вибрав певне угруповання. Після натискання кнопки додавання відображається спеціальний елемент керування Flyout з відповідними полями (Рисунок 1.9).

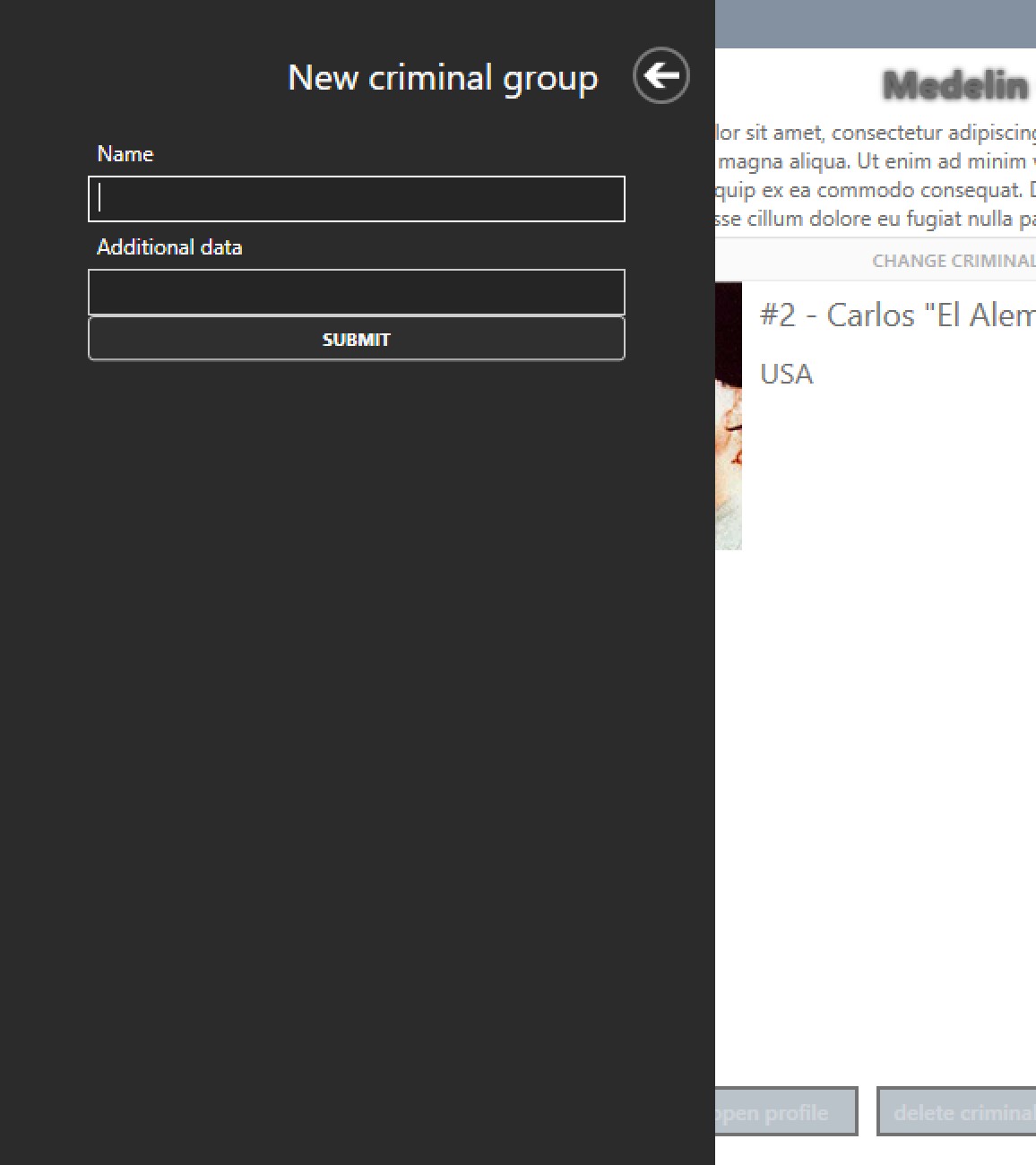


Рисунок 1.9 – Елемент Flyout для додавання нового угруповання

Аналогічний Flyout відображається при натисканні кнопки редагування інформації, тільки з попередньо вказаними даними (Рисунок 1.10).

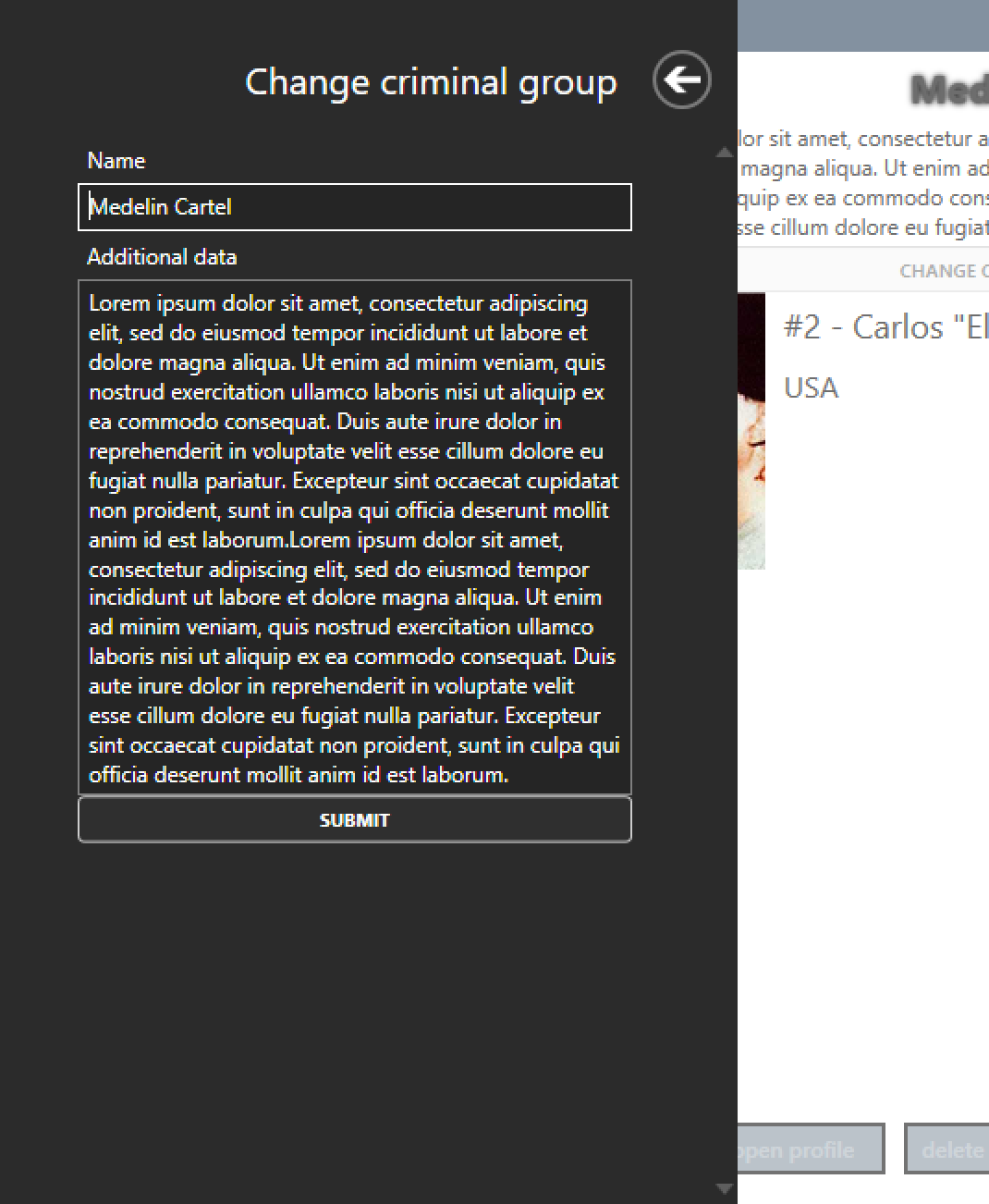


Рисунок 1.10 – Елемент Flyout для редагування інформації про угруповання

# Після натискання кнопки видалення програма потребує від користувача спеціального дозволу від керівництва, і лише після підтвердження наявності такого дозволу видаляє вибране угруповання (Рисунок 1.11).

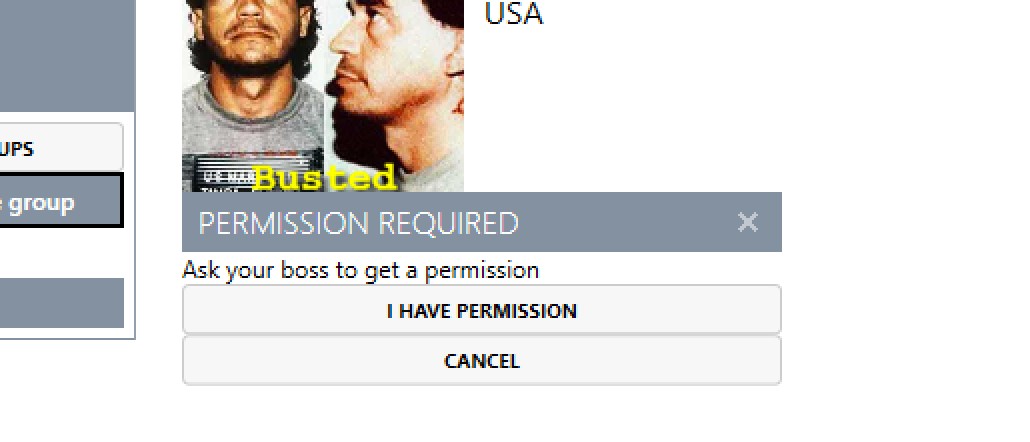


Рисунок 1.11 – Вікно підтвердження наявності дозволу на видалення інформації з архіву

Для додавання нового злочинця до картотеки використовується схожий за виглядом Flyout, але з відповідними полями (Рисунок 1.12). Він відображається при натисканні кнопки додавання під списком злочинців.

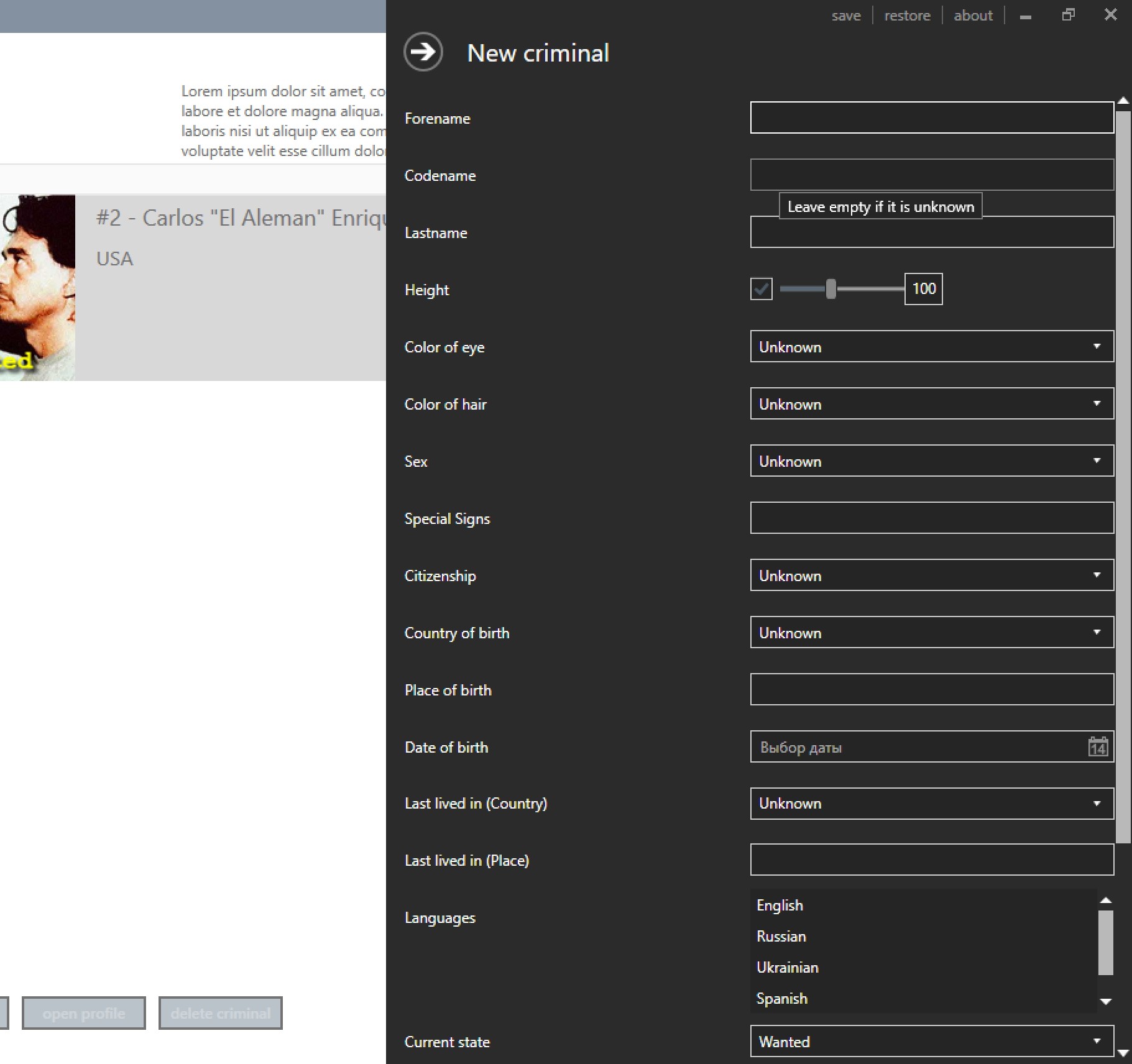


Рисунок 1.12 – Елемент керування Flyout для додавання нового злочинця

При натисканні кнопки видалення злочинця відображається таке ж вікно запиту дозвілу, як і при видаленні угруповання (Рисунок 1.11).

При натисканні кнопки профілю злочинця відображається повна інформація про вибрану персону (Рисунок 1.13). Поля, які не мають значень, позначаються як невідомі.

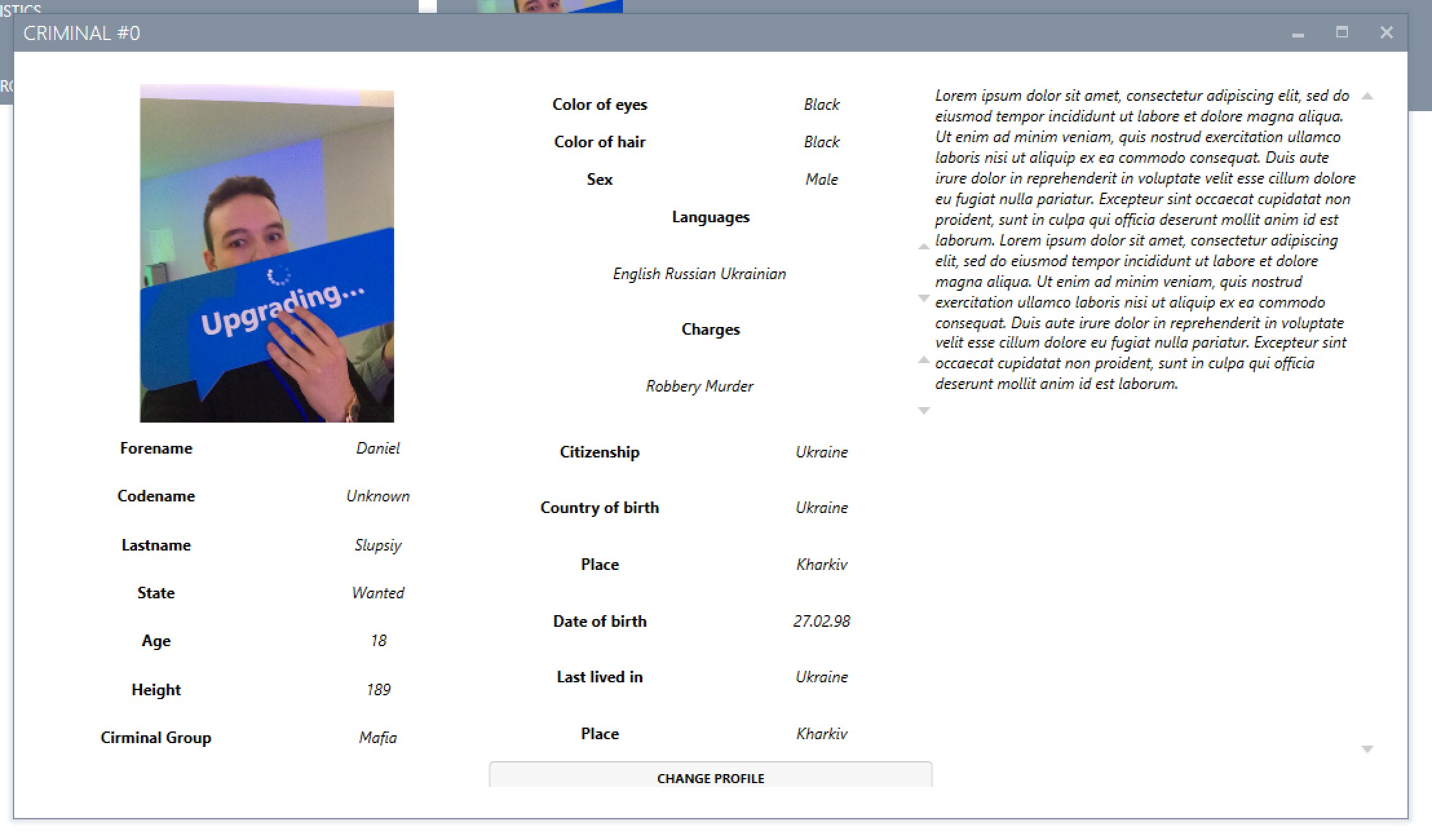


Рисунок 1.13 – Вікно профілю вибраного злочинця

У профілі є кнопка редагування інформації, яка відображає Flyout з попередньо вказаною інформацією (Рисунок 1.14).

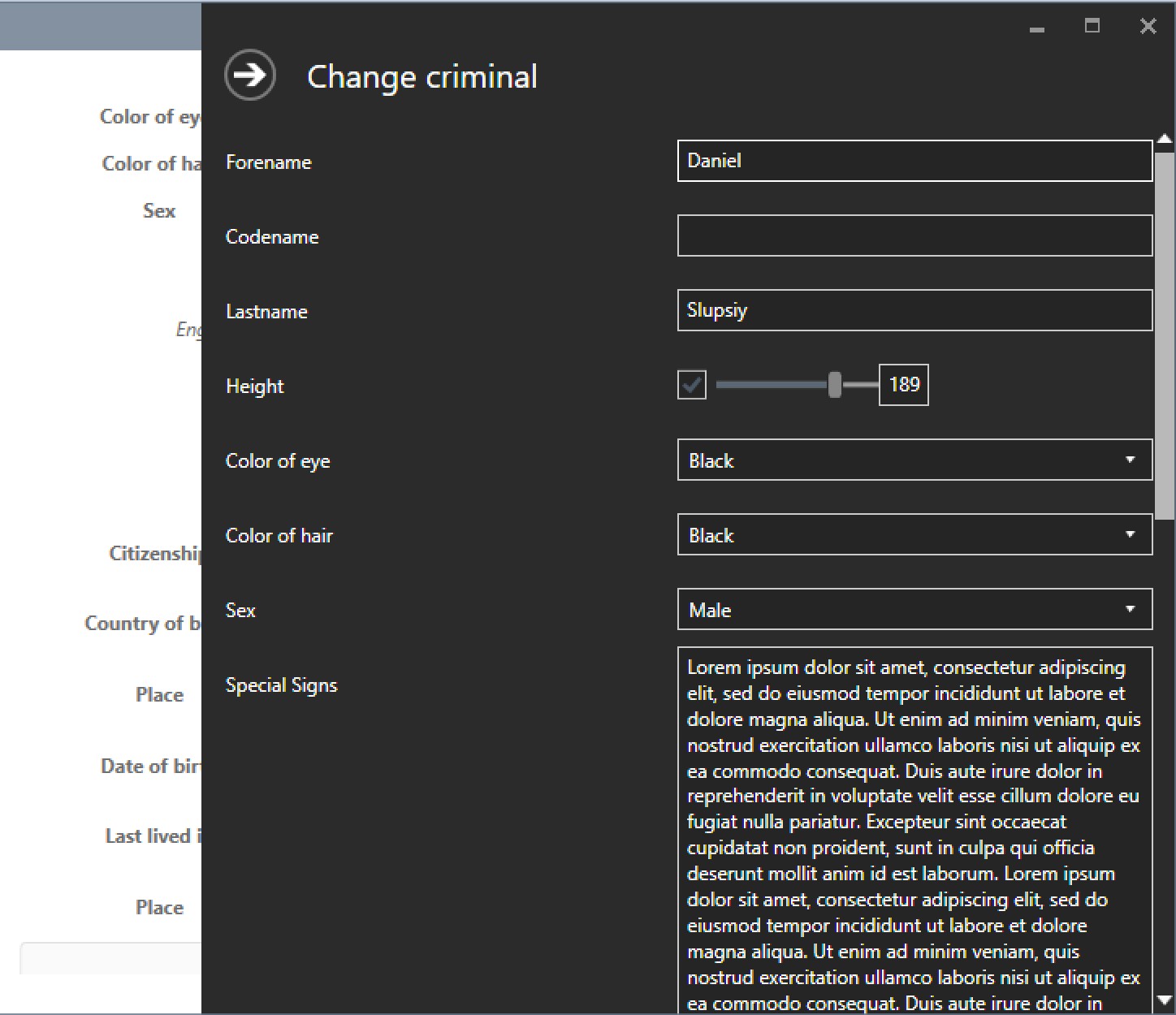


Рисунок 1.14 – Елемент Flyout для редагування профілю злочинця.

Також головне вікно містить меню з 3 полів: відновлення картотеки з файлу, збереження картотеки до файлу та відображення інформації про програму (Рисунок 1.15).



Рисунок 1.15 – Меню головного вікна

Діалогове вікно з інформацією про програму містить завдання до курсового проекту (Рисунок 1.16).

