

```
1 using System;
2 using System.Collections.Generic;
3 using System.Runtime.Serialization;
4
5 namespace InterpolDatabaseProject.Model
6 {
7
8     #region Enums
9     /// <summary>
10    /// Пол
11    /// </summary>
12    public enum SexOptions
13    {
14        Unknown,
15        Male,
16        Female
17    }
18
19    /// <summary>
20    /// Статус преступника
21    /// </summary>
22    public enum CriminalStateOptions
23    {
24        Wanted,
25        Busted,
26        Wasted
27    }
28
29    /// <summary>
30    /// Виды преступлений
31    /// </summary>
32    public enum Crime
33    {
34        Robbery,
35        Murder,
36        Terrorism
37    }
38
39    /// <summary>
40    /// Страна
41    /// </summary>
42    public enum Country
43    {
44        Unknown,
45        USA,
46        Ukraine,
47        Russia,
48        China,
49        Columbia
50    }
51
52    /// <summary>
53    /// Цвет глаз
54    /// </summary>
55    public enum EyeColor
56    {
```

```
57         Unknown,
58         Black,
59         Brown,
60         Red,
61         Gray
62     }
63
64     /// <summary>
65     /// Цвет волос
66     /// </summary>
67     public enum HairColor
68     {
69         Unknown,
70         Black,
71         Brown,
72         Red,
73         Gray,
74         Bold
75     }
76
77     /// <summary>
78     /// Язык
79     /// </summary>
80     public enum Language
81     {
82         English,
83         Russian,
84         Ukrainian,
85         Spanish,
86         French
87     }
88     #endregion
89
90     [Serializable]
91     public class Criminal : ISerializable
92     {
93         #region Fields
94         /// <summary>
95         /// Отображает последний использованный для преступника ID
96         /// </summary>
97         public static int LastId { get; private set; } = -1;
98         #endregion
99         #region Constructors
100         /// <summary>
101         /// Конструктор для десериализации
102         /// </summary>
103         public Criminal(SerializationInfo info, StreamingContext context)
104         {
105             LastId = info.GetInt32("static._lastId");
106             Id = info.GetInt32("Id");
107             Lastname = info.GetString("Lastname");
108             CodeName = info.GetString("CodeName");
109             Forename = info.GetString("Forename");
110             Height = (int?)info.GetValue("Height", typeof(int?));
111             ColorOfEye = (EyeColor)info.GetValue("ColorOfEye", typeof(EyeColor));
```

```

112     ColorOfHair = (HairColor)info.GetValue("ColorOfHair", typeof
        (HairColor));
113     Sex = (SexOptions)info.GetValue("Sex", typeof(SexOptions));
114     SpecialSigns = (string)info.GetValue("SpecialSigns", typeof
        (string));
115     Citizenship = (Country)info.GetValue("Citizenship", typeof
        (Country));
116     BirthCountry = (Country)info.GetValue("BirthCountry", typeof
        (Country));
117     Birthplace = info.GetString("Birthplace");
118     Birthdate = (DateTime?)info.GetValue("Birthdate", typeof
        (DateTime?));
119     LastLivingCountry = (Country)info.GetValue("LastLivingCountry",
        typeof(Country));
120     LastLivingPlace = info.GetString("LastLivingPlace");
121     Languages = (List<Language>)info.GetValue("Languages", typeof
        (List<Language>));
122     State = (CriminalStateOptions)info.GetValue("State", typeof
        (CriminalStateOptions));
123     CriminalGroupMembership = (CriminalGroup)info.GetValue
        ("CriminalGroupMembership", typeof(CriminalGroup));
124     PhotoFileName = info.GetString("PhotoFileName");
125     Charges = (List<Crime>)info.GetValue("Charges", typeof
        (List<Crime>));
126 }
127
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 Criminal(string lastname, string forename, string codeName,
        int? height,

```

```

151      EyeColor colorOfEye, HairColor colorOfHair, SexOptions sex, string ↗
        specialSigns,
152      Country citizenship, Country birthCountry, string birthplace, ↗
        DateTime? birthdate,
153      Country lastLivingCountry, string lastLivingPlace, List<Language> ↗
        languages, CriminalStateOptions state, string photoFileName,
154      CriminalGroup criminalGroup, List<Crime> charges)
155  {
156      Id = ++LastId;
157      Lastname = lastname;
158      Forename = forename;
159      CodeName = codeName;
160      Height = height;
161      ColorOfEye = colorOfEye;
162      ColorOfHair = colorOfHair;
163      Sex = sex;
164      SpecialSigns = specialSigns;
165      Citizenship = citizenship;
166      BirthCountry = birthCountry;
167      Birthplace = birthplace;
168      Birthdate = birthdate;
169      LastLivingCountry = lastLivingCountry;
170      LastLivingPlace = lastLivingPlace;
171      Languages = languages;
172      State = state;
173      if (criminalGroup != null)
174          SetCriminalGroup(criminalGroup);
175      PhotoFileName = photoFileName;
176      Charges = charges;
177  }
178  #endregion
179  #region Properties
180
181      /// <summary>
182      /// Уникальный номер преступника
183      /// </summary>
184      public int Id { get; set; }
185
186      /// <summary>
187      /// Фамилия преступника
188      /// </summary>
189      public string Lastname { get; set; }
190
191      /// <summary>
192      /// Имя преступника
193      /// </summary>
194      public string Forename { get; set; }
195
196      /// <summary>
197      /// Кличка преступника
198      /// </summary>
199      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
208     /// </summary>
209     public EyeColor ColorOfEye { get; set; }
210
211     /// <summary>
212     /// Цвет волос преступника
213     /// </summary>
214     public HairColor ColorOfHair { get; set; }
215
216     /// <summary>
217     /// Пол преступника
218     /// </summary>
219     public SexOptions Sex { get; set; }
220
221     /// <summary>
222     /// Особые признаки, дополнительные данные
223     /// </summary>
224     public string SpecialSigns { get; set; }
225
226     /// <summary>
227     /// Гражданство преступника
228     /// </summary>
229     public Country Citizenship { get; set; }
230
231     /// <summary>
232     /// Страна рождения преступника
233     /// </summary>
234     public Country BirthCountry { get; set; }
235
236     /// <summary>
237     /// Место рождения
238     /// </summary>
239     public string Birthplace { get; set; }
240
241     /// <summary>
242     /// Дата рождения
243     /// </summary>
244     public DateTime? Birthdate { get; set; }
245
246     /// <summary>
247     /// Последняя зарегистрированная страна проживания
248     /// </summary>
249     public Country LastLivingCountry { get; set; }
250
251     /// <summary>
252     /// Последнее зарегистрированное место проживания
253     /// </summary>
254     public string LastLivingPlace { get; set; }
255
256     /// <summary>
257     /// Список языков, на которых говорит преступник
258     /// </summary>
259     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
283     /// </summary>
284     public int? Age
285     {
286         get
287         {
288             if (Birthdate == null) return null;
289             return DateTime.Today.Year - Birthdate.Value.Year;
290         }
291     }
292     #endregion
293     #region Methods
294     /// <summary>
295     /// Метод для установления членства в преступной группировке.