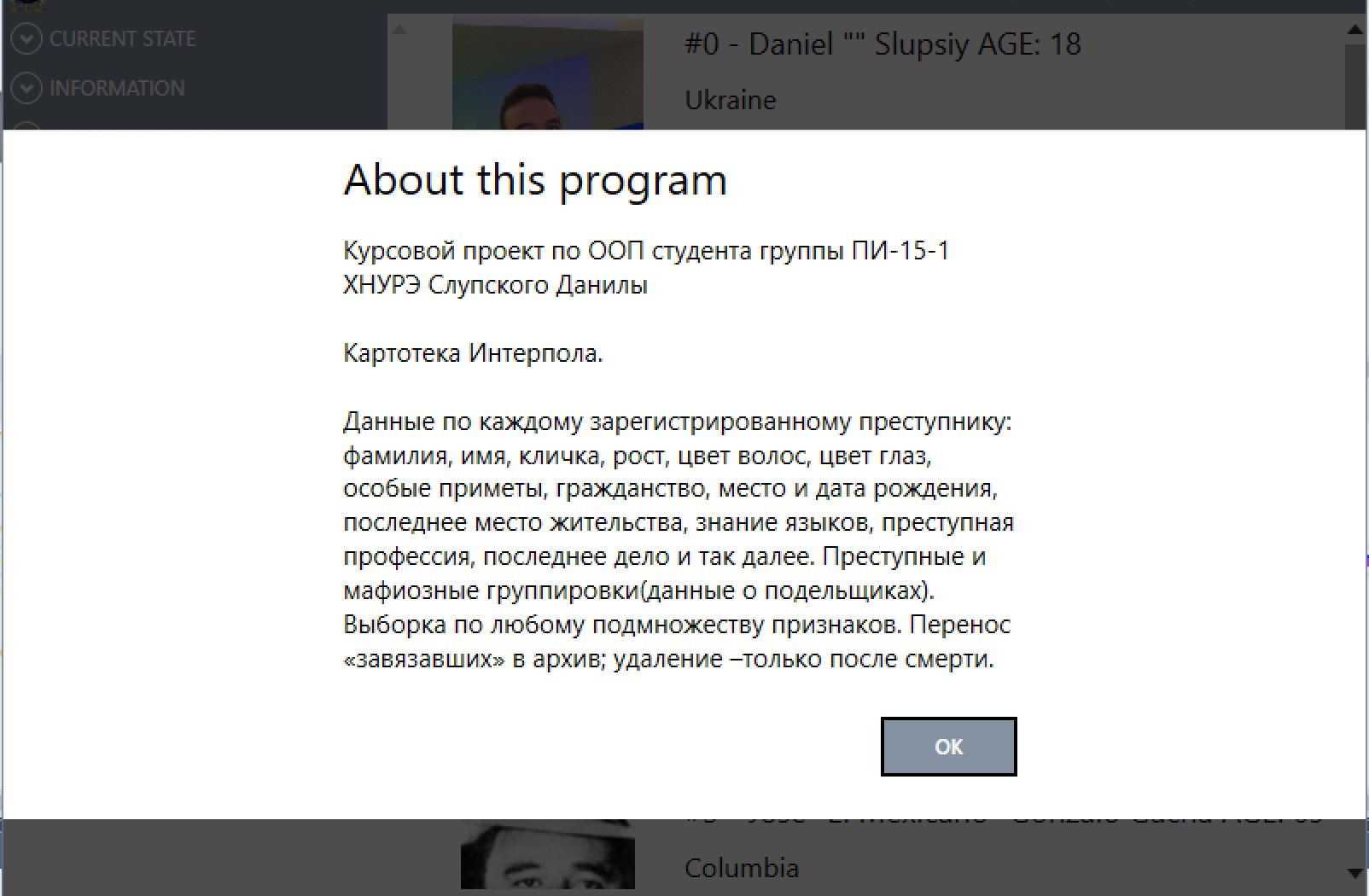


Рисунок 1.16 – Інформація про програму

2 ПРОЕКТНА СПЕЦИФІКАЦІЯ

2.1 Архітектурний стиль

Графічна підсистема Windows Presentation Foundation (WPF) орієнтована на події, більшість часу проходить у очікуванні від користувача певних дій. WPF є складовою Microsoft .NET Framework, починаючи з версії 3.0.

Під час розробки програми використовувалися деякі положення паттерну

MVC, модель програми повністю відокремлена від відображення та контролеру. Для зберігання данних використовується бінарна сериалізація у файл.

2.2 Об'єктна структура програми

Програма складається з набору класів, перерахувань, файлів з даними у бінарному вигляді та спеціальних файлів з розміткою XAML (рисунок 2.1).

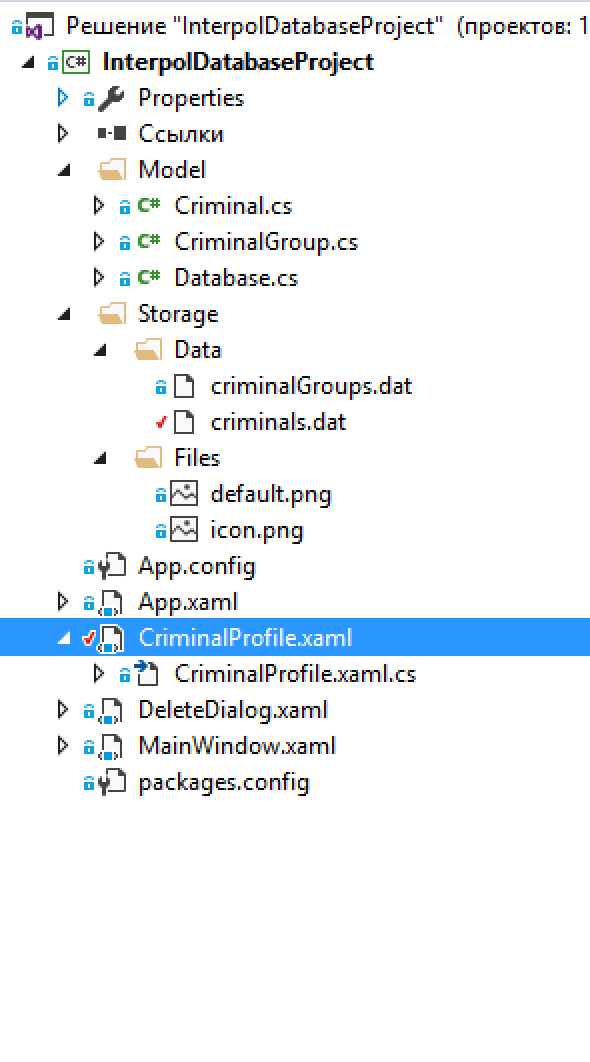


Рисунок 2.1 – Структура проекту Структура проекту відображена на Рисунках 2.2-2.3.

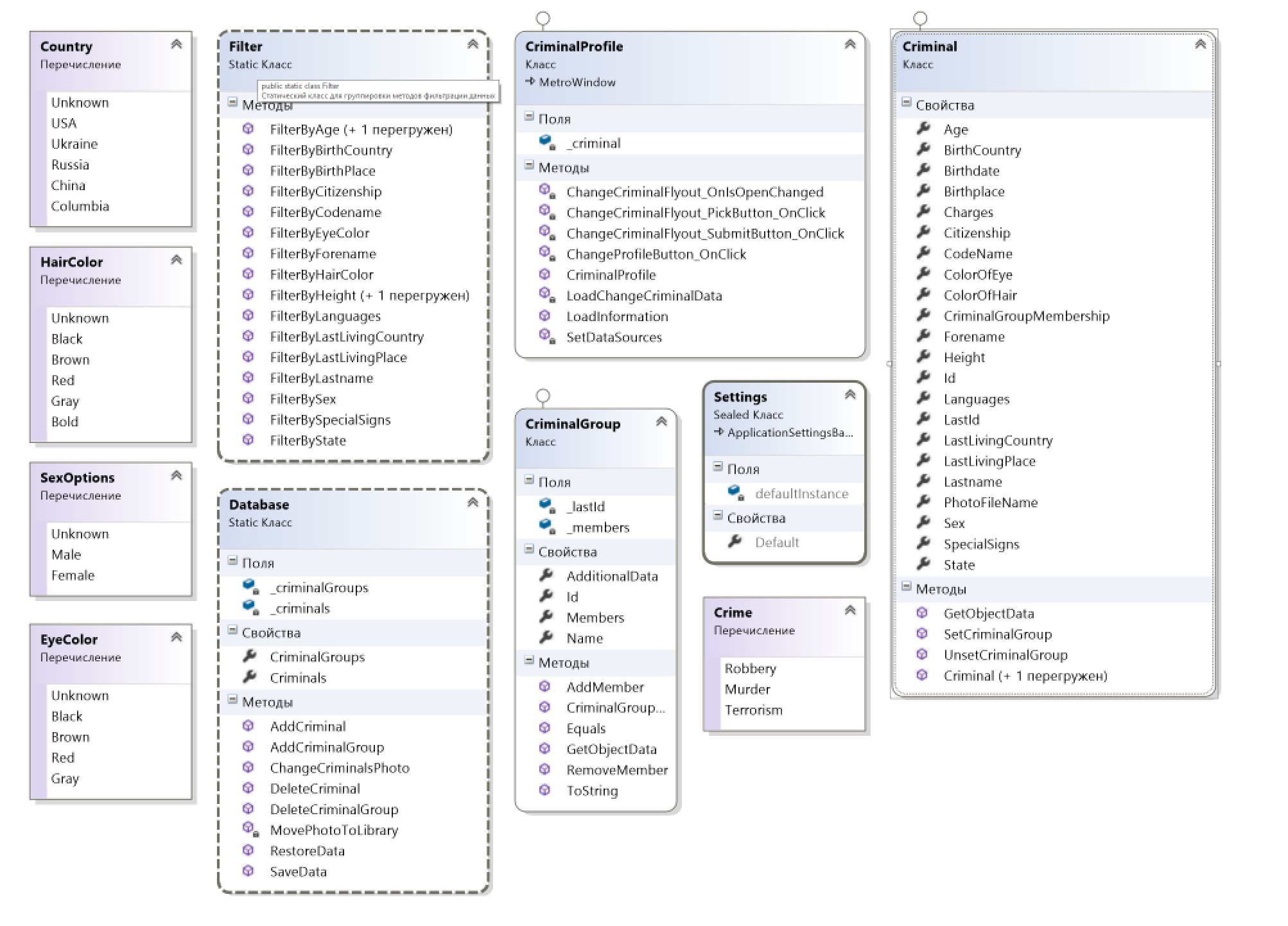


Рисунок 2.2 – UML схема проекту (частина 1)

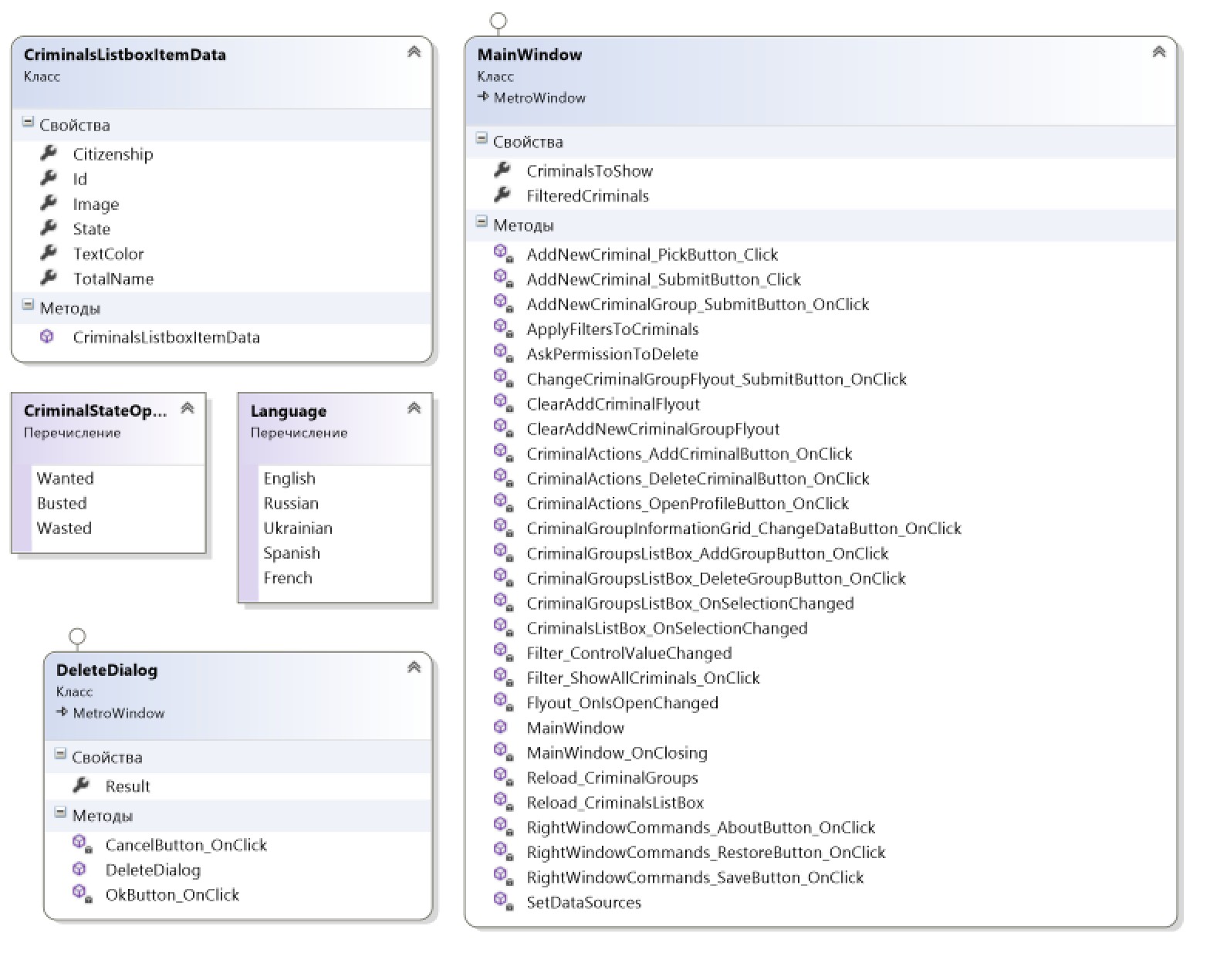


Рисунок 2.3 – UML схема проекту (частина 2)

2.3. Формат даних

Інформація зберігається у бінарному файлі з використанням механізму серіалізації, переведення об’єкта у потік байтів. Для вилучення інформації з файлу використовується зворотній механізм під назвою десеріалізація.

Файли з даними мають формат .dat.

3 ІНСТРУКЦІЯ КОРИСТУВАЧА

Встановлення програми:

а) зклонувати git-репозиторій програми на свій комп'ютер або скопіювати файли програми з фізичного носія;

б) відкрити .exe-файл.

Робота з програмою

Після запуску программи відображається головне вікно, що містить повний список злочинців, наявних у картотеці (Рисунок 3.1).



Рисунок 3.1 – Головне вікно программи

Фільтрування:

а) натиснути на потрібну категорію фільтру; б) ввести відомі дані (Рисунок 3.2).

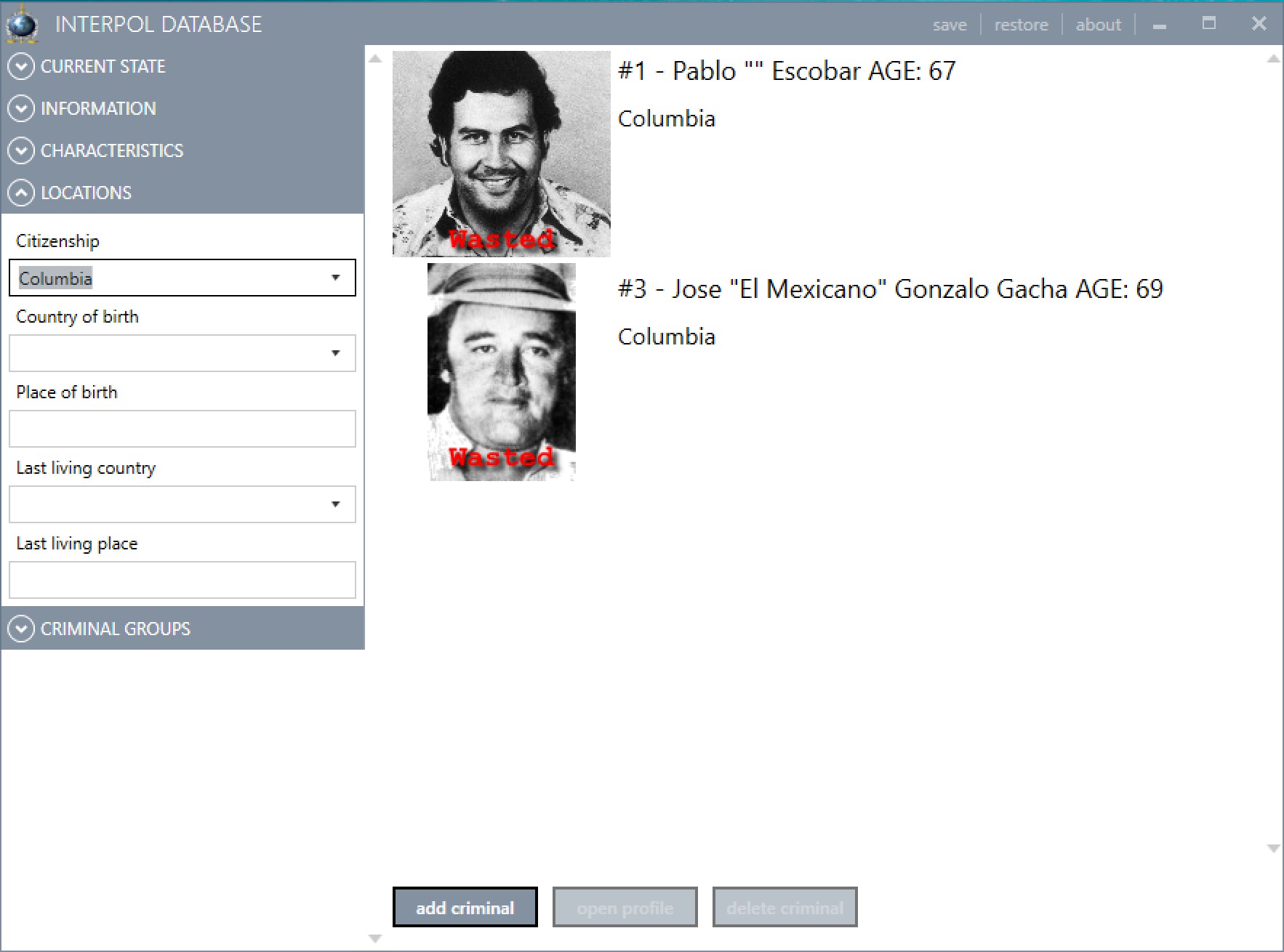


Рисунок 3.2 – Фільтр по локаціям злочинця з обраним параметром

“Громадянство”

Відображення профілю злочинця (Рисунок 3.3):

а) вибрати злочинця із списку;

б) натиснути на кнопку «open profile».

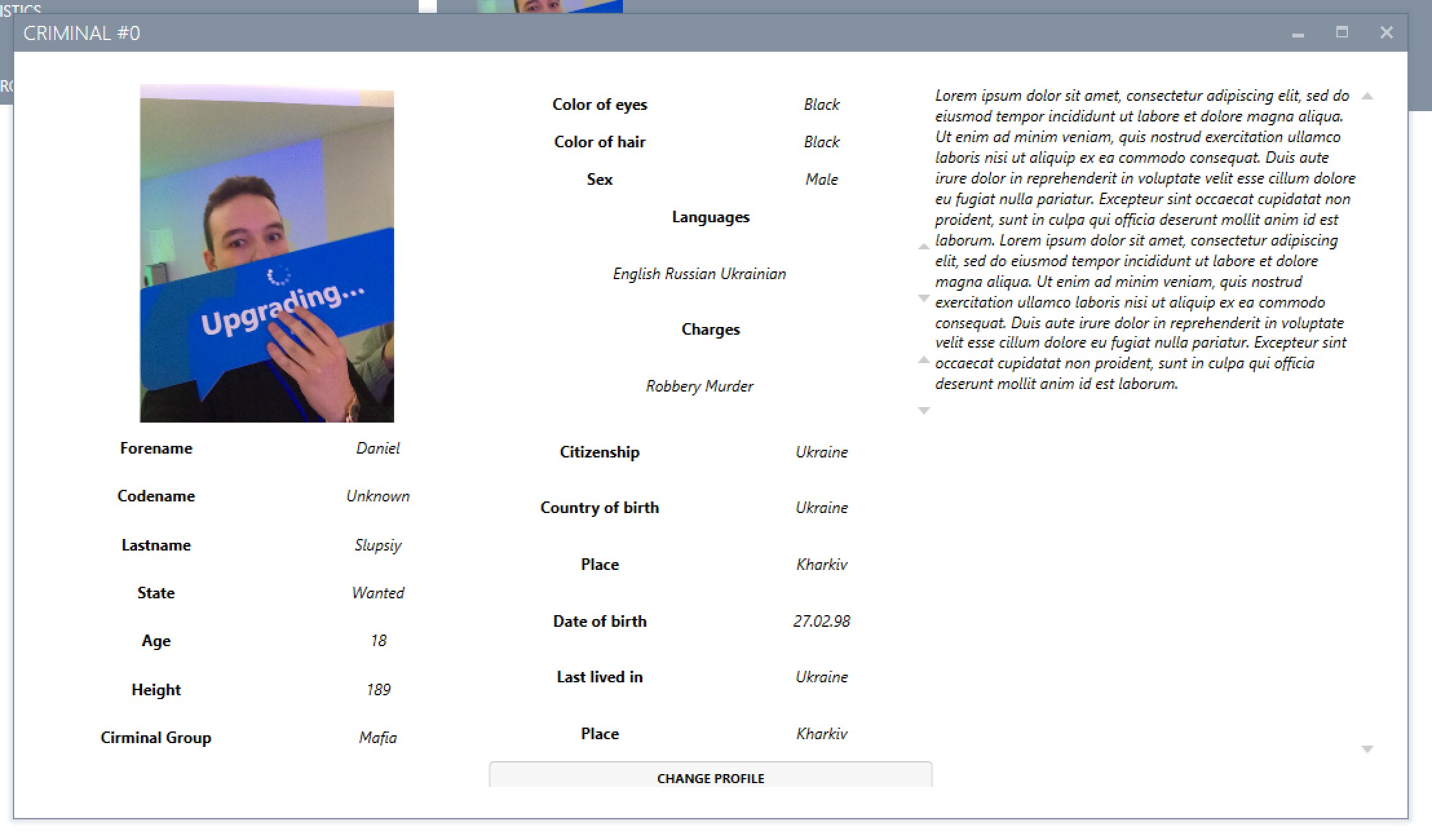


Рисунок 3.3 – Відображення повної інформації про злочинця

Додавання нового злочинця (Рисунок 3.4):

а) натиснути на кнопку «add criminal»;

б) у відкритому вікні ввести усю відому інформацію;

в) натиснути на кнопку «Sumbit» внизу відкритого вікна.