296     /// Необходим для синхронизации коллекции членов группировки
297     /// и свойства CriminalGroupMembership
298     /// </summary>
299     /// <param name="criminalGroup">Группировка, членом которой является
300     преступник</param>
301     public void SetCriminalGroup(CriminalGroup criminalGroup) {
302         if (CriminalGroupMembership != null) CriminalGroupMembership =
303             null;
304         if (!criminalGroup.Members.ContainsKey(Id))
305             criminalGroup.AddMember(this);
306         else CriminalGroupMembership = criminalGroup;
307     }
308
309     /// <summary>
310     /// Метод для удаления членства в преступной группировке.
311     /// Необходим для синхронизации коллекции членов группировки
312     /// и свойства CriminalGroupMembership
313     /// </summary>
314     public void UnsetCriminalGroup()
315     {
```

```
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         info.AddValue("static._lastId", LastId, typeof(int));
326         info.AddValue("Id", Id, typeof(int));
327
328         info.AddValue("Lastname", Lastname, typeof(string));
329         info.AddValue("CodeName", CodeName, typeof(string));
330         info.AddValue("Forename", Forename, typeof(string));
331         info.AddValue("Height", Height, typeof(int?));
332         info.AddValue("ColorOfEye", ColorOfEye, typeof(EyeColor));
333         info.AddValue("ColorOfHair", ColorOfHair, typeof(HairColor));
334         info.AddValue("Sex", Sex, typeof(SexOptions));
335         info.AddValue("SpecialSigns", SpecialSigns, typeof(List<string>));
336         info.AddValue("Citizenship", Citizenship, typeof(Country));
337         info.AddValue("BirthCountry", BirthCountry, typeof(Country));
338         info.AddValue("Birthplace", Birthplace, typeof(string));
339         info.AddValue("Birthdate", Birthdate, typeof(DateTime?));
340         info.AddValue("LastLivingCountry", LastLivingCountry, typeof(Country));
341         info.AddValue("LastLivingPlace", LastLivingPlace, typeof(string));
342         info.AddValue("Languages", Languages, typeof(List<Language>));
343         info.AddValue("State", State, typeof(CriminalStateOptions));
344         info.AddValue("CriminalGroupMembership", CriminalGroupMembership,
345             typeof(CriminalGroup));
346         info.AddValue("PhotoFileName", PhotoFileName, typeof(string));
347         info.AddValue("Charges", Charges, typeof(List<Crime>));
348     }
349     #endregion
350 }
351
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Collections.ObjectModel;
4 using System.Runtime.Serialization;
5
6 namespace InterpolDatabaseProject.Model
7 {
8     [Serializable]
9     public class CriminalGroup : ISerializable, IEquatable<CriminalGroup>
10     {
11         #region Fields
12         /// <summary>
13         /// Отображает последний использованный для группировки ID
14         /// </summary>
15         private static int _lastId = -1;
16         #endregion
17         #region Constructors
18         /// <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         {
25             Id = ++_lastId;
26             Name = name;
27             AdditionalData = additionalData;
28         }
29
30         /// <summary>
31         /// Конструктор для десериализации
32         /// </summary>
33         public CriminalGroup(SerializationInfo info, StreamingContext context)
34         {
35             _lastId = info.GetInt32("static._lastId");
36             Id = info.GetInt32("Id");
37             Name = info.GetString("Name");
38             AdditionalData = info.GetString("AdditionalData");
39         }
40         #endregion
41         #region Properties
42         /// <summary>
43         /// Уникальный номер группировки
44         /// </summary>
45         public int Id { get; set; }
46
47         /// <summary>
48         /// Название группировки
49         /// </summary>
50         public string Name { get; set; }
51
52         /// <summary>
53         /// Дополнительная информация о группировке
54         /// </summary>
55         public string AdditionalData { get; set; }
56     }
```



```
57     /// <summary>
58     /// Коллекция членов группировки. Только для чтения
59     /// </summary>
60     public ReadOnlyDictionary<int, Criminal> Members => new      ↗
61         ReadOnlyDictionary<int, Criminal>(_members);
62     private readonly Dictionary<int, Criminal> _members = new      ↗
63         Dictionary<int, Criminal>();
64     #endregion
65     #region Methods
66     /// <summary>
67     /// Добавление члена в группировку
68     /// Контролирует корректность поля CriminalGroupMembership аргумента  ↗
69     /// criminal
70     /// </summary>
71     /// <param name="criminal">Преступник, добавляемый в группировку</  ↗
72     param>
73     public void AddMember(Criminal criminal)
74     {
75         if (_members.ContainsKey(criminal.Id))
76             _members.Remove(criminal.Id);
77         _members.Add(criminal.Id, criminal);
78         criminal.SetCriminalGroup(this);
79     }
80     /// <summary>
81     /// Удаление преступника из группировки
82     /// Контролирует корректность поля CriminalGroupMembership аргумента  ↗
83     /// criminal
84     /// </summary>
85     /// <param name="id">Уникальный номер преступника</param>
86     public void RemoveMember(int id)
87     {
88         if(!_members.ContainsKey(id)) throw new KeyNotFoundException();
89         Criminal criminal = _members[id];
90         _members.Remove(id);
91         criminal.UnsetCriminalGroup();
92     }
93     /// <summary>
94     /// Заполняет SerializationInfo данными, необходимыми для сериализации
95     /// </summary>
96     public void GetObjectData(SerializationInfo info, StreamingContext  ↗
97         context)
98     {
99         info.AddValue("static._lastId", _lastId, typeof(int));
100         info.AddValue("Id", Id, typeof(int));
101         info.AddValue("Name", Name, typeof(string));
102         info.AddValue("AdditionalData", AdditionalData, typeof(string));
103     }
104     /// <summary>
105     /// Реализация интерфейса IEquatable<CriminalGroup>
106     /// </summary>
107     public bool Equals(CriminalGroup other)
108     {
109         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
12     /// </summary>
13     [Serializable]
14     public static class Database
15     {
16         #region Properties
17         #region _public
18         /// <summary>
19         /// Коллекция преступников. Только для чтения
20         /// </summary>
21         public static ReadOnlyDictionary<int, Criminal> Criminals => new      ↗
22             ReadOnlyDictionary<int, Criminal>(_criminals);
23
24         /// <summary>
25         /// Коллекция группировок. Только для чтения
26         /// </summary>
27         public static ReadOnlyDictionary<int, CriminalGroup> CriminalGroups => ↗
28             new ReadOnlyDictionary<int, CriminalGroup>(_criminalGroups);
29
30         #endregion
31         #region _private
32         private static Dictionary<int, Criminal> _criminals = new          ↗
33             Dictionary<int, Criminal>();
34         private static Dictionary<int, CriminalGroup> _criminalGroups = new ↗
35             Dictionary<int, CriminalGroup>();
36         #endregion
37         #region Methods
38         #region Serialization
39         /// <summary>
40         /// Метод для сериализации данных
41         /// </summary>
42         public static void SaveData()
43         {
44             BinaryFormatter binFormat = new BinaryFormatter();
45             using (Stream stream = new FileStream("../Storage/Data/
46                 criminals.dat", FileMode.Create, FileAccess.Write,
47                 FileShare.None))
48                 binFormat.Serialize(stream, _criminals);
49             using (Stream stream = new FileStream("../Storage/Data/
50                 criminalGroups.dat", FileMode.Create, FileAccess.Write,
51                 FileShare.None))
52                 binFormat.Serialize(stream, _criminalGroups);
53         }
54         /// <summary>
55         /// Метод для восстановления данных
```

```

49     /// </summary>
50     public static void RestoreData()
51     {
52         BinaryFormatter binFormat = new BinaryFormatter();
53         using (Stream stream = new FileStream("../Storage/Data/
criminalGroups.dat", FileMode.Open, FileAccess.Read,
FileShare.None))
54             _criminalGroups = (Dictionary<int, CriminalGroup>)
binFormat.Deserialize(stream);
55         using (Stream stream = new FileStream("../Storage/Data/
criminals.dat", FileMode.Open, FileAccess.Read, FileShare.None))
56             _criminals = (Dictionary<int, Criminal>)binFormat.Deserialize
(stream);
57
58         //Связывание преступных группировок с преступниками
59         foreach (var criminal in Criminals)
60         {
61             if(criminal.Value.CriminalGroupMembership!=null)
62                 criminal.Value.SetCriminalGroup(CriminalGroups
[criminal.Value.CriminalGroupMembership.Id]);
63         }
64     }
65     #endregion
66     #region _criminals Actions
67     /// <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(Criminal 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     {
84         _criminals[id].UnsetCriminalGroup();
85         _criminals.Remove(id);
86     }
87
88     #endregion
89     #region _criminalGroups Actions
90     /// <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>

```

```

107      /// Метод для удаления группировки из коллекции
108      /// </summary>
109      /// <param name="id">Уникальный номер группировки</param>
110      public static void DeleteCriminalGroup(int id)
111      {
112          while (CriminalGroups[id].Members.Count > 0)
113              CriminalGroups[id].Members.First().Value.UnsetCriminalGroup();
114          _criminalGroups.Remove(id);
115      }
116      #endregion
117      #region Other
118      /// <summary>
119      /// Метод для изменения фотографии преступника
120      /// </summary>
121      /// <param name="photoFilePath">Путь к фотографии</param>
122      /// <param name="criminalId">Уникальный номер преступника</param>
123      public static void ChangeCriminalsPhoto(string photoFilePath, int criminalId)
124      {
125          Criminals[criminalId].PhotoFileName = MovePhotoToLibrary(
126              photoFilePath, criminalId);
127      }
128      /// <summary>
129      /// Метод для сохранения фотографии в каталоге программы
130      /// </summary>
131      /// <param name="photoFilePath">Путь к фотографии</param>
132      /// <param name="id">Уникальный номер преступника</param>
133      /// <returns>Конечное имя файла</returns>
134      private static string MovePhotoToLibrary(string photoFilePath, int id)
135      {
136          if (!File.Exists(photoFilePath)) throw new FileNotFoundException();
137          int count = 1;
138          string fileName = id.ToString();
139          string fileExtension = Path.GetExtension(photoFilePath);
140          string path = "../Storage/Files/";
141          string newFullPath = path + fileName + fileExtension;
142          while (File.Exists(newFullPath))
143          {
144              string tempFileName = string.Format("{0}({1})", fileName, count++);