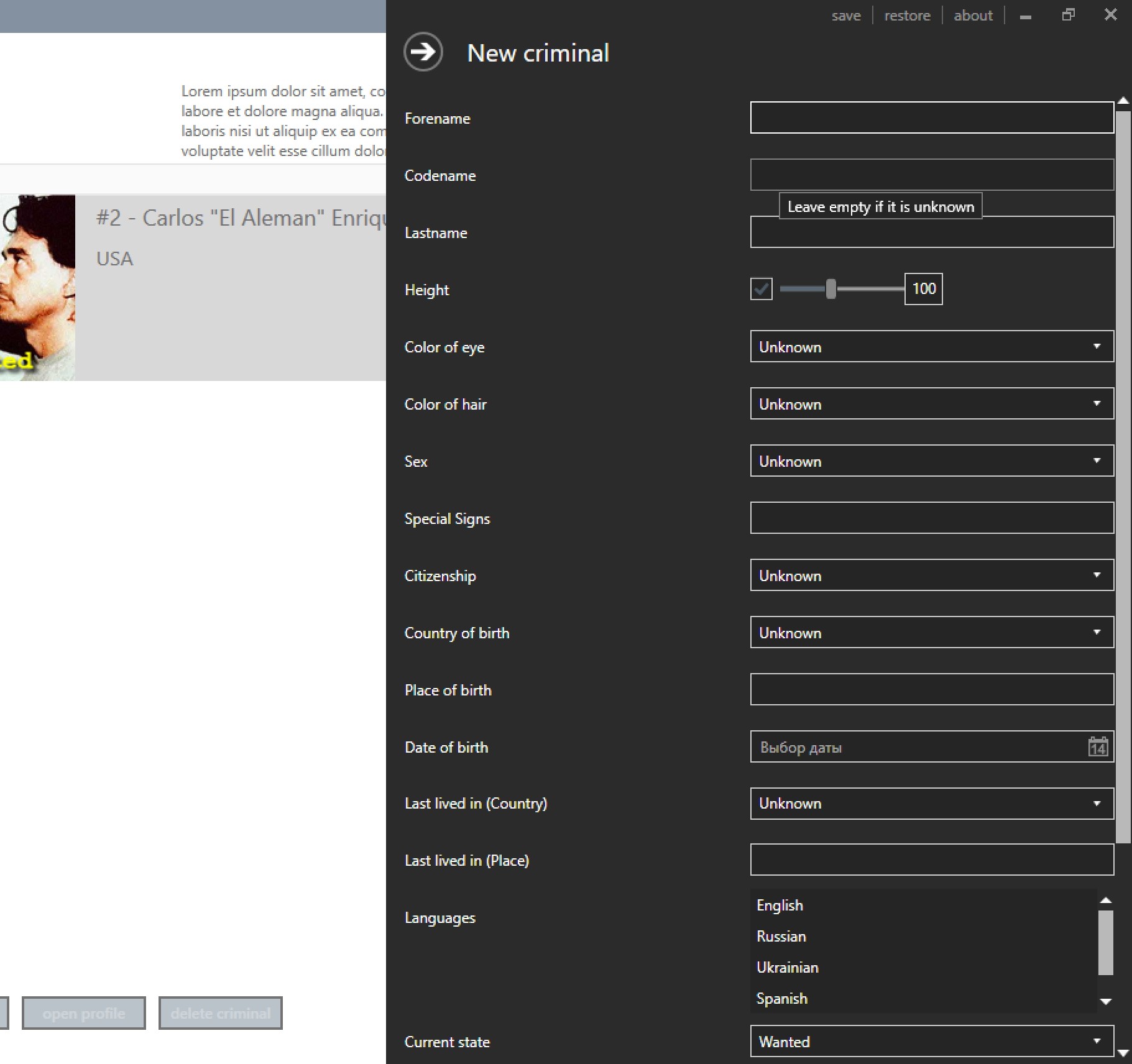


Рисунок 3.4 – Додавання нового злочинця

Видалення злочинця:

а) натиснути на кнопку «delete criminal»;

б) у відкритому вікні підтвердити наявність дозволу на видалення.

Редагування профілю злочинця:

а) відкрити профіль злочинця (інструкція була наведена вище); б) у вікні профілю натиснути на кнопку «CHANGE PROFILE»;

в) у новому відкритому вікні (Рисунок 3.5) ввести нові дані та натиснути на кнопку «Submit».

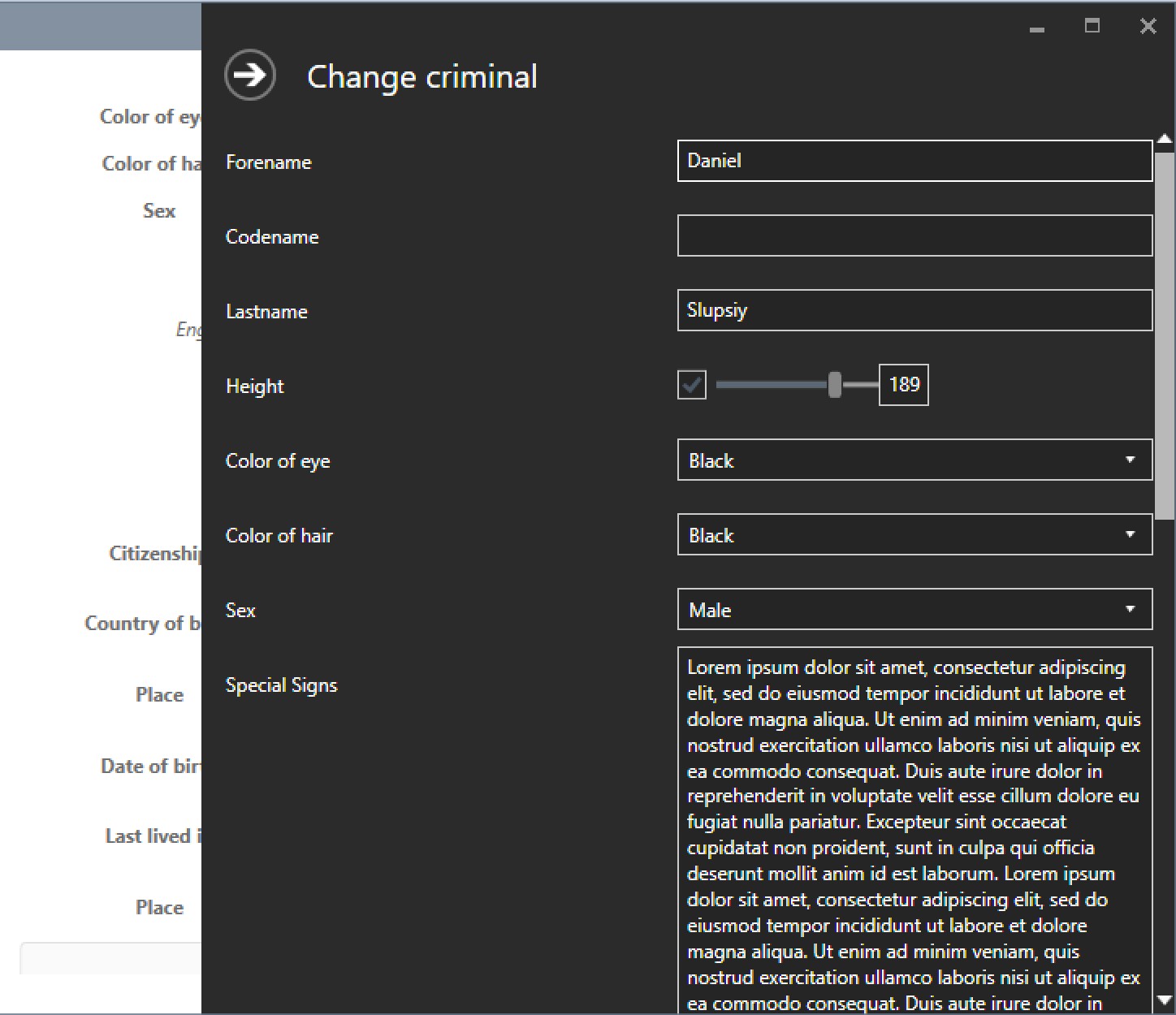


Рисунок 3.5 – Редагування профілю злочинця

# Відображення списку угруповань:

а) відкрити головне вікно программи (Рисунок 3.1);

б) відкрити розділ «CRIMINAL GROUPS», що знаходиться зліва від списку злочинців (Рисунок 3.6).

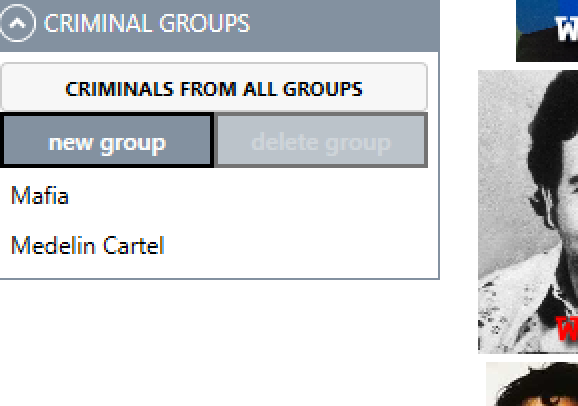


Рисунок 3.6 – Відображення списку угруповань Відображення інформації про вибране угруповання:

а) відобразити список угруповань (попередній пункт);

б) вибрати угруповання, інформація відобразиться над списком злочинців

(Рисунок 3.7).



Рисунок 3.7 – Відображення інформації про вибране угруповання

# Додавання нового угруповання:

а) відобразити список угруповань (описано раніше); б) натиснути на кнопку «new group»;

в) у відкритому вікні ввести відомі дані (Рисунок 3.8);

г) натиснути «Submit».

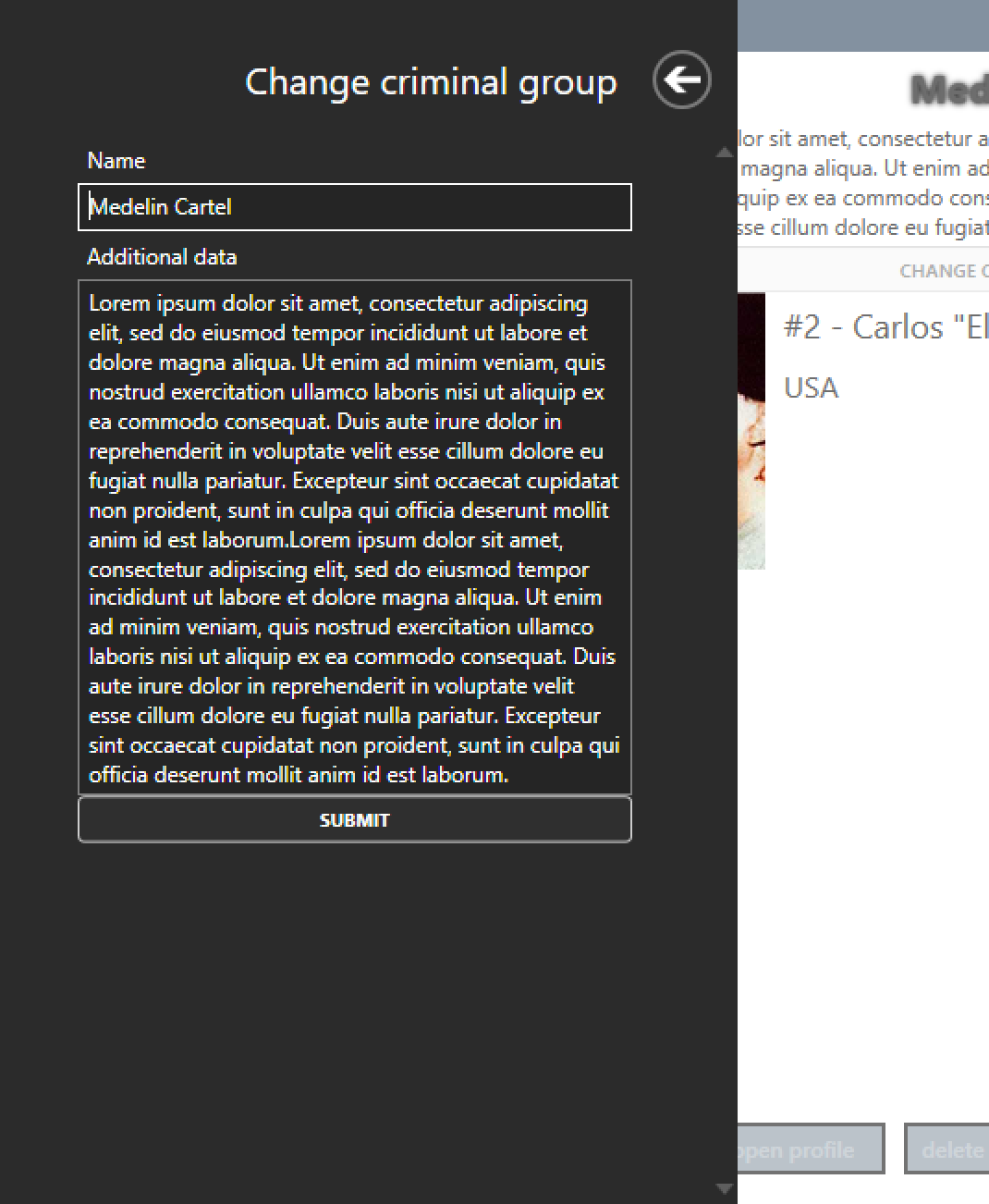


Рисунок 3.8 – Додавання нового угруповання

Видалення угруповання:

а) відобразити список угруповань (описано раніше); б) натиснути на кнопку «delete group» (Рисунок 3.6); в) підтвердити наявність дозволу на видалення.

Редагування інформації про угруповання:

а) зберігання / зчитування картотеки до/з файлу:

б) відкрити головне вікно программи (описано раніше);

в) натиснути на кнопку “save”/”restore” у верхньому меню (Рисунок 3.9)



Рисунок 3.9 – Меню головного вікна

ВИСНОВКИ

Під час розробки та супроводження програмної системи я пройшов всі етапи розробки програми: від постановки завдання до налагодження готового продукту.

У результаті розробки проекту була створена програма «Картотека Інтерполу» на платформі .NET з використанням мови програмування C# та графічної підсистеми Windows Presentation Foundation.

Програма реалізує усі функції, наведені у Розділі 1.1.

Під час роботи над проектом я розглянув наступні можливості платформи

.Net: LINQ, бінарна серіалізація, перерахування, події, проектування програмних інтерфейсів за допомогою Windows Presentation Foundation. Також я отримав досвід у проектуванні, реалізації, документуванні програмного продукту.

ПЕРЕЛІК ПОСИЛАНЬ

1. Бондарев В. М. «Объектно-ориентированное программирование на С#» [Текст]: учеб. пособ / В. М. Бондарев. – Х.: Смит, 2009. – 224 с.

2. Ерохін О. C# 5.0 и платформа .NET 4.5 [Електронний ресурс] / Олександр Ерохін. – 2015. – Режим доступу до ресурсу: [http://professorweb.ru/my/csharp/charp\_theory/level1/infocsharp.php.](http://professorweb.ru/my/csharp/charp_theory/level1/infocsharp.php)

1. Ерохін О. WPF - Windows Presentation Foundation [Електронний ресурс] / Олександр Ерохін. – 2015. – Режим доступу до ресурсу: [http://professorweb.ru/my/WPF/base\_WPF/level1/info\_WPF.php.](http://professorweb.ru/my/WPF/base_WPF/level1/info_WPF.php)

4. Ерохін О. Expression Blend 4 [Електронний ресурс] / Олександр Ерохін. –

2014. – Режим доступу до ресурсу: [http://professorweb.ru/my/expression\_blend/base/level1/info\_eb4.php.](http://professorweb.ru/my/expression_blend/base/level1/info_eb4.php)

5. Ерохін О. LINQ - язык интегрированных запросов [Електронний ресурс] / Олександр Ерохін – Режим доступу до ресурсу: [http://professorweb.ru/my/LINQ/base/level1/info\_linq.php.](http://professorweb.ru/my/LINQ/base/level1/info_linq.php)

6. Троелсен: «Мова програмування С# 2005 та платформа .NET 2.0» [Текст]: учеб. пособ, 3е видання. : Пер. з англ. – Москва. : ООО «И.Д. Вільямс», 2007. – 1168с.

Додаток А – Код програми

Цей додаток містить лістинги наступних файлів у зазначеному порядку:

* 1. Criminal.cs
  2. CriminalGroup.cs
  3. Database.cs
  4. CriminalProfile.xaml
  5. CriminalProfile.xaml.cs
  6. DeleteDialog.xaml
  7. DeleteDialog.xaml.cs
  8. MainWindow.xaml
  9. MainWindow.xaml.cs

1. using System;
2. using System.Collections.Generic;
3. using System.Runtime.Serialization; 4

5 namespace InterpolDatabaseProject.Model 6 {

7

1. #region Enums
2. /// <summary>

10 /// Пол

1. /// </summary>
2. public enum SexOptions 13 {
3. Unknown,
4. Male,
5. Female

17 }

18

19 /// <summary>

20 /// Статус преступника

1. /// </summary>
2. public enum CriminalStateOptions 23 {
3. Wanted,
4. Busted,
5. Wasted

27 }

28

29 /// <summary>

30 /// Виды преступлений

1. /// </summary>
2. public enum Crime

33 {

1. Robbery,
2. Murder,
3. Terrorism 37 }

38

39 /// <summary>

40 /// Страна

1. /// </summary>
2. public enum Country

43 {

1. Unknown,
2. USA,
3. Ukraine,
4. Russia,
5. China,
6. Columbia 50 }

51

52 /// <summary>

53 /// Цвет глаз

1. /// </summary>
2. public enum EyeColor

56 {

57

58

59

60

61

62 }

63

Unknown, Black, Brown, Red, Gray

64 /// <summary>

65 /// Цвет волос

1. /// </summary>
2. public enum HairColor 68 {
3. Unknown,
4. Black,
5. Brown,
6. Red,
7. Gray,
8. Bold

75 }

76

77 /// <summary>

78 /// Язык

1. /// </summary>
2. public enum Language

81 {

1. English,
2. Russian,
3. Ukrainian,
4. Spanish,
5. French

87 }

88 #endregion 89

1. [Serializable]
2. public class Сriminal : ISerializable 92 {
3. #region Fields
4. /// <summary>

95 /// Отображает последний использованный для преступника ID

1. /// </summary>
2. public static int LastId { get; private set; } = -1;
3. #endregion
4. #region Constructors
5. /// <summary>

101 /// Конструктор для десериализации

1. /// </summary>
2. public Сriminal(SerializationInfo info, StreamingContext context)

104 {

1. LastId = info.GetInt32("static.\_lastId");
2. Id = info.GetInt32("Id");
3. Lastname = info.GetString("Lastname");
4. CodeName = info.GetString("CodeName");
5. Forename = info.GetString("Forename");
6. Height = (int?)info.GetValue("Height", typeof(int?));
7. ColorOfEye = (EyeColor)info.GetValue("ColorOfEye", typeof (EyeColor));

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126 }

127

ColorOfHair = (HairColor)info.GetValue("ColorOfHair", typeof (HairColor));

Sex = (SexOptions)info.GetValue("Sex", typeof(SexOptions));

SpecialSigns = (string)info.GetValue("SpecialSigns", typeof (string));

Citizenship = (Country)info.GetValue("Citizenship", typeof (Country));

BirthCountry = (Country)info.GetValue("BirthCountry", typeof (Country));

Birthplace = info.GetString("Birthplace");

Birthdate = (DateTime?)info.GetValue("Birthdate", typeof (DateTime?));

LastLivingCountry = (Country)info.GetValue("LastLivingCountry", typeof(Country));

LastLivingPlace = info.GetString("LastLivingPlace");

Languages = (List<Language>)info.GetValue("Languages", typeof (List<Language>));

State = (CriminalStateOptions)info.GetValue("State", typeof (CriminalStateOptions));

CriminalGroupMembership = (CriminalGroup)info.GetValue ("CriminalGroupMembership", typeof(CriminalGroup));

PhotoFileName = info.GetString("PhotoFileName");

Charges = (List<Crime>)info.GetValue("Charges", typeof (List<Crime>));

128 /// <summary>

129 /// Основной конструктор

130 /// </summary>

131 /// <param name="lastname">Фамилия преступника</param> 132 /// <param name="forename">Имя преступника</param> 133 /// <param name="codeName">Кличка преступника</param> 134 /// <param name="height">Рост преступника</param>

135 /// <param name="colorOfEye">Цвет глаз преступника</param> 136 /// <param name="colorOfHair">Цвет волос преступника</param> 137 /// <param name="sex">Пол преступника</param>

138 /// <param name="specialSigns">Особые приметы преступника, дополнительные данные</param>

139 /// <param name="citizenship">Гражданство преступника</param>

140 /// <param name="birthCountry">Страна рождения преступника</param> 141 /// <param name="birthplace">Место рождения преступника</param> 142 /// <param name="birthdate">Дата рождения преступника</param>

143 /// <param name="lastLivingCountry">Последняя зарегистрированная страна проживания преступника</param>

144 /// <param name="lastLivingPlace">Последнее зарегистрированное место проживания преступника</param>

145 /// <param name="languages">Языки, которыми владеет преступник</param> 146 /// <param name="state">Текущее состояние преступника</param>

147 /// <param name="photoFileName">Название файла с фотографией преступника</param>

148 /// <param name="criminalGroup">Принадлежность преступника к криминальной группировке</param>

149 /// <param name="charges">Преступления, в которых обвиняется преступник</param>

150 public Сriminal(string lastname, string forename, string codeName, int? height,

151

152

153

154

155 {

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177 }

EyeColor colorOfEye, HairColor colorOfHair, SexOptions sex, string specialSigns,

Country citizenship, Country birthCountry, string birthplace, DateTime? birthdate,

Country lastLivingCountry, string lastLivingPlace, List<Language> languages, CriminalStateOptions state, string photoFileName,

CriminalGroup criminalGroup, List<Crime> charges)

Id = ++LastId; Lastname = lastname; Forename = forename; CodeName = codeName; Height = height;

ColorOfEye = colorOfEye; ColorOfHair = colorOfHair; Sex = sex;

SpecialSigns = specialSigns; Citizenship = citizenship; BirthCountry = birthCountry; Birthplace = birthplace; Birthdate = birthdate;

LastLivingCountry = lastLivingCountry; LastLivingPlace = lastLivingPlace; Languages = languages;

State = state;

if (criminalGroup != null) SetCriminalGroup(criminalGroup);

PhotoFileName = photoFileName; Charges = charges;

1. #endregion
2. #region Properies 180

181 /// <summary>

182 /// Уникальный номер преступника

1. /// </summary>
2. public int Id { get; set; } 185

186 /// <summary>

187 /// Фамилия преступника

1. /// </summary>
2. public string Lastname { get; set; } 190

191 /// <summary>

192 /// Имя преступника

1. /// </summary>
2. public string Forename { get; set; } 195

196 /// <summary>

197 /// Кличка преступника

1. /// </summary>
2. public string CodeName { get; set; } 200

201 /// <summary>

202 /// Рост преступника

203 /// </summary>

204 public int? Height { get; set; } 205

206 /// <summary>

207 /// Цвет глаз преступника

1. /// </summary>
2. public EyeColor ColorOfEye { get; set; } 210

211 /// <summary>

212 /// Цвет волос преступника

1. /// </summary>
2. public HairColor ColorOfHair { get; set; } 215

216 /// <summary>

217 /// Пол преступника

1. /// </summary>
2. public SexOptions Sex { get; set; } 220

221 /// <summary>

222 /// Особые признаки, дополнительные данные

1. /// </summary>
2. public string SpecialSigns { get; set; } 225

226 /// <summary>

227 /// Гражданство преступника

1. /// </summary>
2. public Country Citizenship { get; set; } 230

231 /// <summary>

232 /// Страна рождения преступника

1. /// </summary>
2. public Country BirthCountry { get; set; } 235

236 /// <summary>

237 /// Место рождения

1. /// </summary>
2. public string Birthplace { get; set; } 240

241 /// <summary>

242 /// Дата рождения

1. /// </summary>
2. public DateTime? Birthdate { get; set; } 245

246 /// <summary>

247 /// Последняя зарегистрированная страна проживания

1. /// </summary>
2. public Country LastLivingCountry { get; set; } 250

251 /// <summary>

252 /// Последнее зарегистрированное место проживания

1. /// </summary>
2. public string LastLivingPlace { get; set; } 255

256 /// <summary>

257 /// Список языков, на которых говорит преступник

1. /// </summary>
2. public List<Language> Languages { get; set; }

|  |  |  |
| --- | --- | --- |
| 260 |  | |
| 261 | /// | <summary> |
| 262 | /// | Текущий статус преступника |
| 263 | /// | </summary> |
| 264 public CriminalStateOptions State { get; set; } 265 | | |
| 266 | /// | <summary> |
| 267 | /// | Членство в преступной группировке |
| 268 | /// | </summary> |
| 269 public CriminalGroup CriminalGroupMembership { get; private set; } 270 | | |
| 271 | /// | <summary> |
| 272 | /// | Имя файла с фотографией преступника |
| 273 | /// | </summary> |
| 274 public string PhotoFileName { get; set; } 275 | | |
| 276 | /// | <summary> |
| 277 | /// | Обвинения в преступлениях |
| 278 | /// | </summary> |

279 public List<Crime> Charges { get; set; } 280

281 /// <summary>

282 /// Текущий возраст преступника

1. /// </summary>
2. public int? Age

285 {

286 get

287 {

1. if (Birthdate == null) return null;
2. return DateTime.Today.Year - Birthdate.Value.Year;

290 }

291 }

1. #endregion
2. #region Methods
3. /// <summary>

295 /// Метод для установления членства в преступной группировке.