145              newFullPath = Path.Combine(path, tempFileName + fileExtension);
146          }
147          File.Copy(photoFilePath, newFullPath);
148          return Path.GetFileName(newFullPath);
149      }
150      #endregion
151      #endregion
152      }
153  }

```

```

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}"
      GlowBrush="{DynamicResource AccentColorBrush}" >
10
11
12  <controls:MetroWindow.Flyouts>
13      <controls:FlyoutsControl>
14          <controls:Flyout Name="ChangeCriminalFlyout" Header="Change
      criminal" Position="Right" Width="600"
      IsOpenChanged="ChangeCriminalFlyout_OnIsOpenChanged">
15              <ScrollViewer>
16                  <Grid>
17                      <Grid.Resources>
18                          <Style TargetType="{x:Type Label}">
19                              <Setter Property="Margin" Value="10" />
20                          </Style>
21                      </Grid.Resources>
22                      <Grid.ColumnDefinitions>
23                          <ColumnDefinition/></ColumnDefinition>
24                          <ColumnDefinition/></ColumnDefinition>
25                      </Grid.ColumnDefinitions>
26                      <Grid.RowDefinitions>
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                          <RowDefinition/></RowDefinition>
38                          <RowDefinition/></RowDefinition>
39                          <RowDefinition/></RowDefinition>
40                          <RowDefinition/></RowDefinition>
41                          <RowDefinition/></RowDefinition>
42                          <RowDefinition/></RowDefinition>
43                          <RowDefinition/></RowDefinition>
44                          <RowDefinition/></RowDefinition>
45                          <RowDefinition/></RowDefinition>
46                          <RowDefinition/></RowDefinition>
47                          <RowDefinition/></RowDefinition>
48                      </Grid.RowDefinitions>
49                      <Label Grid.Column="0" Grid.Row="0">Forename</Label>
50                      <Label Grid.Column="0" Grid.Row="1">Codename</Label>
51                      <Label Grid.Column="0" Grid.Row="2">Lastname</Label>
52                      <Label Grid.Column="0" Grid.Row="3">Height</Label>
53                      <Label Grid.Column="0" Grid.Row="4">Color of eye</

```

```

Label>
54     <Label Grid.Column="0" Grid.Row="5">Color of hair</Label>
Label>
55     <Label Grid.Column="0" Grid.Row="6">Sex</Label>
56     <Label Grid.Column="0" Grid.Row="7">Special Signs</Label>
Label>
57     <Label Grid.Column="0" Grid.Row="8">Citizenship</Label>
Label>
58     <Label Grid.Column="0" Grid.Row="9">Country of birth</Label>
Label>
59     <Label Grid.Column="0" Grid.Row="10">Place of birth</Label>
Label>
60     <Label Grid.Column="0" Grid.Row="11">Date of birth</Label>
Label>
61     <Label Grid.Column="0" Grid.Row="12">Last lived in
(Country)</Label>
62     <Label Grid.Column="0" Grid.Row="13">Last lived in
(Place)</Label>
63     <Label Grid.Column="0" Grid.Row="14">Languages</Label>
64     <Label Grid.Column="0" Grid.Row="15">Current state</Label>
Label>
65     <Label Grid.Column="0" Grid.Row="16">Photo</Label>
66     <Label Grid.Column="0" Grid.Row="18">Criminal group</Label>
Label>
67     <Label Grid.Column="0" Grid.Row="19">Charges</Label>
68
69
70     <TextBox Name="ChangeCriminalFlyout_ForenameTextBox"
ToolTip="Leave empty if it is unknown" Grid.Column="1"
Grid.Row="0" HorizontalAlignment="Stretch"
VerticalAlignment="Center"></TextBox>
71     <TextBox Name="ChangeCriminalFlyout_CodenameTextBox"
ToolTip="Leave empty if it is unknown" Grid.Column="1"
Grid.Row="1" HorizontalAlignment="Stretch"
VerticalAlignment="Center"></TextBox>
72     <TextBox Name="ChangeCriminalFlyout_LastnameTextBox"
ToolTip="Leave empty if it is unknown" Grid.Column="1"
Grid.Row="2" HorizontalAlignment="Stretch"
VerticalAlignment="Center"></TextBox>
73     <StackPanel Grid.Column="1" Grid.Row="3"
Orientation="Horizontal">
74         <CheckBox
Name="ChangeCriminalFlyout_IsHeightKnownCheckBox"
IsChecked="True"/>
75         <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>
76         <TextBox IsEnabled="{Binding
ElementName=ChangeCriminalFlyout_IsHeightKnownCheckBox,
Path=IsChecked}" HorizontalAlignment="Stretch"
VerticalAlignment="Center" IsReadOnly="True"

```

```

Text="{Binding
ElementName=ChangeCriminalFlyout_HeightSlider,
Path=Value}"/>
77     </StackPanel>
78     <ComboBox Name="ChangeCriminalFlyout_EyeColorComboBox"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
Grid.Column="1" SelectedIndex="0" Grid.Row="4"/>
79     <ComboBox
Name="ChangeCriminalFlyout_HairColorComboBox"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
Grid.Column="1" SelectedIndex="0" Grid.Row="5"/>
80     <ComboBox Name="ChangeCriminalFlyout_SexComboBox"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
Grid.Column="1" Grid.Row="6" SelectedIndex="0"/>
81     <TextBox Name=
"ChangeCriminalFlyout_SpecialSignsTextBox" ToolTip="Leave
empty if it is unknown" HorizontalAlignment="Stretch"
VerticalAlignment="Center" Grid.Column="1" Grid.Row="7"
TextWrapping="Wrap"></TextBox>
82     <ComboBox
Name="ChangeCriminalFlyout_CitizenshipComboBox"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
Grid.Column="1" Grid.Row="8" SelectedIndex="0"/>
83     <ComboBox
Name="ChangeCriminalFlyout_BirthCountryComboBox"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
Grid.Column="1" Grid.Row="9" SelectedIndex="0"/>
84     <TextBox Name="ChangeCriminalFlyout_BirthplaceTextBox"
ToolTip="Leave empty if criminal is singleton"
Grid.Column="1" Grid.Row="10"
HorizontalAlignment="Stretch"
VerticalAlignment="Center"></TextBox>
85     <DatePicker
Name="ChangeCriminalFlyout_BirthdateDatePicker"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
Grid.Column="1" Grid.Row="11" ></DatePicker>
86     <ComboBox
Name="ChangeCriminalFlyout_LastLivingCountryComboBox"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
Grid.Column="1" Grid.Row="12" SelectedIndex="0"/>
87     <TextBox
Name="ChangeCriminalFlyout_LastLivingPlaceTextBox"
ToolTip="Leave empty if it is unknown" Grid.Column="1"
Grid.Row="13" HorizontalAlignment="Stretch"
VerticalAlignment="Center"></TextBox>
88     <ScrollView Grid.Column="1" Grid.Row="14"
MaxHeight="100">
89     <ListBox
Name="ChangeCriminalFlyout_LanguagesListBox"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
SelectionMode="Multiple"/>
90     </ScrollView>
91     <ComboBox
Name="ChangeCriminalFlyout_CurrentStateComboBox"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
Grid.Column="1" Grid.Row="15" SelectedIndex="0"/>

```



```

92         <Button Name="ChangeCriminalFlyout_PickButton"
Click="ChangeCriminalFlyout_PickButton_OnClick"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
Grid.Column="1" Grid.Row="16">Pick file</Button>
93         <TextBlock Grid.Column="1" Grid.Row="17"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
Name="ChangeCriminalFlyout_PhotoFilePath"></TextBlock>
94         <ComboBox
Name="ChangeCriminalFlyout_CriminalGroupComboBox"
ToolTip="Leave empty if criminal is singleton"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
Grid.Column="1" Grid.Row="18"/>
95         <ScrollViewer Grid.Column="1" Grid.Row="19"
MaxHeight="100">
96             <ListBox
Name="ChangeCriminalFlyout_ChargesListBox"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
SelectionMode="Multiple"/>
97         </ScrollViewer>
98         <Button Name="ChangeCriminalFlyout_SubmitButton"
Grid.Column="0" Grid.Row="20" Grid.ColumnSpan="2"
Click="ChangeCriminalFlyout_SubmitButton_OnClick">SUBMIT</
Button>
99         </Grid>
100     </ScrollViewer>
101 </controls:Flyout>
102 </controls:FlyoutsControl>
103 </controls:MetroWindow.Flyouts>
104
105 <Grid Margin="25" Name="MainGrid">
106     <Grid.ColumnDefinitions>
107         <ColumnDefinition></ColumnDefinition>
108         <ColumnDefinition></ColumnDefinition>
109         <ColumnDefinition></ColumnDefinition>
110     </Grid.ColumnDefinitions>
111     <Grid.RowDefinitions>
112         <RowDefinition></RowDefinition>
113         <RowDefinition></RowDefinition>
114         <RowDefinition Height="20"></RowDefinition>
115     </Grid.RowDefinitions>
116     <Image Grid.Row="0" Name="CriminalImage" Source="Storage/Files/
default.png" Stretch="Uniform"/>
117     <Grid Grid.Row="1" Grid.Column="0">
118         <Grid.ColumnDefinitions>
119             <ColumnDefinition></ColumnDefinition>
120             <ColumnDefinition></ColumnDefinition>
121         </Grid.ColumnDefinitions>
122         <Grid.RowDefinitions>
123             <RowDefinition></RowDefinition>
124             <RowDefinition></RowDefinition>
125             <RowDefinition></RowDefinition>
126             <RowDefinition></RowDefinition>
127             <RowDefinition></RowDefinition>
128             <RowDefinition></RowDefinition>
129             <RowDefinition></RowDefinition>
130     </Grid.RowDefinitions>

```

```

131      <Label Grid.Column="0" Grid.Row="0" HorizontalAlignment="Center"
      VerticalAlignment="Center" FontWeight="Bold">Forename</Label>
132      <Label Grid.Column="0" Grid.Row="1" HorizontalAlignment="Center"
      VerticalAlignment="Center" FontWeight="Bold">Codename</Label>
133      <Label Grid.Column="0" Grid.Row="2" HorizontalAlignment="Center"
      VerticalAlignment="Center" FontWeight="Bold">Lastname</Label>
134      <Label Grid.Column="0" Grid.Row="3" HorizontalAlignment="Center"
      VerticalAlignment="Center" FontWeight="Bold">State</Label>
135      <Label Grid.Column="0" Grid.Row="4" HorizontalAlignment="Center"
      VerticalAlignment="Center" FontWeight="Bold">Age</Label>
136      <Label Grid.Column="0" Grid.Row="5" HorizontalAlignment="Center"
      VerticalAlignment="Center" FontWeight="Bold">Height</Label>
137      <Label Grid.Column="0" Grid.Row="6" HorizontalAlignment="Center"
      VerticalAlignment="Center" FontWeight="Bold">Criminal Group</
      Label>
138
139      <Label Name="ForenameLabel" Grid.Column="1" Grid.Row="0"
      HorizontalAlignment="Center" VerticalAlignment="Center"
      FontStyle="Italic">Daniel</Label>
140      <Label Name="CodenameLabel" Grid.Column="1" Grid.Row="1"
      HorizontalAlignment="Center" VerticalAlignment="Center"
      FontStyle="Italic">Big D</Label>
141      <Label Name="LastnameLabel" Grid.Column="1" Grid.Row="2"
      HorizontalAlignment="Center" VerticalAlignment="Center"
      FontStyle="Italic">Slupskiy</Label>
142      <Label Name="StateLabel" Grid.Column="1" Grid.Row="3"
      HorizontalAlignment="Center" VerticalAlignment="Center"
      FontStyle="Italic">Wanted</Label>
143      <Label Name="AgeLabel" Grid.Column="1" Grid.Row="4"
      HorizontalAlignment="Center" VerticalAlignment="Center"
      FontStyle="Italic">18</Label>
144      <Label Name="HeightLabel" Grid.Column="1" Grid.Row="5"
      HorizontalAlignment="Center" VerticalAlignment="Center"
      FontStyle="Italic">189</Label>
145      <Label Name="CriminalGroupLabel" Grid.Column="1" Grid.Row="6"
      HorizontalAlignment="Center" VerticalAlignment="Center"