296 /// Необходим для синхронизации коллекции членов группировки

1. /// и свойства CriminalGroupMembership
2. /// </summary>

299 /// <param name="criminalGroup">Группировка, членом которой является преступник</param>

1. public void SetCriminalGroup(CriminalGroup criminalGroup) {
2. if (CriminalGroupMembership != null) CriminalGroupMembership = null;
3. if (!criminalGroup.Members.ContainsKey(Id))
4. criminalGroup.AddMember(this);
5. else CriminalGroupMembership = criminalGroup;

305 }

306

307 /// <summary>

308 /// Метод для удаления членства в преступной группировке.

309 /// Необходим для синхронизации коллекции членов группировки

1. /// и свойства CriminalGroupMembership
2. /// </summary>
3. public void UnsetCriminalGroup()

313 {

|  |  |  |
| --- | --- | --- |
| 314 |  | if (CriminalGroupMembership==null) return; |
| 315 |  | if (CriminalGroupMembership.Members.ContainsKey(Id)) |
| 316 |  | CriminalGroupMembership.RemoveMember(Id); |
| 317 |  | else CriminalGroupMembership = null; |
| 318 | } |  |
| 319 |  |  |
| 320 | /// | <summary> |
| 321 | /// | Заполняет SerializationInfo данными, необходимыми для сериализации |
| 322 | /// | </summary> |

323 public void GetObjectData(SerializationInfo info, StreamingContext context)

324 {

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347 }

info.AddValue("static.\_lastId", LastId, typeof(int)); info.AddValue("Id", Id, typeof(int));

info.AddValue("Lastname", Lastname, typeof(string)); info.AddValue("CodeName", CodeName, typeof(string)); info.AddValue("Forename", Forename, typeof(string)); info.AddValue("Height", Height, typeof(int?)); info.AddValue("ColorOfEye", ColorOfEye, typeof(EyeColor)); info.AddValue("ColorOfHair", ColorOfHair, typeof(HairColor)); info.AddValue("Sex", Sex, typeof(SexOptions)); info.AddValue("SpecialSigns", SpecialSigns, typeof(List<string>)); info.AddValue("Citizenship", Citizenship, typeof(Country)); info.AddValue("BirthCountry", BirthCountry, typeof(Country)); info.AddValue("Birthplace", Birthplace, typeof(string)); info.AddValue("Birthdate", Birthdate, typeof(DateTime?)); info.AddValue("LastLivingCountry", LastLivingCountry, typeof

(Country));

info.AddValue("LastLivingPlace", LastLivingPlace, typeof(string)); info.AddValue("Languages", Languages, typeof(List<Language>)); info.AddValue("State", State, typeof(CriminalStateOptions)); info.AddValue("CriminalGroupMembership", CriminalGroupMembership,

typeof(CriminalGroup));

info.AddValue("PhotoFileName", PhotoFileName, typeof(string)); info.AddValue("Charges", Charges, typeof(List<Crime>));

348

349

350 } 351

#endregion

}

1. using System;
2. using System.Collections.Generic;
3. using System.Collections.ObjectModel;
4. using System.Runtime.Serialization; 5

6 namespace InterpolDatabaseProject.Model 7 {

1. [Serializable]
2. public class CriminalGroup : ISerializable, IEquatable<CriminalGroup> 10 {
3. #region Fields
4. /// <summary>

13 /// Отображает последний использованный для группировки ID

1. /// </summary>
2. private static int \_lastId = -1;
3. #endregion
4. #region Constructors
5. /// <summary>

19 /// Основной конструктор

20 /// </summary>

21 /// <param name="name">Название группировки</param>

22 /// <param name="additionalData">Дополнительная информация</param>

23 public CriminalGroup(string name, string additionalData) 24 {

1. Id = ++\_lastId;
2. Name = name;
3. AdditionalData = additionalData; 28 }

29

30 /// <summary>

31 /// Конструктор для десериализации

1. /// </summary>
2. public CriminalGroup(SerializationInfo info, StreamingContext context) 34 {
3. \_lastId = info.GetInt32("static.\_lastId");
4. Id = info.GetInt32("Id");
5. Name = info.GetString("Name");
6. AdditionalData = info.GetString("AdditionalData"); 39 }
7. #endregion
8. #region Properties
9. /// <summary>

43 /// Уникальный номер группировки

1. /// </summary>
2. public int Id { get; set; } 46

47 /// <summary>

48 /// Название группировки

1. /// </summary>
2. public string Name { get; set; } 51

52 /// <summary>

53 /// Дополнительная информация о группировке

1. /// </summary>
2. public string AdditionalData { get; set; } 56

57 /// <summary>

58 /// Коллекция членов группировки. Только для чтения

1. /// </summary>
2. public ReadOnlyDictionary<int, Сriminal> Members => new ReadOnlyDictionary<int, Сriminal>(\_members);
3. private readonly Dictionary<int, Сriminal> \_members = new Dictionary<int, Сriminal>();
4. #endregion
5. #region Methods
6. /// <summary>

65 /// Добавление члена в группировку

66 /// Контролирует корректность поля CriminalGroupMembership аргумента criminal

67 /// </summary>

68 /// <param name="criminal">Преступник, добавляемый в группировку</ param>

69 public void AddMember(Сriminal criminal) 70 {

1. if (\_members.ContainsKey(criminal.Id))
2. \_members.Remove(criminal.Id);
3. \_members.Add(criminal.Id, criminal);
4. criminal.SetCriminalGroup(this); 75 }

76

77 /// <summary>

78 /// Удаление преступника из группировки

79 /// Контролирует корректность поля CriminalGroupMembership аргумента criminal

80 /// </summary>

81 /// <param name="id">Уникальный номер преступника</param>

82 public void RemoveMember(int id)

83 {

1. if(!\_members.ContainsKey(id)) throw new KeyNotFoundException();
2. Сriminal criminal = \_members[id];
3. \_members.Remove(id);
4. criminal.UnsetCriminalGroup(); 88 }

89

90 /// <summary>

91 /// Заполняет SerializationInfo данными, необходимыми для сериализации

1. /// </summary>
2. public void GetObjectData(SerializationInfo info, StreamingContext context)

94 {

95

96

97

98

99 }

100

info.AddValue("static.\_lastId", \_lastId, typeof(int)); info.AddValue("Id", Id, typeof(int)); info.AddValue("Name", Name, typeof(string));

info.AddValue("AdditionalData", AdditionalData, typeof(string));

101 /// <summary>

102 /// Реализация интерфейса IEquatable<CriminalGroup>

1. /// </summary>
2. public bool Equals(CriminalGroup other)

105 {

106 return Id == other.Id;

|  |  |  |  |
| --- | --- | --- | --- |
| 107 |  |  | } |
| 108 |  |  |  |
| 109 |  |  | /// <summary> |
| 110 |  |  | /// Преобразование данных группировки в строку |
| 111 |  |  | /// </summary> |
| 112 |  |  | public override string ToString() |
| 113 |  |  | { |
| 114 |  |  | return Name; |
| 115 |  |  | } |
| 116 |  |  | #endregion |
| 117 |  | } |  |
| 118 | } |  |  |
| 119 |  |  |  |

1. using System;
2. using System.Collections.Generic;
3. using System.Collections.ObjectModel;
4. using System.IO;
5. using System.Linq;
6. using System.Runtime.Serialization.Formatters.Binary; 7

8 namespace InterpolDatabaseProject.Model 9 {

10 /// <summary>

11 /// Класс для управления данными программы

1. /// </summary>
2. [Serializable]
3. public static class Database 15 {
4. #region Properties
5. #region \_public
6. /// <summary>

19 /// Коллекция преступников. Только для чтения

1. /// </summary>
2. public static ReadOnlyDictionary<int, Сriminal> Criminals => new ReadOnlyDictionary<int, Сriminal>(\_criminals);

22

23 /// <summary>

24 /// Коллекция группировок. Только для чтения

1. /// </summary>
2. public static ReadOnlyDictionary<int, CriminalGroup> CriminalGroups => new ReadOnlyDictionary<int, CriminalGroup>(\_criminalGroups);
3. #endregion
4. #region \_private
5. private static Dictionary<int, Сriminal> \_criminals = new Dictionary<int, Сriminal>();
6. private static Dictionary<int, CriminalGroup> \_criminalGroups = new Dictionary<int, CriminalGroup>();
7. #endregion
8. #endregion
9. #region Methods
10. #region Serialization
11. /// <summary>

36 /// Метод для сериализации данных

1. /// </summary>
2. public static void SaveData()

39 {

1. BinaryFormatter binFormat = new BinaryFormatter();
2. using (Stream stream = new FileStream("../../Storage/Data/ criminals.dat", FileMode.Create, FileAccess.Write, FileShare.None))
3. binFormat.Serialize(stream, \_criminals);
4. using (Stream stream = new FileStream("../../Storage/Data/ criminalGroups.dat", FileMode.Create, FileAccess.Write, FileShare.None))
5. binFormat.Serialize(stream, \_criminalGroups); 45

46 }

47 /// <summary>

48 /// Метод для восстановления данных

1. /// </summary>
2. public static void RestoreData()

51 {

1. BinaryFormatter binFormat = new BinaryFormatter();
2. using (Stream stream = new FileStream("../../Storage/Data/ criminalGroups.dat", FileMode.Open, FileAccess.Read, FileShare.None))
3. \_criminalGroups = (Dictionary<int, CriminalGroup>) binFormat.Deserialize(stream);
4. using (Stream stream = new FileStream("../../Storage/Data/ criminals.dat", FileMode.Open, FileAccess.Read, FileShare.None))
5. \_criminals = (Dictionary<int, Сriminal>)binFormat.Deserialize (stream);

57

58 //Cвязывание преступных группировок с преступниками

59 foreach (var criminal in Criminals)

60 {

1. if(criminal.Value.CriminalGroupMembership!=null)
2. criminal.Value.SetCriminalGroup(CriminalGroups [criminal.Value.CriminalGroupMembership.Id]);

63 }

64 }

1. #endregion
2. #region \_criminals Actions
3. /// <summary>

68 /// Добавление преступника в коллекцию.

69 /// Контролирует совпадение ключа в словаре с ID преступника.

70 /// </summary>

71 /// <param name="criminal">Преступник, которого добавляем</param>

72 public static void AddCriminal(Сriminal criminal) 73 {

74 \_criminals.Add(criminal.Id, criminal); 75 }

76

77 /// <summary>

78 /// Удаление преступника из коллекции

79 /// Контролирует также удаление преступника из списка членов группировки

80 /// </summary>

81 /// <param name="id">Уникальный номер преступника</param>

82 public static void DeleteCriminal(int id) 83 {

1. \_criminals[id].UnsetCriminalGroup();
2. \_criminals.Remove(id); 86 }

87

1. #endregion
2. #region \_criminalGroups Actions
3. /// <summary>

91 /// Добавление группировки в коллекцию

92 /// </summary>

93 /// <param name="criminalGroup">Добавляемая группировка</param>

94 public static void AddCriminalGroup(CriminalGroup criminalGroup) =>

\_criminalGroups.Add(criminalGroup.Id, criminalGroup);

95

96 /// <summary>

97 /// Метод для удаления группировки из коллекции

98 /// </summary>

99 /// <param name="id">Уникальный номер группировки</param>

100 public static void DeleteCriminalGroup(int id)

101 {

1. while (CriminalGroups[id].Members.Count > 0)
2. CriminalGroups[id].Members.First().Value.UnsetCriminalGroup();
3. \_criminalGroups.Remove(id);

105 }

1. #endregion
2. #region Other
3. /// <summary>

109 /// Метод для изменения фотографии преступника

1. /// </summary>
2. /// <param name="photoFilePath">Путь к фотографии</param>

112 /// <param name="criminalId">Уникальный номер преступника</param>

113 public static void ChangeCriminalsPhoto(string photoFilePath, int criminalId)

114 {

115

116 }

117

Criminals[criminalId].PhotoFileName = MovePhotoToLibrary (photoFilePath, criminalId);

118 /// <summary>

119 /// Метод для сохраниения фотографии в каталоге программы

1. /// </summary>
2. /// <param name="photoFilePath">Путь к фотографии</param> 122 /// <param name="id">Уникальный номер преступника</param> 123 /// <returns>Конечное имя файла</returns>

124 private static string MovePhotoToLibrary(string photoFilePath, int id)

125 {

1. if (!File.Exists(photoFilePath)) throw new FileNotFoundException ();
2. int count = 1;
3. string fileName = id.ToString();
4. string fileExtension = Path.GetExtension(photoFilePath);
5. string path = "../../Storage/Files/";
6. string newFullPath = path + fileName + fileExtension; 132

133 while (File.Exists(newFullPath))

134 {

1. string tempFileName = string.Format("{0}({1})", fileName, count++);
2. newFullPath = Path.Combine(path, tempFileName + fileExtension);

|  |  |  |  |
| --- | --- | --- | --- |
| 137 |  |  | } |
| 138 |  |  |  |
| 139 |  |  | File.Copy(photoFilePath, newFullPath); |
| 140 |  |  | return Path.GetFileName(newFullPath); |
| 141 |  |  | } |
| 142 |  |  | #endregion |
| 143 |  |  | #endregion |
| 144 |  | } |  |
| 145 | } |  |  |

* 1. <controls:MetroWindow x:Class="InterpolDatabaseProject.CriminalProfile"
  2. xmlns="<http://schemas.microsoft.com/winfx/2006/xaml/presentation>"
  3. xmlns:x="<http://schemas.microsoft.com/winfx/2006/xaml>"
  4. xmlns:d="<http://schemas.microsoft.com/expression/blend/2008>"
  5. xmlns:mc="<http://schemas.openxmlformats.org/markup-compatibility/2006>"
  6. xmlns:local="clr-namespace:InterpolDatabaseProject"
  7. xmlns:controls="<http://metro.mahapps.com/winfx/xaml/controls>"
  8. mc:Ignorable="d"
  9. Title="Criminal profile" MinHeight="600" MinWidth="800" Height="{Binding Path=MinHeight}"
  10. GlowBrush="{DynamicResource AccentColorBrush}" > 11

1. <controls:MetroWindow.Flyouts>
2. <controls:FlyoutsControl>
3. <controls:Flyout Name="ChangeCriminalFlyout" Header="Change criminal" Position="Right" Width="600" IsOpenChanged="ChangeCriminalFlyout\_OnIsOpenChanged">
4. <ScrollViewer>
5. <Grid>
6. <Grid.Resources>
7. <Style TargetType="{x:Type Label}">
8. <Setter Property="Margin" Value="10" />
9. </Style>
10. </Grid.Resources>
11. <Grid.ColumnDefinitions>
12. <ColumnDefinition></ColumnDefinition>
13. <ColumnDefinition></ColumnDefinition>
14. </Grid.ColumnDefinitions>
15. <Grid.RowDefinitions>
16. <RowDefinition></RowDefinition>
17. <RowDefinition></RowDefinition>
18. <RowDefinition></RowDefinition>
19. <RowDefinition></RowDefinition>
20. <RowDefinition></RowDefinition>
21. <RowDefinition></RowDefinition>
22. <RowDefinition></RowDefinition>
23. <RowDefinition></RowDefinition>
24. <RowDefinition></RowDefinition>
25. <RowDefinition></RowDefinition>
26. <RowDefinition></RowDefinition>
27. <RowDefinition></RowDefinition>
28. <RowDefinition></RowDefinition>
29. <RowDefinition></RowDefinition>
30. <RowDefinition></RowDefinition>
31. <RowDefinition></RowDefinition>
32. <RowDefinition></RowDefinition>
33. <RowDefinition></RowDefinition>
34. <RowDefinition></RowDefinition>
35. <RowDefinition></RowDefinition>
36. <RowDefinition></RowDefinition>
37. </Grid.RowDefinitions>
38. <Label Grid.Column="0" Grid.Row="0">Forename</Label>
39. <Label Grid.Column="0" Grid.Row="1">Codename</Label>
40. <Label Grid.Column="0" Grid.Row="2">Lastname</Label>
41. <Label Grid.Column="0" Grid.Row="3">Height</Label>
42. <Label Grid.Column="0" Grid.Row="4">Color of eye</

Label>

1. <Label Grid.Column="0" Grid.Row="5">Color of hair</ Label>
2. <Label Grid.Column="0" Grid.Row="6">Sex</Label>
3. <Label Grid.Column="0" Grid.Row="7">Special Signs</ Label>
4. <Label Grid.Column="0" Grid.Row="8">Citizenship</ Label>
5. <Label Grid.Column="0" Grid.Row="9">Country of birth</ Label>
6. <Label Grid.Column="0" Grid.Row="10">Place of birth</ Label>
7. <Label Grid.Column="0" Grid.Row="11">Date of birth</ Label>
8. <Label Grid.Column="0" Grid.Row="12">Last lived in (Country)</Label>
9. <Label Grid.Column="0" Grid.Row="13">Last lived in (Place)</Label>
10. <Label Grid.Column="0" Grid.Row="14">Languages</Label>
11. <Label Grid.Column="0" Grid.Row="15">Current state</ Label>
12. <Label Grid.Column="0" Grid.Row="16">Photo</Label>
13. <Label Grid.Column="0" Grid.Row="18">Criminal group</ Label>
14. <Label Grid.Column="0" Grid.Row="19">Charges</Label> 68

69

1. <TextBox Name="ChangeCriminalFlyout\_ForenameTextBox" ToolTip="Leave empty if it is unknown" Grid.Column="1" Grid.Row="0" HorizontalAlignment="Stretch" VerticalAlignment="Center"></TextBox>
2. <TextBox Name="ChangeCriminalFlyout\_CodenameTextBox" ToolTip="Leave empty if it is unknown" Grid.Column="1" Grid.Row="1" HorizontalAlignment="Stretch" VerticalAlignment="Center"></TextBox>
3. <TextBox Name="ChangeCriminalFlyout\_LastnameTextBox" ToolTip="Leave empty if it is unknown" Grid.Column="1" Grid.Row="2" HorizontalAlignment="Stretch" VerticalAlignment="Center"></TextBox>
4. <StackPanel Grid.Column="1" Grid.Row="3" Orientation="Horizontal">
5. <CheckBox

Name="ChangeCriminalFlyout\_IsHeightKnownCheckBox" IsChecked="True"/>

1. <Slider IsEnabled="{Binding

ElementName=ChangeCriminalFlyout\_IsHeightKnownCheckBox, Path=IsChecked}" Width="100" Name="ChangeCriminalFlyout\_HeightSlider" HorizontalAlignment="Stretch" VerticalAlignment="Center" Minimum="0" Maximum="250" SmallChange="1" LargeChange="5" Value="100" IsSnapToTickEnabled="True" TickFrequency="1"></Slider>

1. <TextBox IsEnabled="{Binding ElementName=ChangeCriminalFlyout\_IsHeightKnownCheckBox, Path=IsChecked}" HorizontalAlignment="Stretch" VerticalAlignment="Center" IsReadOnly="True"

Text="{Binding ElementName=ChangeCriminalFlyout\_HeightSlider, Path=Value}"/>

1. </StackPanel>
2. <ComboBox Name="ChangeCriminalFlyout\_EyeColorComboBox" HorizontalAlignment="Stretch" VerticalAlignment="Center" Grid.Column="1" SelectedIndex="0" Grid.Row="4"/>
3. <ComboBox

Name="ChangeCriminalFlyout\_HairColorComboBox" HorizontalAlignment="Stretch" VerticalAlignment="Center" Grid.Column="1" SelectedIndex="0" Grid.Row="5"/>

1. <ComboBox Name="ChangeCriminalFlyout\_SexComboBox" HorizontalAlignment="Stretch" VerticalAlignment="Center" Grid.Column="1" Grid.Row="6" SelectedIndex="0"/>
2. <TextBox Name=

"ChangeCriminalFlyout\_SpecialSignsTextBox" ToolTip="Leave empty if it is unknown" HorizontalAlignment="Stretch" VerticalAlignment="Center" Grid.Column="1" Grid.Row="7" TextWrapping="Wrap"></TextBox>

1. <ComboBox

Name="ChangeCriminalFlyout\_CitizenshipComboBox" HorizontalAlignment="Stretch" VerticalAlignment="Center" Grid.Column="1" Grid.Row="8" SelectedIndex="0"/>

1. <ComboBox

Name="ChangeCriminalFlyout\_BirthCountryComboBox" HorizontalAlignment="Stretch" VerticalAlignment="Center" Grid.Column="1" Grid.Row="9" SelectedIndex="0"/>