      FontStyle="Italic">ISIS</Label>
146
147
148      </Grid>
149      <Grid Grid.Row="0" Grid.Column="1">
150          <Grid.ColumnDefinitions>
151              <ColumnDefinition></ColumnDefinition>
152              <ColumnDefinition></ColumnDefinition>
153          </Grid.ColumnDefinitions>
154          <Grid.RowDefinitions>
155              <RowDefinition></RowDefinition>
156              <RowDefinition></RowDefinition>
157              <RowDefinition></RowDefinition>
158              <RowDefinition></RowDefinition>
159              <RowDefinition Height="2*"></RowDefinition>
160              <RowDefinition></RowDefinition>
161              <RowDefinition Height="2*"></RowDefinition>
162          </Grid.RowDefinitions>
163          <Label Grid.Column="0" Grid.Row="0" HorizontalAlignment="Center"
      VerticalAlignment="Center" FontWeight="Bold">Color of eyes</

```

```

    Label>
164    <Label Grid.Column="0" Grid.Row="1" HorizontalAlignment="Center"  7
        VerticalAlignment="Center" FontWeight="Bold">Color of hair</  7
        Label>
165    <Label Grid.Column="0" Grid.Row="2" HorizontalAlignment="Center"  7
        VerticalAlignment="Center" FontWeight="Bold">Sex</Label>
166
167    <Label Name="ColorOfEyesLabel" Grid.Column="1" Grid.Row="0"  7
        HorizontalAlignment="Center" VerticalAlignment="Center"  7
        FontStyle="Italic">Black</Label>
168    <Label Name="ColorOfHairLabel" Grid.Column="1" Grid.Row="1"  7
        HorizontalAlignment="Center" VerticalAlignment="Center"  7
        FontStyle="Italic">Black</Label>
169    <Label Name="SexLabel" Grid.Column="1" Grid.Row="2"  7
        HorizontalAlignment="Center" VerticalAlignment="Center"  7
        FontStyle="Italic">Male</Label>
170
171    <Label Grid.Column="0" Grid.ColumnSpan="2" Grid.Row="3"  7
        HorizontalAlignment="Center" VerticalAlignment="Center"  7
        FontWeight="Bold">Languages</Label>
172    <ScrollView Grid.Column="0" Grid.Row="4" Grid.ColumnSpan="2" >
173        <TextBlock Name="LanguagesTextBlock" TextWrapping="Wrap"  7
            HorizontalAlignment="Center" VerticalAlignment="Center"  7
            FontStyle="Italic">English Russian</TextBlock>
174    </ScrollView>
175    <Label Grid.Column="0" Grid.ColumnSpan="2" Grid.Row="5"  7
        HorizontalAlignment="Center" VerticalAlignment="Center"  7
        FontWeight="Bold">Charges</Label>
176    <ScrollView Grid.Column="0" Grid.Row="6" Grid.ColumnSpan="2">
177        <TextBlock Name="ChargesTextBlock" TextWrapping="Wrap"  7
            HorizontalAlignment="Center" VerticalAlignment="Center"  7
            FontStyle="Italic">Murder Robbery</TextBlock>
178    </ScrollView>
179
180    </Grid>
181    <Grid Grid.Row="1" Grid.Column="1">
182        <Grid.ColumnDefinitions>
183            <ColumnDefinition></ColumnDefinition>
184            <ColumnDefinition></ColumnDefinition>
185        </Grid.ColumnDefinitions>
186        <Grid.RowDefinitions>
187            <RowDefinition></RowDefinition>
188            <RowDefinition></RowDefinition>
189            <RowDefinition></RowDefinition>
190            <RowDefinition></RowDefinition>
191            <RowDefinition></RowDefinition>
192            <RowDefinition></RowDefinition>
193        </Grid.RowDefinitions>
194        <Label Grid.Column="0" Grid.Row="0" HorizontalAlignment="Center"  7
            VerticalAlignment="Center" FontWeight="Bold">Citizenship</Label>
195        <Label Grid.Column="0" Grid.Row="1" HorizontalAlignment="Center"  7
            VerticalAlignment="Center" FontWeight="Bold">Country of birth</  7
            Label>
196        <Label Grid.Column="0" Grid.Row="2" HorizontalAlignment="Center"  7
            VerticalAlignment="Center" FontWeight="Bold">Place</Label>
197        <Label Grid.Column="0" Grid.Row="3" HorizontalAlignment="Center"  7

```

```

    VerticalAlignment="Center" FontWeight="Bold">Date of birth</
    Label>
198    <Label Grid.Column="0" Grid.Row="4" HorizontalAlignment="Center"
    VerticalAlignment="Center" FontWeight="Bold">Last lived in</
    Label>
199    <Label Grid.Column="0" Grid.Row="5" HorizontalAlignment="Center"
    VerticalAlignment="Center" FontWeight="Bold">Place</Label>
200
201    <Label Name="CitizenshipLabel" Grid.Column="1" Grid.Row="0"
    HorizontalAlignment="Center" VerticalAlignment="Center"
    FontStyle="Italic">Ukraine</Label>
202    <Label Name="CountryOfBirthLabel" Grid.Column="1" Grid.Row="1"
    HorizontalAlignment="Center" VerticalAlignment="Center"
    FontStyle="Italic">Ukraine</Label>
203    <Label Name="PlaceOfBirthLabel" Grid.Column="1" Grid.Row="2"
    HorizontalAlignment="Center" VerticalAlignment="Center"
    FontStyle="Italic">Kharkiv</Label>
204    <Label Name="DateOfBirthLabel" Grid.Column="1" Grid.Row="3"
    HorizontalAlignment="Center" VerticalAlignment="Center"
    FontStyle="Italic">27.02.98</Label>
205    <Label Name="LastLivedInLabel" Grid.Column="1" Grid.Row="4"
    HorizontalAlignment="Center" VerticalAlignment="Center"
    FontStyle="Italic">Ukraine</Label>
206    <Label Name="LastPlaceOfLiving" Grid.Column="1" Grid.Row="5"
    HorizontalAlignment="Center" VerticalAlignment="Center"
    FontStyle="Italic">Kharkiv</Label>
207    </Grid>
208    <ScrollView Grid.Row="0" Grid.RowSpan="2" Grid.Column="2">
209        <TextBlock Name="SpecialSignsTextBlock" TextWrapping="Wrap"
    HorizontalAlignment="Center" FontStyle="Italic">
210            Some special information about this bastard. Blah, blah,
    blah,blah, blah, blah
211            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.  