1. <TextBox Name="ChangeCriminalFlyout\_BirthplaceTextBox" ToolTip="Leave empty if criminal is singleton"

Grid.Column="1" Grid.Row="10" HorizontalAlignment="Stretch" VerticalAlignment="Center"></TextBox>

1. <DatePicker

Name="ChangeCriminalFlyout\_BirthdateDatePicker" HorizontalAlignment="Stretch" VerticalAlignment="Center" Grid.Column="1" Grid.Row="11" ></DatePicker>

1. <ComboBox

Name="ChangeCriminalFlyout\_LastLivingCountryComboBox" HorizontalAlignment="Stretch" VerticalAlignment="Center" Grid.Column="1" Grid.Row="12" SelectedIndex="0"/>

1. <TextBox

Name="ChangeCriminalFlyout\_LastLivingPlaceTextBox" ToolTip="Leave empty if it is unknown" Grid.Column="1" Grid.Row="13" HorizontalAlignment="Stretch" VerticalAlignment="Center"></TextBox>

1. <ScrollViewer Grid.Column="1" Grid.Row="14" MaxHeight="100">
2. <ListBox

Name="ChangeCriminalFlyout\_LanguagesListBox" HorizontalAlignment="Stretch" VerticalAlignment="Center" SelectionMode="Multiple"/>

1. </ScrollViewer>
2. <ComboBox

Name="ChangeCriminalFlyout\_CurrentStateComboBox" HorizontalAlignment="Stretch" VerticalAlignment="Center" Grid.Column="1" Grid.Row="15" SelectedIndex="0"/>

1. <Button Name="ChangeCriminalFlyout\_PickButton" Click="ChangeCriminalFlyout\_PickButton\_OnClick" HorizontalAlignment="Stretch" VerticalAlignment="Center" Grid.Column="1" Grid.Row="16">Pick file</Button>
2. <TextBlock Grid.Column="1" Grid.Row="17" HorizontalAlignment="Stretch" VerticalAlignment="Center" Name="ChangeCriminalFlyout\_PhotoFilePath"></TextBlock>
3. <ComboBox

Name="ChangeCriminalFlyout\_CriminalGroupComboBox" ToolTip="Leave empty if criminal is singleton" HorizontalAlignment="Stretch" VerticalAlignment="Center" Grid.Column="1" Grid.Row="18"/>

1. <ScrollViewer Grid.Column="1" Grid.Row="19" MaxHeight="100">
2. <ListBox

Name="ChangeCriminalFlyout\_ChargesListBox" HorizontalAlignment="Stretch" VerticalAlignment="Center" SelectionMode="Multiple"/>

1. </ScrollViewer>
2. <Button Name="ChangeCriminalFlyout\_SubmitButton" Grid.Column="0" Grid.Row="20" Grid.ColumnSpan="2" Click="ChangeCriminalFlyout\_SubmitButton\_OnClick">SUBMIT</ Button>
3. </Grid>
4. </ScrollViewer>
5. </controls:Flyout>
6. </controls:FlyoutsControl>
7. </controls:MetroWindow.Flyouts> 104
8. <Grid Margin="25" Name="MainGrid">
9. <Grid.ColumnDefinitions>
10. <ColumnDefinition></ColumnDefinition>
11. <ColumnDefinition></ColumnDefinition>
12. <ColumnDefinition></ColumnDefinition>
13. </Grid.ColumnDefinitions>
14. <Grid.RowDefinitions>
15. <RowDefinition></RowDefinition>
16. <RowDefinition></RowDefinition>
17. <RowDefinition Height="20"></RowDefinition>
18. </Grid.RowDefinitions>
19. <Image Grid.Row="0" Name="CriminalImage" Source="Storage/Files/ default.png" Stretch="Uniform"/>
20. <Grid Grid.Row="1" Grid.Column="0">
21. <Grid.ColumnDefinitions>
22. <ColumnDefinition></ColumnDefinition>
23. <ColumnDefinition></ColumnDefinition>
24. </Grid.ColumnDefinitions>
25. <Grid.RowDefinitions>
26. <RowDefinition></RowDefinition>
27. <RowDefinition></RowDefinition>
28. <RowDefinition></RowDefinition>
29. <RowDefinition></RowDefinition>
30. <RowDefinition></RowDefinition>
31. <RowDefinition></RowDefinition>
32. <RowDefinition></RowDefinition>
33. </Grid.RowDefinitions>
34. <Label Grid.Column="0" Grid.Row="0" HorizontalAlignment="Center" VerticalAlignment="Center" FontWeight="Bold">Forename</Label>
35. <Label Grid.Column="0" Grid.Row="1" HorizontalAlignment="Center" VerticalAlignment="Center" FontWeight="Bold">Codename</Label>
36. <Label Grid.Column="0" Grid.Row="2" HorizontalAlignment="Center" VerticalAlignment="Center" FontWeight="Bold">Lastname</Label>
37. <Label Grid.Column="0" Grid.Row="3" HorizontalAlignment="Center" VerticalAlignment="Center" FontWeight="Bold">State</Label>
38. <Label Grid.Column="0" Grid.Row="4" HorizontalAlignment="Center" VerticalAlignment="Center" FontWeight="Bold">Age</Label>
39. <Label Grid.Column="0" Grid.Row="5" HorizontalAlignment="Center" VerticalAlignment="Center" FontWeight="Bold">Height</Label>
40. <Label Grid.Column="0" Grid.Row="6" HorizontalAlignment="Center" VerticalAlignment="Center" FontWeight="Bold">Cirminal Group</ Label>

138

1. <Label Name="ForenameLabel" Grid.Column="1" Grid.Row="0" HorizontalAlignment="Center" VerticalAlignment="Center" FontStyle="Italic">Daniel</Label>
2. <Label Name="CodenameLabel" Grid.Column="1" Grid.Row="1" HorizontalAlignment="Center" VerticalAlignment="Center" FontStyle="Italic">Big D</Label>
3. <Label Name="LastnameLabel" Grid.Column="1" Grid.Row="2" HorizontalAlignment="Center" VerticalAlignment="Center" FontStyle="Italic">Slupskiy</Label>
4. <Label Name="StateLabel" Grid.Column="1" Grid.Row="3" HorizontalAlignment="Center" VerticalAlignment="Center" FontStyle="Italic">Wanted</Label>
5. <Label Name="AgeLabel" Grid.Column="1" Grid.Row="4" HorizontalAlignment="Center" VerticalAlignment="Center" FontStyle="Italic">18</Label>
6. <Label Name="HeightLabel" Grid.Column="1" Grid.Row="5" HorizontalAlignment="Center" VerticalAlignment="Center" FontStyle="Italic">189</Label>
7. <Label Name="CriminalGroupLabel" Grid.Column="1" Grid.Row="6" HorizontalAlignment="Center" VerticalAlignment="Center" FontStyle="Italic">ISIS</Label>

146

147

1. </Grid>
2. <Grid Grid.Row="0" Grid.Column="1">
3. <Grid.ColumnDefinitions>
4. <ColumnDefinition></ColumnDefinition>
5. <ColumnDefinition></ColumnDefinition>
6. </Grid.ColumnDefinitions>
7. <Grid.RowDefinitions>
8. <RowDefinition></RowDefinition>
9. <RowDefinition></RowDefinition>
10. <RowDefinition></RowDefinition>
11. <RowDefinition></RowDefinition>
12. <RowDefinition Height="2\*"></RowDefinition>
13. <RowDefinition></RowDefinition>
14. <RowDefinition Height="2\*"></RowDefinition>
15. </Grid.RowDefinitions>
16. <Label Grid.Column="0" Grid.Row="0" HorizontalAlignment="Center" VerticalAlignment="Center" FontWeight="Bold">Color of eyes</

Label>

1. <Label Grid.Column="0" Grid.Row="1" HorizontalAlignment="Center" VerticalAlignment="Center" FontWeight="Bold">Color of hair</ Label>
2. <Label Grid.Column="0" Grid.Row="2" HorizontalAlignment="Center" VerticalAlignment="Center" FontWeight="Bold">Sex</Label>

166

1. <Label Name="ColorOfEyesLabel" Grid.Column="1" Grid.Row="0" HorizontalAlignment="Center" VerticalAlignment="Center" FontStyle="Italic">Black</Label>
2. <Label Name="ColorOfHairLabel" Grid.Column="1" Grid.Row="1" HorizontalAlignment="Center" VerticalAlignment="Center" FontStyle="Italic">Black</Label>
3. <Label Name="SexLabel" Grid.Column="1" Grid.Row="2" HorizontalAlignment="Center" VerticalAlignment="Center" FontStyle="Italic">Male</Label>

170

1. <Label Grid.Column="0" Grid.ColumnSpan="2" Grid.Row="3" HorizontalAlignment="Center" VerticalAlignment="Center" FontWeight="Bold">Languages</Label>
2. <ScrollViewer Grid.Column="0" Grid.Row="4" Grid.ColumnSpan="2" >
3. <TextBlock Name="LanguagesTextBlock" TextWrapping="Wrap" HorizontalAlignment="Center" VerticalAlignment="Center" FontStyle="Italic">English Russian</TextBlock>
4. </ScrollViewer>
5. <Label Grid.Column="0" Grid.ColumnSpan="2" Grid.Row="5" HorizontalAlignment="Center" VerticalAlignment="Center" FontWeight="Bold">Charges</Label>
6. <ScrollViewer Grid.Column="0" Grid.Row="6" Grid.ColumnSpan="2">
7. <TextBlock Name="ChargesTextBlock" TextWrapping="Wrap" HorizontalAlignment="Center" VerticalAlignment="Center" FontStyle="Italic">Murder Robbery</TextBlock>
8. </ScrollViewer> 179
9. </Grid>
10. <Grid Grid.Row="1" Grid.Column="1">
11. <Grid.ColumnDefinitions>
12. <ColumnDefinition></ColumnDefinition>
13. <ColumnDefinition></ColumnDefinition>
14. </Grid.ColumnDefinitions>
15. <Grid.RowDefinitions>
16. <RowDefinition></RowDefinition>
17. <RowDefinition></RowDefinition>
18. <RowDefinition></RowDefinition>
19. <RowDefinition></RowDefinition>
20. <RowDefinition></RowDefinition>
21. <RowDefinition></RowDefinition>
22. </Grid.RowDefinitions>
23. <Label Grid.Column="0" Grid.Row="0" HorizontalAlignment="Center" VerticalAlignment="Center" FontWeight="Bold">Citizenship</Label>
24. <Label Grid.Column="0" Grid.Row="1" HorizontalAlignment="Center" VerticalAlignment="Center" FontWeight="Bold">Country of birth</ Label>
25. <Label Grid.Column="0" Grid.Row="2" HorizontalAlignment="Center" VerticalAlignment="Center" FontWeight="Bold">Place</Label>
26. <Label Grid.Column="0" Grid.Row="3" HorizontalAlignment="Center"

VerticalAlignment="Center" FontWeight="Bold">Date of birth</ Label>

1. <Label Grid.Column="0" Grid.Row="4" HorizontalAlignment="Center" VerticalAlignment="Center" FontWeight="Bold">Last lived in</ Label>
2. <Label Grid.Column="0" Grid.Row="5" HorizontalAlignment="Center" VerticalAlignment="Center" FontWeight="Bold">Place</Label>

200

1. <Label Name="CitizenshipLabel" Grid.Column="1" Grid.Row="0" HorizontalAlignment="Center" VerticalAlignment="Center" FontStyle="Italic">Ukraine</Label>
2. <Label Name="CountryOfBirthLabel" Grid.Column="1" Grid.Row="1" HorizontalAlignment="Center" VerticalAlignment="Center" FontStyle="Italic">Ukraine</Label>
3. <Label Name="PlaceOfBirthLabel" Grid.Column="1" Grid.Row="2" HorizontalAlignment="Center" VerticalAlignment="Center" FontStyle="Italic">Kharkiv</Label>
4. <Label Name="DateOfBirthLabel" Grid.Column="1" Grid.Row="3" HorizontalAlignment="Center" VerticalAlignment="Center" FontStyle="Italic">27.02.98</Label>
5. <Label Name="LastLivedInLabel" Grid.Column="1" Grid.Row="4" HorizontalAlignment="Center" VerticalAlignment="Center" FontStyle="Italic">Ukraine</Label>
6. <Label Name="LastPlaceOfLiving" Grid.Column="1" Grid.Row="5" HorizontalAlignment="Center" VerticalAlignment="Center" FontStyle="Italic">Kharkiv</Label>
7. </Grid>
8. <ScrollViewer Grid.Row="0" Grid.RowSpan="2" Grid.Column="2">
9. <TextBlock Name="SpecialSignsTextBlock" TextWrapping="Wrap" HorizontalAlignment="Center" FontStyle="Italic">
10. Some special information about this bastard. Blah, blah, blah,blah, blah, blah
11. Sed egestas, nisl et bibendum gravida, elit est condimentum turpis, nec lacinia magna lorem vitae metus. Proin libero risus, aliquam ut erat porta, ullamcorper iaculis magna. Ut ac orci pellentesque diam facilisis congue. Fusce sit amet laoreet est. Cras ac scelerisque lacus. Quisque venenatis ac augue laoreet convallis. Vivamus quis sodales arcu, ac malesuada nisi. In elit sem, venenatis vitae vulputate id, malesuada non tortor. Aenean diam nunc, lacinia non ornare id, porttitor in nisi. Nam condimentum non ligula non varius. Nulla facilisi. Mauris eu varius nisi. Suspendisse sed orci ut odio accumsan porta id rhoncus sem. In in ante sed lorem ultricies bibendum id in mauris. Suspendisse in libero sit amet lorem sollicitudin ullamcorper.Sed egestas, nisl et bibendum gravida, elit est condimentum turpis, nec lacinia magna lorem vitae metus. Proin libero risus, aliquam ut erat porta, ullamcorper iaculis magna. Ut ac orci pellentesque diam facilisis congue. Fusce sit amet laoreet est. Cras ac scelerisque lacus. Quisque venenatis ac augue laoreet convallis. Vivamus quis sodales arcu, ac malesuada nisi. In elit sem, venenatis vitae vulputate id, malesuada non tortor. Aenean diam nunc, lacinia non ornare id, porttitor in nisi. Nam condimentum non ligula non varius. Nulla facilisi. Mauris eu varius nisi. Suspendisse sed orci ut odio accumsan porta id rhoncus sem. In in ante sed lorem

ultricies bibendum id in mauris. Suspendisse in libero sit amet lorem sollicitudin ullamcorper.

1. </TextBlock>
2. </ScrollViewer>
3. <Button Name="ChangeProfileButton" Grid.Row="2" Grid.Column="1" Click="ChangeProfileButton\_OnClick">Change profile</Button>
4. </Grid>
5. </controls:MetroWindow> 217
6. using System;
7. using System.Collections.Generic;
8. using System.IO;
9. using System.Linq;
10. using System.Windows;
11. using System.Windows.Media.Imaging;
12. using InterpolDatabaseProject.Model;
13. using Microsoft.CSharp.RuntimeBinder; 9

10 namespace InterpolDatabaseProject 11 {

12 /// <summary>

13 /// Логика взаимодействия для CriminalProfile.xaml 14 /// Окно отображает полную информацию о преступнике

1. /// </summary>
2. public partial class CriminalProfile 17 {
3. #region Fields
4. /// <summary>

20 /// Отображаемый преступник

1. /// </summary>
2. private readonly Сriminal \_criminal;
3. #endregion
4. #region Constructors
5. public CriminalProfile(Сriminal criminal) 26 {
6. InitializeComponent();
7. \_criminal = criminal;
8. LoadInformation();
9. SetDataSources(); 31 }

32 #endregion 33

34 /// <summary>

35 /// Загружает актуальную информацию в элементы управления окна

1. /// </summary>
2. public void LoadInformation()

38 {

1. if (\_criminal == null) return;
2. const string unknown = "Unknown";
3. const string nodata = "No data"; 42
4. Title = "Criminal #" + \_criminal.Id;
5. if (\_criminal.PhotoFileName != null) CriminalImage.Source = new BitmapImage(new Uri(Path.GetFullPath("..\\..\\Storage\\Files\\"

+ \_criminal.PhotoFileName)));

45

1. ForenameLabel.Content = (\_criminal.Forename=="")? unknown :

\_criminal.Forename;

1. CodenameLabel.Content = (\_criminal.CodeName == "") ? unknown :

\_criminal.CodeName;

1. LastnameLabel.Content = (\_criminal.Lastname == "") ? unknown :

\_criminal.Lastname;

1. StateLabel.Content = \_criminal.State;
2. HeightLabel.Content = (\_criminal.Height==null)? unknown :

\_criminal.Height.ToString();

1. AgeLabel.Content = (\_criminal.Age == null) ? unknown :

\_criminal.Age.ToString();

1. CriminalGroupLabel.Content =

\_criminal.CriminalGroupMembership?.ToString() ?? "Singleton";

1. ColorOfEyesLabel.Content = \_criminal.ColorOfEye;
2. ColorOfHairLabel.Content = \_criminal.ColorOfHair;
3. SexLabel.Content = \_criminal.Sex;
4. LanguagesTextBlock.Text = \_criminal.Languages.Count > 0 ?
5. \_criminal.Languages.Aggregate("", (current, language) => current + language + " ")

58 :

1. nodata;
2. ChargesTextBlock.Text = \_criminal.Charges.Count > 0 ?
3. \_criminal.Charges.Aggregate("", (current, charge) => current + charge + " ")

62

63

64

65

66

67

68

69

70

71 }

72

:

nodata;

CitizenshipLabel.Content = \_criminal.Citizenship; CountryOfBirthLabel.Content = \_criminal.BirthCountry; PlaceOfBirthLabel.Content = (\_criminal.Birthplace=="")?

unknown:\_criminal.Birthplace;

DateOfBirthLabel.Content = \_criminal.Birthdate?.ToShortDateString () ?? unknown;

LastLivedInLabel.Content = \_criminal.LastLivingCountry; LastPlaceOfLiving.Content = (\_criminal.LastLivingPlace == "") ?

unknown : \_criminal.LastLivingPlace; SpecialSignsTextBlock.Text = (\_criminal.SpecialSigns == "") ?

nodata : \_criminal.SpecialSigns;

73 /// <summary>

74 /// Сохраняет внесенные в преступника изменения

1. /// </summary>
2. private void ChangeCriminalFlyout\_SubmitButton\_OnClick(object sender, RoutedEventArgs e)

77 {

78

79

80

81

82

83

84

85

86

87

88

List<Language> languages = ChangeCriminalFlyout\_LanguagesListBox.SelectedItems.Cast<Languag e>().ToList();

List<Crime> charges = ChangeCriminalFlyout\_ChargesListBox.SelectedItems.Cast<Crime> ().ToList();

\_criminal.Lastname = ChangeCriminalFlyout\_LastnameTextBox.Text;

\_criminal.Forename = ChangeCriminalFlyout\_ForenameTextBox.Text;

\_criminal.CodeName = ChangeCriminalFlyout\_CodenameTextBox.Text;

\_criminal.Height = (ChangeCriminalFlyout\_IsHeightKnownCheckBox.IsChecked.Value) ? (int?) ChangeCriminalFlyout\_HeightSlider.Value : null;

\_criminal.ColorOfEye = (EyeColor) ChangeCriminalFlyout\_EyeColorComboBox.SelectedItem;

\_criminal.ColorOfHair = (HairColor) ChangeCriminalFlyout\_HairColorComboBox.SelectedItem;

\_criminal.Sex = (SexOptions) ChangeCriminalFlyout\_SexComboBox.SelectedItem;

\_criminal.SpecialSigns =

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108 }

109

ChangeCriminalFlyout\_SpecialSignsTextBox.Text;

\_criminal.Citizenship = (Country) ChangeCriminalFlyout\_CitizenshipComboBox.SelectedItem;

\_criminal.BirthCountry = (Country) ChangeCriminalFlyout\_BirthCountryComboBox.SelectedItem;

\_criminal.Birthplace = ChangeCriminalFlyout\_BirthplaceTextBox.Text;

\_criminal.Birthdate = ChangeCriminalFlyout\_BirthdateDatePicker.SelectedDate;

\_criminal.LastLivingCountry = (Country) ChangeCriminalFlyout\_LastLivingCountryComboBox.SelectedIndex;

\_criminal.LastLivingPlace = ChangeCriminalFlyout\_LastLivingPlaceTextBox.Text;

\_criminal.Languages = languages;

\_criminal.Charges = charges;

\_criminal.State = (CriminalStateOptions) ChangeCriminalFlyout\_CurrentStateComboBox.SelectedIndex;

\_criminal.UnsetCriminalGroup(); if(ChangeCriminalFlyout\_CriminalGroupComboBox.SelectedIndex != -1)

\_criminal.SetCriminalGroup(Database.CriminalGroups[ ((KeyValuePair<int, CriminalGroup>) ChangeCriminalFlyout\_CriminalGroupComboBox.SelectedItem).K ey]);

if (ChangeCriminalFlyout\_PhotoFilePath.Text != "")

Database.ChangeCriminalsPhoto (ChangeCriminalFlyout\_PhotoFilePath.Text, \_criminal.Id);

LoadInformation(); ChangeCriminalFlyout.IsOpen = false;

110 /// <summary>

111 /// Позволяет выбрать новую фотографию для преступника

1. /// </summary>
2. private void ChangeCriminalFlyout\_PickButton\_OnClick(object sender, RoutedEventArgs e)

114 {

115

Microsoft.Win32.OpenFileDialog dlg = new Microsoft.Win32.OpenFileDialog

116 {

117

118

119

120 };

DefaultExt = ".png", Filter =

"Images (\*.jpeg, \*.png, \*.jpg)|\*.jpeg;\*.png;\*.jpg"

121

122

123

124

125

126 }

127

bool? result = dlg.ShowDialog(); if (result == true)

{

ChangeCriminalFlyout\_PhotoFilePath.Text = dlg.FileName;

}

128 /// <summary>

129 /// Подгружает информацию о преступнике в элементы управления Flyout

1. /// </summary>
2. private void ChangeCriminalFlyout\_OnIsOpenChanged(object sender,

132 {

133

134

135

136 }

137

RoutedEventArgs e)

MainGrid.IsEnabled = !ChangeCriminalFlyout.IsOpen; if (ChangeCriminalFlyout.IsOpen)

LoadChangeCriminalData();

138 /// <summary>

139 /// Подгружает текущую информацию о преступнике в Flyout

1. /// </summary>
2. private void LoadChangeCriminalData()

142 {

1. ChangeCriminalFlyout\_ForenameTextBox.Text = \_criminal.Forename;
2. ChangeCriminalFlyout\_CodenameTextBox.Text = \_criminal.CodeName;
3. ChangeCriminalFlyout\_LastnameTextBox.Text = \_criminal.Lastname;
4. ChangeCriminalFlyout\_HeightSlider.Value = \_criminal.Height ?? 0;
5. ChangeCriminalFlyout\_EyeColorComboBox.SelectedIndex = (int)

\_criminal.ColorOfEye;

1. ChangeCriminalFlyout\_HairColorComboBox.SelectedIndex = (int)

\_criminal.ColorOfHair;

1. ChangeCriminalFlyout\_SexComboBox.SelectedIndex = (int)

\_criminal.Sex;

1. ChangeCriminalFlyout\_CitizenshipComboBox.SelectedIndex = (int)

\_criminal.Citizenship;

1. ChangeCriminalFlyout\_BirthCountryComboBox.SelectedIndex = (int)

\_criminal.BirthCountry;

1. ChangeCriminalFlyout\_BirthplaceTextBox.Text =

\_criminal.Birthplace;

1. ChangeCriminalFlyout\_BirthdateDatePicker.SelectedDate =

\_criminal.Birthdate;

1. ChangeCriminalFlyout\_LastLivingCountryComboBox.SelectedIndex = (int)\_criminal.LastLivingCountry;
2. ChangeCriminalFlyout\_LastLivingPlaceTextBox.Text =

\_criminal.LastLivingPlace;

1. ChangeCriminalFlyout\_CurrentStateComboBox.SelectedIndex = (int)

\_criminal.State;

1. ChangeCriminalFlyout\_SpecialSignsTextBox.Text =

\_criminal.SpecialSigns;

1. foreach (var language in \_criminal.Languages)
2. ChangeCriminalFlyout\_LanguagesListBox.SelectedItems.Add (ChangeCriminalFlyout\_LanguagesListBox.Items[(int) language]);
3. foreach (var charge in \_criminal.Charges)
4. ChangeCriminalFlyout\_ChargesListBox.SelectedItems.Add (ChangeCriminalFlyout\_ChargesListBox.Items[(int)charge]);
5. if (\_criminal.CriminalGroupMembership == null) return;

|  |  |  |  |
| --- | --- | --- | --- |
| 163 | for | | (int i = 0; i < Database.CriminalGroups.Values.Count; i++) |
| 164 | { | |  |
| 165 |  | | if (\_criminal.CriminalGroupMembership.Id != |
|  |  | | Database.CriminalGroups.Values.ElementAt(i).Id) continue; |
| 166 |  | | ChangeCriminalFlyout\_CriminalGroupComboBox.SelectedIndex = i; |
| 167 break; | | | |
| 168 |  | } | |
| 169 | } |  | |
| 170 |  |  | |
| 171 | /// | <summary> | |

172 /// Устанавливает источники данных для элементов управления

1. /// </summary>
2. private void SetDataSources()

175 {

1. ChangeCriminalFlyout\_EyeColorComboBox.ItemsSource =
2. Enum.GetValues(typeof(EyeColor)).Cast<EyeColor>();
3. ChangeCriminalFlyout\_HairColorComboBox.ItemsSource =
4. Enum.GetValues(typeof(HairColor)).Cast<HairColor>();
5. ChangeCriminalFlyout\_SexComboBox.ItemsSource =
6. Enum.GetValues(typeof(SexOptions)).Cast<SexOptions>();
7. ChangeCriminalFlyout\_CitizenshipComboBox.ItemsSource =
8. Enum.GetValues(typeof(Country)).Cast<Country>();
9. ChangeCriminalFlyout\_BirthCountryComboBox.ItemsSource =
10. Enum.GetValues(typeof(Country)).Cast<Country>();
11. ChangeCriminalFlyout\_LastLivingCountryComboBox.ItemsSource =
12. Enum.GetValues(typeof(Country)).Cast<Country>();
13. ChangeCriminalFlyout\_CurrentStateComboBox.ItemsSource =
14. Enum.GetValues(typeof (CriminalStateOptions)).Cast<CriminalStateOptions>();
15. ChangeCriminalFlyout\_LanguagesListBox.ItemsSource =
16. Enum.GetValues(typeof(Language)).Cast<Language>();
17. ChangeCriminalFlyout\_ChargesListBox.ItemsSource =
18. Enum.GetValues(typeof(Crime)).Cast<Crime>();
19. ChangeCriminalFlyout\_CriminalGroupComboBox.ItemsSource = Database.CriminalGroups;

195

196

197

198

199

200

201

202 } 203

}

/// <summary>

/// Открывает Flyout для изменения данных о преступнике

/// </summary>

private void ChangeProfileButton\_OnClick(object sender, RoutedEventArgs e) => ChangeCriminalFlyout.IsOpen = true;

}

...tabaseProject\InterpolDatabaseProject\DeleteDialog.xaml 65

* 1. <controls:MetroWindow x:Class="InterpolDatabaseProject.DeleteDialog"
  2. xmlns="<http://schemas.microsoft.com/winfx/2006/xaml/presentation>"
  3. xmlns:x="<http://schemas.microsoft.com/winfx/2006/xaml>"
  4. xmlns:d="<http://schemas.microsoft.com/expression/blend/2008>"
  5. xmlns:mc="<http://schemas.openxmlformats.org/markup-compatibility/2006>"
  6. xmlns:local="clr-namespace:InterpolDatabaseProject"
  7. xmlns:controls="<http://metro.mahapps.com/winfx/xaml/controls>"
  8. mc:Ignorable="d"
  9. Title="Permission required" Height="100" Width="300"
  10. WindowStartupLocation="CenterScreen"
  11. ResizeMode="NoResize">
  12. <StackPanel>
  13. <TextBlock Text="Ask your boss to get a permission" />
  14. <Button Name="OkButton" Content="I have permission" Click="OkButton\_OnClick" />
  15. <Button Name="CancelButton" Content="Cancel" Click="CancelButton\_OnClick" />

16

1. </StackPanel>
2. </controls:MetroWindow> 19

20

...aseProject\InterpolDatabaseProject\DeleteDialog.xaml.cs 66

1 using System.Windows; 2

3 namespace InterpolDatabaseProject 4 {

5 /// <summary>

6 /// Логика взаимодействия для DeleteDialog.xaml

7 /// Данное окно используется для запроса разрешения на удаление данных

1. /// </summary>
2. public partial class DeleteDialog 10 {
3. #region Properties
4. /// <summary>

13 /// Ответ пользователя

1. /// </summary>
2. public bool Result { get; private set; }
3. #endregion
4. #region Constructors
5. /// <summary>

19 /// Основной конструкторв

1. /// </summary>
2. public DeleteDialog()

22 {

23 InitializeComponent(); 24 }

1. #endregion
2. #region Methods
3. /// <summary>

28 /// Закрывает окно с положительным ответом

1. /// </summary>
2. private void OkButton\_OnClick(object sender, RoutedEventArgs e) 31 {
3. Result = true;
4. Close();

34 }

35 /// <summary>

36 /// Закрывает окно с отрицательным ответом

1. /// </summary>
2. private void CancelButton\_OnClick(object sender, RoutedEventArgs e) 39 {
3. Result = false;
4. Close();

42 }

43 #endregion 44 }

45 }

46

1. <Controls:MetroWindow x:Class="InterpolDatabaseProject.MainWindow"
2. xmlns="<http://schemas.microsoft.com/winfx/2006/xaml/presentation>"
3. xmlns:x="<http://schemas.microsoft.com/winfx/2006/xaml>"
4. xmlns:d="<http://schemas.microsoft.com/expression/blend/2008>"
5. xmlns:mc="<http://schemas.openxmlformats.org/markup-compatibility/2006>"
6. xmlns:Controls="clr- namespace:MahApps.Metro.Controls;assembly=MahApps.Metro"
7. xmlns:interpolDatabaseProject="clr-namespace:InterpolDatabaseProject"
8. mc:Ignorable="d"
9. Title="Interpol database" MinHeight="600" Height="600" MinWidth="800" Width="800"
10. Icon="Storage/Files/icon.png" GlowBrush="{DynamicResource AccentColorBrush}"
11. WindowStartupLocation="CenterScreen" Closing="MainWindow\_OnClosing">
12. <Controls:MetroWindow.Flyouts>
13. <Controls:FlyoutsControl>
14. <Controls:Flyout Name="AddNewCriminalFlyout" Header="New criminal" Position="Right"

15 Width="600"

IsOpenChanged="Flyout\_OnIsOpenChanged">

1. <ScrollViewer>
2. <Grid>
3. <Grid.Resources>
4. <Style TargetType="{x:Type Label}">
5. <Setter Property="Margin" Value="10" />
6. </Style>
7. </Grid.Resources>
8. <Grid.ColumnDefinitions>
9. <ColumnDefinition></ColumnDefinition>
10. <ColumnDefinition></ColumnDefinition>
11. </Grid.ColumnDefinitions>
12. <Grid.RowDefinitions>
13. <RowDefinition></RowDefinition>
14. <RowDefinition></RowDefinition>
15. <RowDefinition></RowDefinition>
16. <RowDefinition></RowDefinition>
17. <RowDefinition></RowDefinition>
18. <RowDefinition></RowDefinition>
19. <RowDefinition></RowDefinition>
20. <RowDefinition></RowDefinition>
21. <RowDefinition></RowDefinition>
22. <RowDefinition></RowDefinition>
23. <RowDefinition></RowDefinition>
24. <RowDefinition></RowDefinition>
25. <RowDefinition></RowDefinition>
26. <RowDefinition></RowDefinition>
27. <RowDefinition></RowDefinition>
28. <RowDefinition></RowDefinition>
29. <RowDefinition></RowDefinition>
30. <RowDefinition></RowDefinition>
31. <RowDefinition></RowDefinition>
32. <RowDefinition></RowDefinition>
33. <RowDefinition></RowDefinition>
34. </Grid.RowDefinitions>
35. <Label Grid.Column="0" Grid.Row="0">Forename</Label>
36. <Label Grid.Column="0" Grid.Row="1">Codename</Label>
37. <Label Grid.Column="0" Grid.Row="2">Lastname</Label>
38. <Label Grid.Column="0" Grid.Row="3">Height</Label>
39. <Label Grid.Column="0" Grid.Row="4">Color of eye</ Label>
40. <Label Grid.Column="0" Grid.Row="5">Color of hair</ Label>
41. <Label Grid.Column="0" Grid.Row="6">Sex</Label>
42. <Label Grid.Column="0" Grid.Row="7">Special Signs</ Label>
43. <Label Grid.Column="0" Grid.Row="8">Citizenship</ Label>
44. <Label Grid.Column="0" Grid.Row="9">Country of birth</ Label>
45. <Label Grid.Column="0" Grid.Row="10">Place of birth</ Label>
46. <Label Grid.Column="0" Grid.Row="11">Date of birth</ Label>
47. <Label Grid.Column="0" Grid.Row="12">Last lived in (Country)</Label>
48. <Label Grid.Column="0" Grid.Row="13">Last lived in (Place)</Label>
49. <Label Grid.Column="0" Grid.Row="14">Languages</Label>
50. <Label Grid.Column="0" Grid.Row="15">Current state</ Label>
51. <Label Grid.Column="0" Grid.Row="16">Photo</Label>
52. <Label Grid.Column="0" Grid.Row="18">Criminal group</ Label>
53. <Label Grid.Column="0" Grid.Row="19">Charges</Label> 69

70

1. <TextBox Name="AddNewCriminal\_ForenameTextBox"
2. ToolTip="Leave empty if it is unknown"
3. Grid.Column="1" Grid.Row="0"

HorizontalAlignment="Stretch"

1. VerticalAlignment="Center">
2. </TextBox>
3. <TextBox Name="AddNewCriminal\_CodenameTextBox"
4. ToolTip="Leave empty if it is unknown"
5. Grid.Column="1" Grid.Row="1"
6. HorizontalAlignment="Stretch"
7. VerticalAlignment="Center">
8. </TextBox>
9. <TextBox Name="AddNewCriminal\_LastnameTextBox"
10. ToolTip="Leave empty if it is unknown"
11. Grid.Column="1" Grid.Row="2"
12. HorizontalAlignment="Stretch"
13. VerticalAlignment="Center">
14. </TextBox>
15. <StackPanel Grid.Column="1" Grid.Row="3" Orientation="Horizontal">
16. <CheckBox

Name="AddNewCriminal\_IsHeightKnownCheckBox" IsChecked="True"/>

1. <Slider IsEnabled="{Binding

ElementName=AddNewCriminal\_IsHeightKnownCheckBox,

1. Path=IsChecked}" Width="100"

Name="AddNewCriminal\_HeightSlider"

1. HorizontalAlignment="Stretch"

VerticalAlignment="Center" Minimum="0"

1. Maximum="250" SmallChange="1"

LargeChange="5" Value="100" IsSnapToTickEnabled="True"

1. TickFrequency="1">
2. </Slider>
3. <TextBox IsEnabled="{Binding ElementName=AddNewCriminal\_IsHeightKnownCheckBox, Path=IsChecked}"
4. HorizontalAlignment="Stretch"

VerticalAlignment="Center" IsReadOnly="True"

1. Text="{Binding

ElementName=AddNewCriminal\_HeightSlider, Path=Value}"/>

1. </StackPanel>
2. <ComboBox Name="AddNewCriminal\_EyeColorComboBox" HorizontalAlignment="Stretch" VerticalAlignment="Center" Grid.Column="1" SelectedIndex="0" Grid.Row="4"/>
3. <ComboBox Name="AddNewCriminal\_HairColorComboBox" HorizontalAlignment="Stretch" VerticalAlignment="Center" Grid.Column="1" SelectedIndex="0" Grid.Row="5"/>
4. <ComboBox Name="AddNewCriminal\_SexComboBox" HorizontalAlignment="Stretch" VerticalAlignment="Center" Grid.Column="1" Grid.Row="6" SelectedIndex="0"/>
5. <TextBox Name="AddNewCriminal\_SpecialSignsTextBox" ToolTip="Leave empty if it is unknown" HorizontalAlignment="Stretch" VerticalAlignment="Center" Grid.Column="1" Grid.Row="7" TextWrapping="Wrap"></ TextBox>
6. <ComboBox Name="AddNewCriminal\_CitizenshipComboBox" HorizontalAlignment="Stretch" VerticalAlignment="Center" Grid.Column="1" Grid.Row="8" SelectedIndex="0"/>
7. <ComboBox Name="AddNewCriminal\_BirthCountryComboBox" HorizontalAlignment="Stretch" VerticalAlignment="Center" Grid.Column="1" Grid.Row="9" SelectedIndex="0"/>
8. <TextBox Name="AddNewCriminal\_BirthplaceTextBox" ToolTip="Leave empty if it is unknown" Grid.Column="1" Grid.Row="10" HorizontalAlignment="Stretch" VerticalAlignment="Center"></TextBox>
9. <DatePicker Name="AddNewCriminal\_BirthdateDatePicker" HorizontalAlignment="Stretch" VerticalAlignment="Center" Grid.Column="1" Grid.Row="11" ></DatePicker>
10. <ComboBox

Name="AddNewCriminal\_LastLivingCountryComboBox" HorizontalAlignment="Stretch" VerticalAlignment="Center" Grid.Column="1" Grid.Row="12" SelectedIndex="0"/>

1. <TextBox Name="AddNewCriminal\_LastLivingPlaceTextBox" ToolTip="Leave empty if it is unknown" Grid.Column="1" Grid.Row="13" HorizontalAlignment="Stretch" VerticalAlignment="Center"></TextBox>
2. <ScrollViewer Grid.Column="1" Grid.Row="14" MaxHeight="100">
3. <ListBox Name="AddNewCriminal\_LanguagesListBox" HorizontalAlignment="Stretch" VerticalAlignment="Center" SelectionMode="Multiple"/>
4. </ScrollViewer>
5. <ComboBox Name="AddNewCriminal\_CurrentStateComboBox" HorizontalAlignment="Stretch" VerticalAlignment="Center" Grid.Column="1" Grid.Row="15" SelectedIndex="0"/>
6. <Button Click="AddNewCriminal\_PickButton\_Click" HorizontalAlignment="Stretch" VerticalAlignment="Center" Grid.Column="1" Grid.Row="16">Pick file</Button>
7. <TextBlock Grid.Column="1" Grid.Row="17" HorizontalAlignment="Stretch" VerticalAlignment="Center" Name="AddNewCriminal\_PhotoFilePath"></TextBlock>
8. <ComboBox Name="AddNewCriminal\_CriminalGroupComboBox" DisplayMemberPath="Value.Name" ToolTip="Leave empty if criminal is singleton" HorizontalAlignment="Stretch" VerticalAlignment="Center" Grid.Column="1" Grid.Row="18"/>
9. <ScrollViewer Grid.Column="1" Grid.Row="19" MaxHeight="100">
10. <ListBox Name="AddNewCriminal\_ChargesListBox" HorizontalAlignment="Stretch" VerticalAlignment="Center" SelectionMode="Multiple"/>
11. </ScrollViewer>
12. <Button Grid.Column="0" Grid.Row="20" Grid.ColumnSpan="2" Click="AddNewCriminal\_SubmitButton\_Click">SUBMIT</Button>
13. </Grid>
14. </ScrollViewer>
15. </Controls:Flyout>
16. <Controls:Flyout Name="AddNewCriminalGroupFlyout" Header="New criminal group" Position="Left" Width="400" IsOpenChanged="Flyout\_OnIsOpenChanged">
17. <StackPanel Width="300">
18. <Label>Name</Label>
19. <TextBox Name="AddNewCriminalGroup\_NameTextBox" HorizontalAlignment="Stretch" VerticalAlignment="Center"></TextBox>
20. <Label>Additional data</Label>
21. <TextBox Name="AddNewCriminalGroup\_AdditionalDataTextBox" HorizontalAlignment="Stretch" VerticalAlignment="Center" TextWrapping="Wrap"></TextBox>
22. <Button Name="AddNewCriminalGroup\_SubmitButton" Click="AddNewCriminalGroup\_SubmitButton\_OnClick">Submit</ Button>
23. </StackPanel>
24. </Controls:Flyout>
25. <Controls:Flyout Name="ChangeCriminalGroupFlyout" Header="Change criminal group" Position="Left" Width="400" IsOpenChanged="Flyout\_OnIsOpenChanged">
26. <ScrollViewer>
27. <StackPanel Width="300">
28. <Label>Name</Label>
29. <TextBox Name="ChangeCriminalGroupFlyout\_NameTextBox" HorizontalAlignment="Stretch" VerticalAlignment="Center"/>
30. <Label>Additional data</Label>
31. <TextBox

Name="ChangeCriminalGroupFlyout\_AdditionalDataTextBox" HorizontalAlignment="Stretch" VerticalAlignment="Center" TextWrapping="Wrap"/>

1. <Button Name="ChangeCriminalGroupFlyout\_SubmitButton" Click="ChangeCriminalGroupFlyout\_SubmitButton\_OnClick">Sub mit</Button>
2. </StackPanel>
3. </ScrollViewer>
4. </Controls:Flyout>
5. </Controls:FlyoutsControl>
6. </Controls:MetroWindow.Flyouts>
7. <Controls:MetroWindow.RightWindowCommands>
8. <Controls:WindowCommands>
9. <Button Name="RightWindowCommands\_SaveButton" Click="RightWindowCommands\_SaveButton\_OnClick" Content="save" />
10. <Button Name="RightWindowCommands\_RestoreButton" Click="RightWindowCommands\_RestoreButton\_OnClick" Content="restore" />
11. <Button Name="RightWindowCommands\_AboutButton" Click="RightWindowCommands\_AboutButton\_OnClick" Content="about" />
12. </Controls:WindowCommands>
13. </Controls:MetroWindow.RightWindowCommands>
14. <Grid Name="MainGrid">
15. <Grid.ColumnDefinitions>
16. <ColumnDefinition Width="3\*"></ColumnDefinition>
17. <ColumnDefinition Width="7\*"></ColumnDefinition>
18. </Grid.ColumnDefinitions>
19. <Grid.RowDefinitions>
20. <RowDefinition Height="9\*"></RowDefinition>
21. <RowDefinition Height="1\*"></RowDefinition>
22. </Grid.RowDefinitions>
23. <ScrollViewer Grid.Row="0" Grid.Column="0" Grid.RowSpan="2">
24. <StackPanel>
25. <Expander Header="Current state">
26. <ListBox Name="Filter\_StateListBox" SelectionChanged="Filter\_ControlValueChanged" SelectionMode="Multiple" MaxHeight="100"/>
27. </Expander>
28. <Expander Header="Information">
29. <StackPanel>
30. <Label Content="Lastname"/>
31. <TextBox Name="Filter\_LastnameTextBox" TextChanged="Filter\_ControlValueChanged" Text=""/>
32. <Label Content="Forename"/>
33. <TextBox Name="Filter\_ForenameTextBox" TextChanged="Filter\_ControlValueChanged" Text=""/>
34. <Label Content="Codename"/>
35. <TextBox Name="Filter\_CodenameTextBox" TextChanged="Filter\_ControlValueChanged" Text=""/>
36. </StackPanel>
37. </Expander>
38. <Expander Header="Characteristics">
39. <StackPanel>
40. <Label Content="Age"/>
41. <StackPanel Orientation="Horizontal">
42. <CheckBox Name="Filter\_IsAgeKnown" IsChecked="True" Unchecked="Filter\_ControlValueChanged" Checked="Filter\_ControlValueChanged"/>
43. <Controls:RangeSlider Name="Filter\_AgeRangeSlider" Width="150" IsEnabled="{Binding

ElementName=Filter\_IsAgeKnown, Path=IsChecked}" RangeSelectionChanged="Filter\_ControlValueChanged" Minimum="0" Maximum="150" LowerValue="0" UpperValue="150" AutoToolTipPlacement="TopLeft" AutoToolTipPrecision="0" MoveWholeRange="True" IsSnapToTickEnabled="True"></ Controls:RangeSlider>

1. </StackPanel>
2. <Label Content="Height"/>
3. <StackPanel Orientation="Horizontal">
4. <CheckBox Name="Filter\_IsHeightKnown" Checked="Filter\_ControlValueChanged" Unchecked="Filter\_ControlValueChanged" IsChecked="True"/>
5. <Controls:RangeSlider

Name="Filter\_HeightRangeSlider" Width="150" IsEnabled="{Binding ElementName=Filter\_IsHeightKnown, Path=IsChecked}" RangeSelectionChanged="Filter\_ControlValueChanged" Minimum="0" Maximum="250" LowerValue="0" UpperValue="250" AutoToolTipPlacement="TopLeft" AutoToolTipPrecision="0" MoveWholeRange="True" IsSnapToTickEnabled="True"></ Controls:RangeSlider>