212         </TextBlock>  
213     </ScrollViewer>  
214     <Button Name="ChangeProfileButton" Grid.Row="2" Grid.Column="1"  
        Click="ChangeProfileButton_OnClick">Change profile</Button>  
215 </Grid>  
216 </controls:MetroWindow>  
217
```

```

1  using System;
2  using System.Collections.Generic;
3  using System.IO;
4  using System.Linq;
5  using System.Windows;
6  using System.Windows.Media.Imaging;
7  using InterpolDatabaseProject.Model;
8  using Microsoft.CSharp.RuntimeBinder;
9
10 namespace InterpolDatabaseProject
11 {
12     /// <summary>
13     /// Логика взаимодействия для CriminalProfile.xaml
14     /// Окно отображает полную информацию о преступнике
15     /// </summary>
16     public partial class CriminalProfile
17     {
18         #region Fields
19         /// <summary>
20         /// Отображаемый преступник
21         /// </summary>
22         private readonly Criminal _criminal;
23         #endregion
24         #region Constructors
25         public CriminalProfile(Criminal criminal)
26         {
27             InitializeComponent();
28             _criminal = criminal;
29             LoadInformation();
30             SetDataSources();
31         }
32         #endregion
33
34         /// <summary>
35         /// Загружает актуальную информацию в элементы управления окна
36         /// </summary>
37         public void LoadInformation()
38         {
39             if (_criminal == null) return;
40             const string unknown = "Unknown";
41             const string nodata = "No data";
42
43             Title = "Criminal #" + _criminal.Id;
44             if (_criminal.PhotoFileName != null) CriminalImage.Source = new
45                 BitmapImage(new Uri(Path.GetFullPath("../..\\Storage\\Files\\"
46                     + _criminal.PhotoFileName)));
47
48             ForenameLabel.Content = (_criminal.Forename=="")? unknown :
49                 _criminal.Forename;
50             CodenameLabel.Content = (_criminal.CodeName == "") ? unknown :
51                 _criminal.CodeName;
52             LastnameLabel.Content = (_criminal.Lastname == "") ? unknown :
53                 _criminal.Lastname;
54             StateLabel.Content = _criminal.State;
55             HeightLabel.Content = (_criminal.Height==null)? unknown :
56                 _criminal.Height.ToString();

```

```

51     AgeLabel.Content = (_criminal.Age == null) ? unknown :
        _criminal.Age.ToString();
52     CriminalGroupLabel.Content =
        _criminal.CriminalGroupMembership?.ToString() ?? "Singleton";
53     ColorOfEyesLabel.Content = _criminal.ColorOfEye;
54     ColorOfHairLabel.Content = _criminal.ColorOfHair;
55     SexLabel.Content = _criminal.Sex;
56     LanguagesTextBlock.Text = _criminal.Languages.Count > 0 ?
57     _criminal.Languages.Aggregate("", (current, language) =>
        current + language + " ")
58     :
59     noData;
60     ChargesTextBlock.Text = _criminal.Charges.Count > 0 ?
61     _criminal.Charges.Aggregate("", (current, charge) => current +
        charge + " ")
62     :
63     noData;
64     CitizenshipLabel.Content = _criminal.Citizenship;
65     CountryOfBirthLabel.Content = _criminal.BirthCountry;
66     PlaceOfBirthLabel.Content = (_criminal.Birthplace=="")?
        unknown:_criminal.Birthplace;
67     DateOfBirthLabel.Content = _criminal.Birthdate?.ToShortDateString
        () ?? unknown;
68     LastLivedInLabel.Content = _criminal.LastLivingCountry;
69     LastPlaceOfLiving.Content = (_criminal.LastLivingPlace == "") ?
        unknown : _criminal.LastLivingPlace;
70     SpecialSignsTextBlock.Text = (_criminal.SpecialSigns == "") ?
        noData : _criminal.SpecialSigns;
71 }
72
73 /// <summary>
74 /// Сохраняет внесенные в преступника изменения
75 /// </summary>
76 private void ChangeCriminalFlyout_SubmitButton_OnClick(object sender,
    RoutedEventArgs e)
77 {
78     List<Language> languages =
        ChangeCriminalFlyout_LanguagesListBox.SelectedItems.Cast<Language>
        ().ToList();
79     List<Crime> charges =
        ChangeCriminalFlyout_ChargesListBox.SelectedItems.Cast<Crime>
        ().ToList();
80
81     _criminal.Lastname = ChangeCriminalFlyout_LastnameTextBox.Text;
82     _criminal.Forename = ChangeCriminalFlyout_ForenameTextBox.Text;
83     _criminal.CodeName = ChangeCriminalFlyout_CodenameTextBox.Text;
84     _criminal.Height =
        (ChangeCriminalFlyout_IsHeightKnownCheckBox.IsChecked.Value) ?
        (int?) ChangeCriminalFlyout_HeightSlider.Value : null;
85     _criminal.ColorOfEye = (EyeColor)
        ChangeCriminalFlyout_EyeColorComboBox.SelectedItem;
86     _criminal.ColorOfHair = (HairColor)
        ChangeCriminalFlyout_HairColorComboBox.SelectedItem;
87     _criminal.Sex = (SexOptions)
        ChangeCriminalFlyout_SexComboBox.SelectedItem;
88     _criminal.SpecialSigns =

```

```
        ChangeCriminalFlyout_SpecialSignsTextBox.Text;
89        _criminal.Citizenship = (Country)
        ChangeCriminalFlyout_CitizenshipComboBox.SelectedItem;
90        _criminal.BirthCountry = (Country)
        ChangeCriminalFlyout_BirthCountryComboBox.SelectedItem;
91        _criminal.Birthplace =
        ChangeCriminalFlyout_BirthplaceTextBox.Text;
92        _criminal.Birthdate =
        ChangeCriminalFlyout_BirthdateDatePicker.SelectedDate;
93        _criminal.LastLivingCountry = (Country)
        ChangeCriminalFlyout_LastLivingCountryComboBox.SelectedIndex;
94        _criminal.LastLivingPlace =
        ChangeCriminalFlyout_LastLivingPlaceTextBox.Text;
95        _criminal.Languages = languages;
96        _criminal.Charges = charges;
97        _criminal.State = (CriminalStateOptions)
        ChangeCriminalFlyout_CurrentStateComboBox.SelectedItem;
98        _criminal.UnsetCriminalGroup();