1. </StackPanel>
2. <Label Content="Color of eye"/>
3. <ComboBox Name="Filter\_EyeColorCombobox" SelectionChanged="Filter\_ControlValueChanged" MaxHeight="100" IsEditable="True"/>
4. <Label Content="Color of hair"/>
5. <ComboBox Name="Filter\_HairColorCombobox" SelectionChanged="Filter\_ControlValueChanged" MaxHeight="100" IsEditable="True"/>
6. <Label Content="Sex"/>
7. <ComboBox Name="Filter\_SexCombobox" SelectionChanged="Filter\_ControlValueChanged" MaxHeight="100" IsEditable="True"/>
8. <Label Content="Special signs"/>
9. <TextBox Name="Filter\_SpecialSignsTextbox" TextChanged="Filter\_ControlValueChanged"/>
10. <Label Content="Languages"/>
11. <ListBox Name="Filter\_LanguagesListbox" SelectionChanged="Filter\_ControlValueChanged" SelectionMode="Multiple" MaxHeight="100"/>
12. </StackPanel>
13. </Expander>
14. <Expander Header="Locations">
15. <StackPanel>
16. <Label Content="Citizenship"/>
17. <ComboBox Name="Filter\_CitizenshipComboBox" SelectionChanged="Filter\_ControlValueChanged" MaxHeight="100" IsEditable="True"/>
18. <Label Content="Country of birth"/>
19. <ComboBox Name="Filter\_BirthCountryComboBox" SelectionChanged="Filter\_ControlValueChanged" MaxHeight="100" IsEditable="True"/>
20. <Label Content="Place of birth"/>
21. <TextBox Name="Filter\_PlaceOfBirthTextBox" TextChanged="Filter\_ControlValueChanged"/>
22. <Label Content="Last living country"/>
23. <ComboBox Name="Filter\_LastLivingCountryComboBox" SelectionChanged="Filter\_ControlValueChanged" MaxHeight="100" IsEditable="True"/>
24. <Label Content="Last living place"/>
25. <TextBox Name="Filter\_LastLivingPlaceTextBox" TextChanged="Filter\_ControlValueChanged"/>
26. </StackPanel>
27. </Expander>
28. <Expander Header="Criminal groups">
29. <StackPanel>
30. <Button Name="Filter\_ShowAllCriminals" Click="Filter\_ShowAllCriminals\_OnClick">Criminals from all groups</Button>
31. <Grid>
32. <Grid.ColumnDefinitions>
33. <ColumnDefinition></ColumnDefinition>
34. <ColumnDefinition></ColumnDefinition>
35. </Grid.ColumnDefinitions>
36. <Button

Name="CriminalGroupsListBox\_AddGroupButton" Click="CriminalGroupsListBox\_AddGroupButton\_OnClick" Grid.Column="0" Style="{StaticResource AccentedSquareButtonStyle}">New group</Button>

1. <Button

Name="CriminalGroupsListBox\_DeleteGroupButton" IsEnabled="False" Click="CriminalGroupsListBox\_DeleteGroupButton\_OnClick" Grid.Column="1" Style="{StaticResource AccentedSquareButtonStyle}">Delete group</Button>

1. </Grid>
2. <ListBox Name="CriminalGroupsListBox" DisplayMemberPath="Value.Name" SelectionChanged="CriminalGroupsListBox\_OnSelectionChanged " MaxHeight="100"/>
3. </StackPanel>
4. </Expander>
5. </StackPanel>
6. </ScrollViewer>
7. <ScrollViewer Grid.Row="0" Grid.Column="1" HorizontalScrollBarVisibility="Hidden">
8. <StackPanel>
9. <Grid Name="CriminalGroupInformationGrid" MaxHeight="200" Visibility="Collapsed">
10. <Grid.RowDefinitions>
11. <RowDefinition></RowDefinition>
12. <RowDefinition></RowDefinition>
13. <RowDefinition></RowDefinition>
14. </Grid.RowDefinitions>
15. <Label Name="CriminalGroupInformationGrid\_Name" Grid.Row="0" HorizontalAlignment="Center" FontSize="21.333" FontWeight="Bold">
16. <Label.Effect>
17. <DropShadowEffect ShadowDepth="0.1"/>
18. </Label.Effect>
19. </Label>
20. <ScrollViewer Grid.Row="1">
21. <TextBlock

Name="CriminalGroupInformationGrid\_Information" MaxWidth="500" TextWrapping="Wrap" HorizontalAlignment="Center"/>

1. </ScrollViewer>
2. <Button

Name="CriminalGroupInformationGrid\_ChangeDataButton" Grid.Row="2" Click="CriminalGroupInformationGrid\_ChangeDataButton\_OnCli ck">Change criminal group data</Button>

1. </Grid>
2. <ListBox x:Name="CriminalsListBox" SelectionChanged="CriminalsListBox\_OnSelectionChanged" BorderThickness="0" ItemsSource="{Binding CriminalsToShow}" Tag="{Binding Id}">
3. <ListBox.ItemTemplate>
4. <DataTemplate>
5. <Grid Height="150">
6. <Grid.ColumnDefinitions>
7. <ColumnDefinition Width="1\*"></

ColumnDefinition>

1. <ColumnDefinition Width="5\*"></

ColumnDefinition>

1. </Grid.ColumnDefinitions>
2. <StackPanel Grid.Column="1">
3. <Label Content="{Binding TotalName}" FontSize="18"/>
4. <Label Content="{Binding Citizenship}" FontSize="16"/>
5. </StackPanel>
6. <Image Grid.Column="0" Width="150" Source="{Binding Image}" Stretch="Uniform"/>
7. <Label Content="{Binding State}" HorizontalAlignment="Center" VerticalAlignment="Bottom" FontFamily="Courier New" FontWeight="Bold" FontSize="20" RenderTransformOrigin="0.5,0.5" Foreground="{Binding TextColor}">
8. <Label.Effect>
9. <DropShadowEffect/>
10. </Label.Effect>
11. </Label>
12. </Grid>
13. </DataTemplate>
14. </ListBox.ItemTemplate>
15. </ListBox>
16. </StackPanel>
17. </ScrollViewer>
18. <StackPanel Name="CriminalActionsStackPanel" Grid.Column="1" Grid.Row="1" Orientation="Horizontal">
19. <Button Name="CriminalActions\_AddCriminalButton" Click="CriminalActions\_AddCriminalButton\_OnClick" Style="{StaticResource AccentedSquareButtonStyle}" Margin="5" VerticalAlignment="Center" HorizontalAlignment="Center"

Width="100">Add criminal</Button>

1. <Button Name="CriminalActions\_OpenProfileButton" Click="CriminalActions\_OpenProfileButton\_OnClick" IsEnabled="False" Style="{StaticResource AccentedSquareButtonStyle}" Margin="5" VerticalAlignment="Center" HorizontalAlignment="Center" Width="100">Open profile</Button>
2. <Button Name="CriminalActions\_DeleteCriminalButton" IsEnabled="False" Click="CriminalActions\_DeleteCriminalButton\_OnClick" Style="{StaticResource AccentedSquareButtonStyle}" Margin="5" VerticalAlignment="Center" HorizontalAlignment="Center" Width="100">Delete criminal</Button>
3. </StackPanel>
4. </Grid>
5. </Controls:MetroWindow> 280
6. using System;
7. using System.Collections;
8. using System.Collections.Generic;
9. using System.Collections.ObjectModel;
10. using System.ComponentModel;
11. using System.IO;
12. using System.Linq;
13. using System.Windows;
14. using System.Windows.Controls;
15. using System.Windows.Media;
16. using System.Windows.Media.Imaging;
17. using MahApps.Metro.Controls.Dialogs;
18. using InterpolDatabaseProject.Model; 14

15 namespace InterpolDatabaseProject 16 {

17 /// <summary>

18 /// Логика взаимодействия для MainWindow.xaml

1. /// </summary>
2. public partial class MainWindow 21 {
3. #region Constructors
4. /// <summary>

24 /// Основной конструктор

1. /// </summary>
2. public MainWindow()

27 {

28 InitializeComponent(); 29

1. Database.RestoreData();
2. InitializeComponent();
3. SetDataSources();
4. Reload\_CriminalsListBox();
5. Reload\_CriminalGroups(); 35 }

36 #endregion 37

1. #region Properties
2. /// <summary>

40 /// Отфильтрованная коллекция преступников

1. /// </summary>
2. public List<Сriminal> FilteredСriminals => ApplyFiltersToCriminals (Database.Criminals);

43

44 /// <summary>

45 /// Коллекция объектов CriminalToShow, являющаяся источником данных при отображении списка преступников

1. /// </summary>
2. public List<CriminalsListboxItemData> CriminalsToShow 48 {

49 get

50 {

1. var result =
2. FilteredСriminals.Select(
3. criminal =>
4. new CriminalsListboxItemData(criminal.Id,

55

56

57

58 }

59 }

criminal.Forename, criminal.CodeName,

criminal.Lastname, criminal.Age, criminal.Citizenship.ToString(),

criminal.PhotoFileName ?? "default.png", criminal.State)).ToList();

return result;

60 #endregion 61

62 #region Methods 63

64 /// <summary>

65 /// Применяет необходимые фильтры к коллекции преступников

66 /// </summary>

67 /// <param name="criminals">Полная коллекция преступников</param> 68 /// <returns>Отфильтрованная коллекция преступников</returns>

69 private List<Сriminal> ApplyFiltersToCriminals(ReadOnlyDictionary<int, Сriminal> criminals)

70 {

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

var result = CriminalGroupsListBox.SelectedItems.Count == 0 ? new List<Сriminal>(criminals.Values) : new List<Сriminal> (((KeyValuePair<int, CriminalGroup>) CriminalGroupsListBox.SelectedItem).Value.Members.Values);

if (Filter\_StateListBox.SelectedItems.Count > 0) result = Filter.FilterByState(result, Filter\_StateListBox.SelectedItems);

if (Filter\_ForenameTextBox.Text != "") result = Filter.FilterByForename(result, Filter\_ForenameTextBox.Text);

if (Filter\_CodenameTextBox.Text != "") result = Filter.FilterByCodename(result, Filter\_CodenameTextBox.Text);

if (Filter\_LastnameTextBox.Text != "") result = Filter.FilterByLastname(result, Filter\_LastnameTextBox.Text);

if (!Filter\_IsAgeKnown.IsChecked.Value) result = Filter.FilterByAge(result);

else if (Filter\_AgeRangeSlider.LowerValue > Filter\_AgeRangeSlider.Minimum || Filter\_AgeRangeSlider.UpperValue < Filter\_AgeRangeSlider.Maximum)

result = Filter.FilterByAge(result, (int) Filter\_AgeRangeSlider.LowerValue, (int) Filter\_AgeRangeSlider.UpperValue);

if (!Filter\_IsHeightKnown.IsChecked.Value) result = Filter.FilterByHeight(result);

else if (Filter\_HeightRangeSlider.LowerValue > Filter\_HeightRangeSlider.Minimum || Filter\_HeightRangeSlider.UpperValue < Filter\_HeightRangeSlider.Maximum)

result = Filter.FilterByHeight(result, (int) Filter\_HeightRangeSlider.LowerValue, (int) Filter\_HeightRangeSlider.UpperValue);

if (Filter\_EyeColorCombobox.SelectedIndex != -1)

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111 }

112

result = Filter.FilterByEyeColor(result, (EyeColor) Filter\_EyeColorCombobox.SelectedItem);

if (Filter\_HairColorCombobox.SelectedIndex != -1)

result = Filter.FilterByHairColor(result, (HairColor) Filter\_HairColorCombobox.SelectedItem);

if (Filter\_SexCombobox.SelectedIndex != -1)

result = Filter.FilterBySex(result, (SexOptions) Filter\_SexCombobox.SelectedItem);

if (Filter\_SpecialSignsTextbox.Text != "") result = Filter.FilterBySpecialSigns(result,

Filter\_SpecialSignsTextbox.Text);

if (Filter\_LanguagesListbox.SelectedItems.Count > 0) result = Filter.FilterByLanguages(result,

Filter\_LanguagesListbox.SelectedItems);

if (Filter\_CitizenshipComboBox.SelectedIndex != -1) result = Filter.FilterByCitizenship(result, (Country)

Filter\_CitizenshipComboBox.SelectedItem);

if (Filter\_BirthCountryComboBox.SelectedIndex != -1) result = Filter.FilterByBirthCountry(result, (Country)

Filter\_BirthCountryComboBox.SelectedItem); if (Filter\_PlaceOfBirthTextBox.Text != "")

result = Filter.FilterByBirthPlace(result, Filter\_PlaceOfBirthTextBox.Text);

if (Filter\_LastLivingCountryComboBox.SelectedIndex != -1) result = Filter.FilterByLastLivingCountry(result, (Country)

Filter\_LastLivingCountryComboBox.SelectedItem); if (Filter\_LastLivingPlaceTextBox.Text != "")

result = Filter.FilterByLastLivingPlace(result, Filter\_LastLivingPlaceTextBox.Text);

return result;

113 /// <summary>

114 /// Устанавливает источники неизменяемый в программе данных для соответствующих элементов управления

1. /// </summary>
2. private void SetDataSources()

117 {

1. AddNewCriminal\_EyeColorComboBox.ItemsSource =
2. Enum.GetValues(typeof(EyeColor)).Cast<EyeColor>();
3. AddNewCriminal\_HairColorComboBox.ItemsSource =
4. Enum.GetValues(typeof(HairColor)).Cast<HairColor>();
5. AddNewCriminal\_SexComboBox.ItemsSource =
6. Enum.GetValues(typeof(SexOptions)).Cast<SexOptions>();
7. AddNewCriminal\_CitizenshipComboBox.ItemsSource =
8. Enum.GetValues(typeof(Country)).Cast<Country>();
9. AddNewCriminal\_BirthCountryComboBox.ItemsSource =
10. Enum.GetValues(typeof(Country)).Cast<Country>();
11. AddNewCriminal\_LastLivingCountryComboBox.ItemsSource =
12. Enum.GetValues(typeof(Country)).Cast<Country>();
13. AddNewCriminal\_CurrentStateComboBox.ItemsSource =
14. Enum.GetValues(typeof (CriminalStateOptions)).Cast<CriminalStateOptions>();
15. AddNewCriminal\_LanguagesListBox.ItemsSource =
16. Enum.GetValues(typeof(Language)).Cast<Language>();

|  |  |  |
| --- | --- | --- |
| 134 |  | AddNewCriminal\_ChargesListBox.ItemsSource = |
| 135 |  | Enum.GetValues(typeof(Crime)).Cast<Crime>(); ; |
| 136 |  |  |
| 137 |  | Filter\_EyeColorCombobox.ItemsSource = |
| 138 |  | Enum.GetValues(typeof(EyeColor)).Cast<EyeColor>(); |
| 139 |  | Filter\_HairColorCombobox.ItemsSource = |
| 140 |  | Enum.GetValues(typeof(HairColor)).Cast<HairColor>(); |
| 141 |  | Filter\_SexCombobox.ItemsSource = |
| 142 |  | Enum.GetValues(typeof(SexOptions)).Cast<SexOptions>(); |
| 143 |  | Filter\_LanguagesListbox.ItemsSource = |
| 144 |  | Enum.GetValues(typeof(Language)).Cast<Language>(); |
| 145 |  | Filter\_CitizenshipComboBox.ItemsSource = |
| 146 |  | Enum.GetValues(typeof(Country)).Cast<Country>(); |
| 147 |  | Filter\_BirthCountryComboBox.ItemsSource = |
| 148 |  | Enum.GetValues(typeof(Country)).Cast<Country>(); |
| 149 |  | Filter\_LastLivingCountryComboBox.ItemsSource = |
| 150 |  | Enum.GetValues(typeof(Country)).Cast<Country>(); |
| 151 |  | Filter\_StateListBox.ItemsSource = |
| 152 |  | Enum.GetValues(typeof |
|  |  | (CriminalStateOptions)).Cast<CriminalStateOptions>(); |
| 153 | } |  |
| 154 |  |  |
| 155 | /// | <summary> |
| 156 | /// | Перезагружает источники данных в элементах управления, |

отображающих преступников

1. /// </summary>
2. private void Reload\_CriminalsListBox()

159 {

1. if (CriminalsListBox != null)
2. CriminalsListBox.ItemsSource = CriminalsToShow;

162 }

163

164 /// <summary>

165 /// Перезагружает источники данных в элементах управления, отображающих группировки

1. /// </summary>
2. private void Reload\_CriminalGroups()

168 {

1. if (AddNewCriminal\_CriminalGroupComboBox == null) return;
2. AddNewCriminal\_CriminalGroupComboBox.ItemsSource = Database.CriminalGroups;
3. CriminalGroupsListBox.ItemsSource = Database.CriminalGroups;

172 }

173

1. /// <summary>
2. /// Обработчик события Click для AddNewCriminal\_PickButton. 176 /// Выбирает файл с фотографией преступника
3. /// </summary>
4. private void AddNewCriminal\_PickButton\_Click(object sender, RoutedEventArgs e)

179 {

180

Microsoft.Win32.OpenFileDialog dlg = new Microsoft.Win32.OpenFileDialog

181 {

182

183

DefaultExt = ".png", Filter =

|  |  |  |
| --- | --- | --- |
| 184 |  | "Images (\*.jpeg, \*.png, \*.jpg)|\*.jpeg;\*.png;\*.jpg" |
| 185 |  | }; |
| 186 |  | bool? result = dlg.ShowDialog(); |
| 187 |  |  |
| 188 |  | if (result == true) |
| 189 |  | { |
| 190 |  | AddNewCriminal\_PhotoFilePath.Text = dlg.FileName; |
| 191 |  | } |
| 192 | } |  |
| 193 |  |  |
| 194 | /// | <summary> |
| 195 | /// | Обработчик события Click для AddNewCriminal\_SubmitButton. |
| 196 | /// | Создает и помещает в коллекцию новый экземпляр класса Criminal на |

основе введенных пользователем данных

1. /// </summary>
2. private void AddNewCriminal\_SubmitButton\_Click(object sender, RoutedEventArgs e)

199 {

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

List<Language> languages = AddNewCriminal\_LanguagesListBox.SelectedItems.Cast<Language> ().ToList();

List<Crime> charges = AddNewCriminal\_ChargesListBox.SelectedItems.Cast<Crime>().ToList ();

Database.AddCriminal(new Сriminal( AddNewCriminal\_LastnameTextBox.Text, AddNewCriminal\_ForenameTextBox.Text, AddNewCriminal\_CodenameTextBox.Text, AddNewCriminal\_IsHeightKnownCheckBox.IsChecked.Value ? (int?)

AddNewCriminal\_HeightSlider.Value : null, (EyeColor)AddNewCriminal\_EyeColorComboBox.SelectedItem, (HairColor)AddNewCriminal\_HairColorComboBox.SelectedItem, (SexOptions)AddNewCriminal\_SexComboBox.SelectedItem, AddNewCriminal\_SpecialSignsTextBox.Text, (Country)AddNewCriminal\_CitizenshipComboBox.SelectedItem, (Country)AddNewCriminal\_BirthCountryComboBox.SelectedItem, AddNewCriminal\_BirthplaceTextBox.Text, AddNewCriminal\_BirthdateDatePicker.SelectedDate, (Country)

AddNewCriminal\_LastLivingCountryComboBox.SelectedItem, AddNewCriminal\_LastLivingPlaceTextBox.Text,

languages, (CriminalStateOptions)

AddNewCriminal\_CurrentStateComboBox.SelectedItem, null,

(AddNewCriminal\_CriminalGroupComboBox.SelectedIndex == -1)

? null

: Database.CriminalGroups[((KeyValuePair<int, CriminalGroup>)

AddNewCriminal\_CriminalGroupComboBox.SelectedItem).Key],

226 charges

227 )

228 );

|  |  |  |
| --- | --- | --- |
| 229 |  | |
| 230 |  | if (AddNewCriminal\_PhotoFilePath.Text != "") |
| 231 |  | Database.ChangeCriminalsPhoto |
|  |  | (AddNewCriminal\_PhotoFilePath.Text, Сriminal.LastId); |
| 232 |  |  |
| 233 |  | Reload\_CriminalsListBox(); |
| 234 |  | AddNewCriminalFlyout.IsOpen = false; |
| 235 | } |  |
| 236 |  |  |
| 237 | /// | <summary> |
| 238 | /// | Обработчик события Closing для MainWindow. |
| 239 | /// | Сохраняет данные |
| 240 | /// | </summary> |
| 241 private void MainWindow\_OnClosing(object sender, CancelEventArgs e) => Database.SaveData(); | | |
| 242 |  |  |
| 243 | /// | <summary> |
| 244 | /// | Обработчик события Click для CriminalActions\_AddCriminalButton. |
| 245 | /// | Вызывает функцию очистки AddCriminalFlyout и отображает |

пользователю поля для ввода данных о преступника

1. /// </summary>
2. private void CriminalActions\_AddCriminalButton\_OnClick(object sender, RoutedEventArgs e)

248 {

249

250

251 }

252

ClearAddCriminalFlyout(); AddNewCriminalFlyout.IsOpen = true;

253 /// <summary>

254 /// Метод очистки полей AddCriminalFlyout от данных предыдущих использований

1. /// </summary>
2. private void ClearAddCriminalFlyout()

257 {

1. AddNewCriminal\_BirthdateDatePicker.SelectedDate = null;
2. AddNewCriminal\_LanguagesListBox.UnselectAll();
3. AddNewCriminal\_BirthCountryComboBox.SelectedIndex = 0;
4. AddNewCriminal\_BirthplaceTextBox.Text = "";
5. AddNewCriminal\_ChargesListBox.UnselectAll();
6. AddNewCriminal\_CitizenshipComboBox.SelectedIndex = 0;
7. AddNewCriminal\_CodenameTextBox.Text = "";
8. AddNewCriminal\_CriminalGroupComboBox.SelectedIndex = -1;
9. AddNewCriminal\_CurrentStateComboBox.SelectedIndex = 0;
10. AddNewCriminal\_EyeColorComboBox.SelectedIndex = 0;
11. AddNewCriminal\_HairColorComboBox.SelectedIndex = 0;
12. AddNewCriminal\_ForenameTextBox.Text = "";
13. AddNewCriminal\_LastnameTextBox.Text = "";
14. AddNewCriminal\_SpecialSignsTextBox.Text = "";
15. AddNewCriminal\_LastLivingCountryComboBox.SelectedIndex = 0;
16. AddNewCriminal\_LastLivingPlaceTextBox.Text = "";
17. AddNewCriminal\_SexComboBox.SelectedIndex = 0;
18. AddNewCriminal\_PhotoFilePath.Text = "";

276 }

277

278 /// <summary>

279 /// Обработчик события IsOpenChanged для всех элементов Flyout

280 /// Блокирует (разблокирует) остальные элементы, во время (после) использования Flyout

1. /// </summary>
2. private void Flyout\_OnIsOpenChanged(object sender, RoutedEventArgs e)

283 => MainGrid.IsEnabled = !MainGrid.IsEnabled; 284

1. /// <summary>
2. /// Обработчик события Click для CriminalActions\_DeleteCriminalButton 287 /// Спрашивает разрешение на удаление преступника и в случае

подтверждения удаляет выбранного преступника

1. /// </summary>
2. private void CriminalActions\_DeleteCriminalButton\_OnClick(object sender, RoutedEventArgs e)

290 {

291

292

293

294 }

295

if (!AskPermissionToDelete()) return; Database.DeleteCriminal(((CriminalsListboxItemData)

CriminalsListBox.SelectedItem).Id); Reload\_CriminalsListBox();

1. /// <summary>
2. /// Обработчик события SelectionChanged для CriminalsListBox 298 /// Блокирует/разблокирует кнопки открытия профиля и удаления

преступника

1. /// </summary>
2. private void CriminalsListBox\_OnSelectionChanged(object sender, SelectionChangedEventArgs e)

301 {

302

303

304 }

305

CriminalActions\_DeleteCriminalButton.IsEnabled = CriminalsListBox.SelectedItems.Count != 0;

CriminalActions\_OpenProfileButton.IsEnabled = CriminalsListBox.SelectedItems.Count != 0;

306 /// <summary>

307 /// Обработчик событий измения данных для всех элементов управления фильтра

308 /// Перезагружает список преступников при изменении значений полей фильтра

1. /// </summary>
2. private void Filter\_ControlValueChanged(object sender, object e) => Reload\_CriminalsListBox();

311

1. /// <summary>
2. /// Обработчик события Click для Filter\_ShowAllCriminals 314 /// Сбрасывает выбор группировки
3. /// </summary>
4. private void Filter\_ShowAllCriminals\_OnClick(object sender, RoutedEventArgs e)

317 => CriminalGroupsListBox.UnselectAll(); 318

1. /// <summary>
2. /// Обработчик события SelectionChanged для CriminalGroupsListBox 321 /// Отображает/скрывает данные группировке и разблокирует/блокирует

кнопку удаления

1. /// </summary>
2. private void CriminalGroupsListBox\_OnSelectionChanged(object sender, SelectionChangedEventArgs e)

324 {

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340 }

341

if (CriminalGroupsListBox.SelectedIndex == -1)

{

CriminalGroupInformationGrid.Visibility = Visibility.Collapsed;

CriminalGroupInformationGrid\_Name.Content = ""; CriminalGroupInformationGrid\_Information.Text = "";

}

else

{

var selectedCriminalGroup = ((KeyValuePair<int, CriminalGroup>)CriminalGroupsListBox.SelectedItem).Value;

CriminalGroupInformationGrid.Visibility = Visibility.Visible; CriminalGroupInformationGrid\_Name.Content =

selectedCriminalGroup.Name; CriminalGroupInformationGrid\_Information.Text =

selectedCriminalGroup.AdditionalData;

}

CriminalGroupsListBox\_DeleteGroupButton.IsEnabled = CriminalGroupsListBox.SelectedIndex != -1;

Reload\_CriminalsListBox();

1. /// <summary>
2. /// Обработчик события Click для CriminalGroupsListBox\_AddGroupButton 344 /// Очищает и открывает Flyout для добавления новой группировки
3. /// </summary>
4. private void CriminalGroupsListBox\_AddGroupButton\_OnClick(object sender, RoutedEventArgs e)

347 {

348

349

350 }

351

ClearAddNewCriminalGroupFlyout(); AddNewCriminalGroupFlyout.IsOpen = true;

352 /// <summary>

353 /// Очищает значения элементов управления внутри AddNewCriminalGroupFlyout

1. /// </summary>
2. private void ClearAddNewCriminalGroupFlyout()

356 {

1. AddNewCriminalGroup\_NameTextBox.Text = "";
2. AddNewCriminalGroup\_AdditionalDataTextBox.Text = "";

359 }

360

1. /// <summary>
2. /// Обработчик события Click для AddNewCriminalGroup\_SubmitButton 363 /// Проверяет заполненные пользователем поля при добавлении

группировки

364 /// В случае успешного заполнения создает новую группировку

1. /// </summary>
2. private void AddNewCriminalGroup\_SubmitButton\_OnClick(object sender, RoutedEventArgs e)

367 {

368

if (AddNewCriminalGroup\_NameTextBox.Text == "")

|  |  |  |
| --- | --- | --- |
| 369 | { |  |
| 370 |  | AddNewCriminalGroup\_NameTextBox.Text = "Required"; |
| 371 |  | return; |
| 372 | } |  |
| 373 |  |  |

374

375

376

377 }

378

Database.AddCriminalGroup(new CriminalGroup (AddNewCriminalGroup\_NameTextBox.Text, AddNewCriminalGroup\_AdditionalDataTextBox.Text));

AddNewCriminalGroupFlyout.IsOpen = false; Reload\_CriminalGroups();

379 /// <summary>

380 /// Обработчик события Click для CriminalGroupsListBox\_DeleteGroupButton

381 /// Удаляет выбранную группировку

1. /// </summary>
2. private void CriminalGroupsListBox\_DeleteGroupButton\_OnClick(object sender, RoutedEventArgs e)

384 {

385

386

387

388 }

389

if (!AskPermissionToDelete()) return; Database.DeleteCriminalGroup(((KeyValuePair<int, CriminalGroup>)

CriminalGroupsListBox.SelectedItem).Key); Reload\_CriminalGroups();

1. /// <summary>
2. /// Обработчик события Click для CriminalActions\_OpenProfileButton 392 /// Открывает профиль выбранного преступника
3. /// </summary>
4. private void CriminalActions\_OpenProfileButton\_OnClick(object sender, RoutedEventArgs e)

395 {

396

397

398

399

400 }

401

CriminalProfile criminalProfile = new CriminalProfile (Database.Criminals[((CriminalsListboxItemData) CriminalsListBox.SelectedItem).Id]);

criminalProfile.ShowDialog(); Reload\_CriminalsListBox();

402 /// <summary>

403 /// Обработчик события Click для CriminalGroupInformationGrid\_ChangeDataButton

404 /// Открывает Flyout для изменения данных о группировке

405 /// </summary>

406 private void CriminalGroupInformationGrid\_ChangeDataButton\_OnClick (object sender, RoutedEventArgs e)

407 {

408

409

410

411

CriminalGroup selectedCriminalGroup = ((KeyValuePair<int, CriminalGroup>)CriminalGroupsListBox.SelectedItem).Value;

ChangeCriminalGroupFlyout\_NameTextBox.Text = selectedCriminalGroup.Name;

ChangeCriminalGroupFlyout\_AdditionalDataTextBox.Text = selectedCriminalGroup.AdditionalData;

ChangeCriminalGroupFlyout.IsOpen = true;

412 }

413

414 /// <summary>

415 /// Обработчик события Click для ChangeCriminalGroupFlyout\_SubmitButton

416 /// Подтверждает внесенные в информацию о группировке изменения

417 /// </summary>

418 private void ChangeCriminalGroupFlyout\_SubmitButton\_OnClick(object sender, RoutedEventArgs e)

419 {

420

421

422

423

424

425

426

427

428

429

430

431

432

433 }

434

if (ChangeCriminalGroupFlyout\_NameTextBox.Text == "")

{

ChangeCriminalGroupFlyout\_NameTextBox.Text = "Required"; return;

}

CriminalGroup selectedCriminalGroup = ((KeyValuePair<int, CriminalGroup>)CriminalGroupsListBox.SelectedItem).Value;

selectedCriminalGroup.Name = ChangeCriminalGroupFlyout\_NameTextBox.Text;

selectedCriminalGroup.AdditionalData = ChangeCriminalGroupFlyout\_AdditionalDataTextBox.Text;

ChangeCriminalGroupFlyout.IsOpen = false; Reload\_CriminalGroups(); CriminalGroupInformationGrid\_Name.Content =

selectedCriminalGroup.Name; CriminalGroupInformationGrid\_Information.Text =

selectedCriminalGroup.AdditionalData;

435 /// <summary>

436 /// Обработчик события Click для RightWindowCommands\_AboutButton 437 /// Отображает информацию о данной программе

438 /// </summary>

439 private void RightWindowCommands\_AboutButton\_OnClick(object sender, RoutedEventArgs e)

440 {

441

442

443

444

445 }

446

this.ShowMessageAsync("About this program",

"Курсовой проект по ООП студента группы ПИ-15-1 ХНУРЭ Слупского Данилы\n\n" +

"Картотека Интерпола.\n\n" + "Данные по каждому зарегистрированному преступнику: фамилия, имя, кличка, рост, цвет волос, цвет глаз, особые приметы, гражданство, место и дата рождения, последнее место жительства, знание языков, преступная профессия, последнее дело и так далее. Преступные и мафиозные группировки(данные о подельщиках). Выборка по любому подмножеству признаков. Перенос «завязавших» в архив; удаление –только после смерти.",

MessageDialogStyle.Affirmative);

447 /// <summary>

448 /// Обработчик события Click для RightWindowCommands\_SaveButton 449 /// Сохраняет текущее состояние базы в файл

450 /// </summary>

451 private void RightWindowCommands\_SaveButton\_OnClick(object sender,

RoutedEventArgs e)

452 => Database.SaveData(); 453

454 /// <summary>

455 /// Обработчик события Click для RightWindowCommands\_RestoreButton 456 /// Восстанавливает последнее состояние базы из файла

457 /// </summary>

458 private void RightWindowCommands\_RestoreButton\_OnClick(object sender, RoutedEventArgs e)

459 {

460

461

462

463 }

464

Database.RestoreData(); Reload\_CriminalsListBox(); Reload\_CriminalGroups();

465

466

467

468

469

470

471

472

473

474

475

476 }

477

/// <summary>

/// Требует от пользователя разрешение на удаление данных

/// </summary>

/// <returns>Имеет ли пользователь разрешение на удаление данных</ returns>

private bool AskPermissionToDelete()

{

DeleteDialog dialog = new DeleteDialog(); dialog.ShowDialog();

return dialog.Result;

}

#endregion

478 /// <summary>

479 /// Класс, используемый при отображении краткой информации о преступнике

480 /// </summary>

481 public class CriminalsListboxItemData

482 {

483 #region Constructors

484 /// <summary>

485 /// Основной конструктор

486 /// </summary>

487 /// <param name="id">Уникальный номер преступника</param> 488 /// <param name="forename">Имя преступника</param>

489 /// <param name="codename">Кличка преступника</param> 490 /// <param name="lastname">Фамилия преступника</param> 491 /// <param name="age">Возраст преступника</param>

492 /// <param name="citizenship">Гражданство преступника</param>

493 /// <param name="imageName">Название файла с фото преступника</param> 494 /// <param name="state">Текущий статус преступника</param>

495 public CriminalsListboxItemData(int id, string forename, string codename, string lastname, int? age, string citizenship, string imageName, CriminalStateOptions state)

496 {

497

498

499

500

501

Id = id;

TotalName = "#" + id + " - " + forename + " \"" + codename + "\" "

+ lastname + " AGE: " + ((age == null) ? "-" : age.ToString()); Citizenship = citizenship;

if (!File.Exists("..\\..\\Storage\\Files\\" + imageName)) throw new FileNotFoundException();

|  |  |  |  |
| --- | --- | --- | --- |
| 502 |  | Image = new BitmapImage(new Uri(Path.GetFullPath("..\\..\\Storage\ | |
|  |  | \Files\\" + imageName))); | |
| 503 |  |  | |
| 504 |  | State = state; | |
| 505 |  | switch (state) | |
| 506 |  | { | |
| 507 |  | case CriminalStateOptions.Busted: | |
| 508 |  | TextColor = Brushes.Yellow; | |
| 509 |  | break; | |
| 510 |  | case CriminalStateOptions.Wasted: | |
| 511 |  | TextColor = Brushes.Red; | |
| 512 |  | break; ; | |
| 513 |  | case CriminalStateOptions.Wanted: | |
| 514 |  | TextColor = Brushes.White; | |
| 515 |  | break; | |
| 516 |  | default: | |
| 517 |  | throw new ArgumentOutOfRangeException(nameof(state), | |
|  |  | state, null); | |
| 518 |  | } | |
| 519 |  | } | |
| 520 |  | #endregion | |
| 521 |  | #region Properties | |
| 522 |  | /// <summary> | |
| 523 |  | /// Уникальный номер преступника | |
| 524 |  | /// </summary> | |
| 525 |  | public int Id { get; set; } | |
| 526 |  |  | |
| 527 |  | /// <summary> | |
| 528 |  | /// Отображаемое полное имя преступника | |
| 529 |  | /// </summary> | |
| 530 |  | public string TotalName { get; } | |
| 531 |  |  | |
| 532 |  | /// <summary> | |
| 533 |  | /// Гражданство преступника | |
| 534 |  | /// </summary> | |
| 535 public string Citizenship { get; set; } 536 | | | |
| 537 | /// | | <summary> |
| 538 | /// | | Изображение преступника |
| 539 | /// | | </summary> |

540 public BitmapImage Image { get; set; } 541

542 /// <summary>

543 /// Цвет надписи статуса преступника

544 /// Wanted - белый, Busted - желтый, Wasted - красный

545 /// </summary>

546 public Brush TextColor { get; set; } 547

|  |  |  |
| --- | --- | --- |
| 548 |  | /// <summary> |
| 549 |  | /// Текущий статус преступника |
| 550 |  | /// </summary> |
| 551 |  | public CriminalStateOptions State { get; set; } |
| 552 |  | #endregion |
| 553 | } |  |
| 554 |  |  |
| 555 | /// | <summary> |

556 /// Статический класс для группировки методов фильтрации данных

557 /// </summary>

558 public static class Filter

559 {

560 /// <summary>

561 /// Фильтрация по статусу преступника

562 /// </summary>

563 /// <param name="result">Результат предыдущих фильтраций</param>

564 /// <param name="selectedItems">Выбранные значения</param> 565 /// <returns>Отфильтрованная коллекция преступников</returns>

566 public static List<Сriminal> FilterByState(List<Сriminal> result, IList selectedItems)

567 {

568

569

570

571

572

573

574

575

576 }

577

for (int i = 0; i < result.Count; i++)

{

if (selectedItems.Cast<CriminalStateOptions>()

.Any(selectedItem => result[i].State == selectedItem)) continue;

result.RemoveAt(i); i--;

}

return result;

578 /// <summary>

579 /// Фильтрация по имени преступника

580 /// </summary>

581 /// <param name="result">Результат предыдущих фильтраций</param> 582 /// <param name="text">Выбранное значение</param>

583 /// <returns>Отфильтрованная коллекция преступников</returns>

584 public static List<Сriminal> FilterByForename(List<Сriminal> result, string text)

585 => result.Where(r => r.Forename.ToLower().Contains(text.ToLower ())).ToList();

586

587 /// <summary>

588 /// Фильтрация по кличке преступника

589 /// </summary>

590 /// <param name="result">Результат предыдущих фильтраций</param> 591 /// <param name="text">Выбранное значение</param>

592 /// <returns>Отфильтрованная коллекция преступников</returns>

593 public static List<Сriminal> FilterByCodename(List<Сriminal> result, string text)

594 => result.Where(r => r.CodeName.ToLower().Contains(text.ToLower ())).ToList();

595

596 /// <summary>

597 /// Фильтрация по фамилии преступника

598 /// </summary>

599 /// <param name="result">Результат предыдущих фильтраций</param> 600 /// <param name="text">Выбранное значение</param>

601 /// <returns>Отфильтрованная коллекция преступников</returns>

602 public static List<Сriminal> FilterByLastname(List<Сriminal> result, string text)

603 => result.Where(r => r.Lastname.ToLower().Contains(text.ToLower ())).ToList();

604

605 /// <summary>

606 /// Фильтрация по диапазону возрастов преступника (если возраст известен)

607 /// </summary>

608 /// <param name="result">Результат предыдущих фильтраций</param> 609 /// <param name="lowerValue">Начальный возраст</param>

610 /// <param name="upperValue">Конечный возраст</param>

611 /// <returns>Отфильтрованная коллекция преступников</returns>

612 public static List<Сriminal> FilterByAge(List<Сriminal> result, int lowerValue, int upperValue)

613 => result.Where(r => r.Age >= lowerValue && r.Age <= upperValue).ToList();

614

615 /// <summary>

616 /// Фильтрация по диапазону возрастов преступника (если возраст неизвестен)

617 /// </summary>

618 /// <param name="result">Результат предыдущих фильтраций</param> 619 /// <returns>Отфильтрованная коллекция преступников</returns> 620 public static List<Сriminal> FilterByAge(List<Сriminal> result) 621 => result.Where(r => r.Age == null).ToList();

622

623 /// <summary>

624 /// Фильтрация по диапазону ростов преступника (если рост известен) 625 /// </summary>

626 /// <param name="result">Результат предыдущих фильтраций</param> 627 /// <param name="lowerValue">Начальный рост</param>

628 /// <param name="upperValue">Конечный рост</param>

629 /// <returns>Отфильтрованная коллекция преступников</returns>

630 public static List<Сriminal> FilterByHeight(List<Сriminal> result, int lowerValue, int upperValue)

631 => result.Where(r => r.Height >= lowerValue && r.Height <= upperValue).ToList();

632

633 /// <summary>

634 /// Фильтрация по диапазону ростов преступника (если рост неизвестен) 635 /// </summary>

636 /// <param name="result">Результат предыдущих фильтраций</param> 637 /// <returns>Отфильтрованная коллекция преступников</returns>

638 public static List<Сriminal> FilterByHeight(List<Сriminal> result) 639 => result.Where(r => r.Height == null).ToList();

640

641 /// <summary>

642 /// Фильтрация по цвету глаз преступника 643 /// </summary>

644 /// <param name="result">Результат предыдущих фильтраций</param> 645 /// <param name="selectedValue">Выбранное значение</param>

646 /// <returns>Отфильтрованная коллекция преступников</returns>

647 public static List<Сriminal> FilterByEyeColor(List<Сriminal> result, EyeColor selectedValue)

|  |  |  |
| --- | --- | --- |
| 648 |  | => result.Where(r => r.ColorOfEye == selectedValue).ToList(); |
| 649 |  |  |
| 650 | /// | <summary> |
| 651 | /// | Фильтрация по цвету волос преступника |
| 652 | /// | </summary> |

|  |  |  |
| --- | --- | --- |
| 653 | /// | <param name="result">Результат предыдущих фильтраций</param> |
| 654 | /// | <param name="selectedValue">Выбранное значение</param> |
| 655 | /// | <returns>Отфильтрованная коллекция преступников</returns> |
| 656 public static List<Сriminal> FilterByHairColor(List<Сriminal> result, HairColor selectedValue) | | |
| 657 |  | => result.Where(r => r.ColorOfHair == selectedValue).ToList(); |
| 658 |  |  |
| 659 | /// | <summary> |
| 660 | /// | Фильтрация по полу преступника |
| 661 | /// | </summary> |
| 662 | /// | <param name="result">Результат предыдущих фильтраций</param> |
| 663 | /// | <param name="selectedValue">Выбранное значение</param> |
| 664 | /// | <returns>Отфильтрованная коллекция преступников</returns> |
| 665 public static List<Сriminal> FilterBySex(List<Сriminal> result, SexOptions selectedValue) | | |
| 666 |  | => result.Where(r => r.Sex == selectedValue).ToList(); |
| 667 |  |  |
| 668 | /// | <summary> |
| 669 | /// | Фильтрация по особым приметам преступника |
| 670 | /// | </summary> |
| 671 | /// | <param name="result">Результат предыдущих фильтраций</param> |
| 672 | /// | <param name="text">Выбранное значение</param> |
| 673 | /// | <returns>Отфильтрованная коллекция преступников</returns> |
| 674 public static List<Сriminal> FilterBySpecialSigns(List<Сriminal> result, string text) | | |
| 675 |  | => result.Where(r => r.SpecialSigns.ToLower().Contains |
|  |  | (text.ToLower())).ToList(); |
| 676 |  |  |
| 677 | /// | <summary> |
| 678 | /// | Фильтрация по языкам, которые знает преступник |
| 679 | /// | </summary> |
| 680 | /// | <param name="result">Результат предыдущих фильтраций</param> |
| 681 | /// | <param name="selectedItems">Выбранные значения</param> |
| 682 | /// | <returns>Отфильтрованная коллекция преступников</returns> |
| 683 public static List<Сriminal> FilterByLanguages(List<Сriminal> result, IList selectedItems) | | |
| 684 | { |  |
| 685 |  | for (int i = 0; i < result.Count; i++) |
| 686 |  | { |
| 687 |  | if (selectedItems.Cast<Language>().All( |
| 688 |  | selectedItem => result[i].Languages.Contains |
|  |  | (selectedItem))) continue; |
| 689 |  | result.RemoveAt(i); |
| 690 |  | i--; |
| 691 |  | } |
| 692 |  | return result; |
| 693 | } |  |
| 694 |  |  |
| 695 | /// | <summary> |
| 696 | /// | Фильтрация по гражданству преступника |
| 697 | /// | </summary> |
| 698 | /// | <param name="result">Результат предыдущих фильтраций</param> |
| 699 | /// | <param name="selectedValue">Выбранное значение</param> |
| 700 | /// | <returns>Отфильтрованная коллекция преступников</returns> |
| 701 public static List<Сriminal> FilterByCitizenship(List<Сriminal> result, Country selectedValue) => | | |

|  |  |  |
| --- | --- | --- |
| 702 |  | result.Where(r => r.Citizenship == selectedValue).ToList(); |
| 703 |  |  |
| 704 | /// | <summary> |
| 705 | /// | Фильтрация по стране рождения преступника |
| 706 | /// | </summary> |
| 707 | /// | <param name="result">Результат предыдущих фильтраций</param> |
| 708 | /// | <param name="selectedValue">Выбранное значение</param> |
| 709 | /// | <returns>Отфильтрованная коллекция преступников</returns> |

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740 }

public static List<Сriminal> FilterByBirthCountry(List<Сriminal> result, Country selectedValue) =>

result.Where(r => r.BirthCountry == selectedValue).ToList();

/// <summary>

/// Фильтрация по месту рождения преступника

/// </summary>

/// <param name="result">Результат предыдущих фильтраций</param>

/// <param name="text">Выбранное значение</param>

/// <returns>Отфильтрованная коллекция преступников</returns>

public static List<Сriminal> FilterByBirthPlace(List<Сriminal> result, string text) =>

result.Where(r => r.Birthplace.ToLower().Contains(text.ToLower ())).ToList();

/// <summary>

/// Фильтрация по последней зарегистрированной стране проживания преступника

/// </summary>

/// <param name="result">Результат предыдущих фильтраций</param>

/// <param name="selectedValue">Выбранное значение</param>

/// <returns>Отфильтрованная коллекция преступников</returns>

public static List<Сriminal> FilterByLastLivingCountry(List<Сriminal> result, Country selectedValue) =>

result.Where(r => r.LastLivingCountry == selectedValue).ToList();

/// <summary>

/// Фильтрация по последнему зарегистрированному месту проживания преступника

/// </summary>

/// <param name="result">Результат предыдущих фильтраций</param>

/// <param name="text">Выбранное значение</param>

/// <returns>Отфильтрованная коллекция преступников</returns> public static List<Сriminal> FilterByLastLivingPlace(List<Сriminal>

result, string text) =>

result.Where(r => r.LastLivingPlace.ToLower().Contains (text.ToLower())).ToList();

}