99        if(ChangeCriminalFlyout_CriminalGroupComboBox.SelectedIndex != -1)
100            _criminal.SetCriminalGroup(Database.CriminalGroups[
101                ((KeyValuePair<int, CriminalGroup>)
        ChangeCriminalFlyout_CriminalGroupComboBox.SelectedItem).Key]);
102
103        if (ChangeCriminalFlyout_PhotoFilePath.Text != "")
104            Database.ChangeCriminalsPhoto
        (ChangeCriminalFlyout_PhotoFilePath.Text, _criminal.Id);
105
106        LoadInformation();
107        ChangeCriminalFlyout.IsOpen = false;
108    }
109
110    /// <summary>
111    /// Позволяет выбрать новую фотографию для преступника
112    /// </summary>
113    private void ChangeCriminalFlyout_PickButton_OnClick(object sender,
        RoutedEventArgs e)
114    {
115        Microsoft.Win32.OpenFileDialog dlg = new
        Microsoft.Win32.OpenFileDialog
116        {
117            DefaultExt = ".png",
118            Filter =
119                "Images (*.jpeg, *.png, *.jpg)|*.jpeg;*.png;*.jpg"
120        };
121        bool? result = dlg.ShowDialog();
122        if (result == true)
123        {
124            ChangeCriminalFlyout_PhotoFilePath.Text = dlg.FileName;
125        }
126    }
127
128    /// <summary>
129    /// Подгружает информацию о преступнике в элементы управления Flyout
130    /// </summary>
131    private void ChangeCriminalFlyout_OnIsOpenChanged(object sender,
```



```

RoutedEventArgs e)
{
    MainGrid.IsEnabled = !ChangeCriminalFlyout.IsOpen;
    if (ChangeCriminalFlyout.IsOpen)
        LoadChangeCriminalData();
}

/// <summary>
/// Подгружает текущую информацию о преступнике в Flyout
/// </summary>
private void LoadChangeCriminalData()
{
    ChangeCriminalFlyout_ForenameTextBox.Text = _criminal.Forename;
    ChangeCriminalFlyout_CodenameTextBox.Text = _criminal.CodeName;
    ChangeCriminalFlyout_LastnameTextBox.Text = _criminal.Lastname;
    ChangeCriminalFlyout_HeightSlider.Value = _criminal.Height ?? 0;
    ChangeCriminalFlyout_EyeColorComboBox.SelectedIndex = (int)
        _criminal.ColorOfEye;
    ChangeCriminalFlyout_HairColorComboBox.SelectedIndex = (int)
        _criminal.ColorOfHair;
    ChangeCriminalFlyout_SexComboBox.SelectedIndex = (int)
        _criminal.Sex;
    ChangeCriminalFlyout_CitizenshipComboBox.SelectedIndex = (int)
        _criminal.Citizenship;
    ChangeCriminalFlyout_BirthCountryComboBox.SelectedIndex = (int)
        _criminal.BirthCountry;
    ChangeCriminalFlyout_BirthplaceTextBox.Text =
        _criminal.Birthplace;
    ChangeCriminalFlyout_BirthdateDatePicker.SelectedDate =
        _criminal.Birthdate;
    ChangeCriminalFlyout_LastLivingCountryComboBox.SelectedIndex =
        (int)_criminal.LastLivingCountry;
    ChangeCriminalFlyout_LastLivingPlaceTextBox.Text =
        _criminal.LastLivingPlace;
    ChangeCriminalFlyout_CurrentStateComboBox.SelectedIndex = (int)
        _criminal.State;
    ChangeCriminalFlyout_SpecialSignsTextBox.Text =
        _criminal.SpecialSigns;
    foreach (var language in _criminal.Languages)
        ChangeCriminalFlyout_LanguagesListBox.SelectedItems.Add
            (ChangeCriminalFlyout_LanguagesListBox.Items[(int)
                language]);
    foreach (var charge in _criminal.Charges)
        ChangeCriminalFlyout_ChargesListBox.SelectedItems.Add
            (ChangeCriminalFlyout_ChargesListBox.Items[(int)charge]);
    if (_criminal.CriminalGroupMembership == null) return;
    for (int i = 0; i < Database.CriminalGroups.Values.Count; i++)
    {
        if (_criminal.CriminalGroupMembership.Id !=
            Database.CriminalGroups.Values.ElementAt(i).Id) continue;
        ChangeCriminalFlyout_CriminalGroupComboBox.SelectedIndex = i;
        break;
    }
}

/// <summary>

```

```
172     /// Устанавливает источники данных для элементов управления
173     /// </summary>
174     private void SetDataSources()
175     {
176         ChangeCriminalFlyout_EyeColorComboBox.ItemsSource =
177             Enum.GetValues(typeof(EyeColor)).Cast<EyeColor>();
178         ChangeCriminalFlyout_HairColorComboBox.ItemsSource =
179             Enum.GetValues(typeof(HairColor)).Cast<HairColor>();
180         ChangeCriminalFlyout_SexComboBox.ItemsSource =
181             Enum.GetValues(typeof(SexOptions)).Cast<SexOptions>();
182         ChangeCriminalFlyout_CitizenshipComboBox.ItemsSource =
183             Enum.GetValues(typeof(Country)).Cast<Country>();
184         ChangeCriminalFlyout_BirthCountryComboBox.ItemsSource =
185             Enum.GetValues(typeof(Country)).Cast<Country>();
186         ChangeCriminalFlyout_LastLivingCountryComboBox.ItemsSource =
187             Enum.GetValues(typeof(Country)).Cast<Country>();
188         ChangeCriminalFlyout_CurrentStateComboBox.ItemsSource =
189             Enum.GetValues(typeof
190                 (CriminalStateOptions)).Cast<CriminalStateOptions>();
191         ChangeCriminalFlyout_LanguagesListBox.ItemsSource =
192             Enum.GetValues(typeof(Language)).Cast<Language>();
193         ChangeCriminalFlyout_ChargesListBox.ItemsSource =
194             Enum.GetValues(typeof(Crime)).Cast<Crime>();
195         ChangeCriminalFlyout_CriminalGroupComboBox.ItemsSource =
196             Database.CriminalGroups;
197     }
198     /// <summary>
199     /// Открывает Flyout для изменения данных о преступнике
200     /// </summary>
201     private void ChangeProfileButton_OnClick(object sender,
202         RoutedEventArgs e) => ChangeCriminalFlyout.IsOpen = true;
203 }
```

```
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"
15             Click="OkButton_OnClick" />
16         <Button Name="CancelButton" Content="Cancel"
17             Click="CancelButton_OnClick" />
18     </StackPanel>
19 </controls:MetroWindow>
20
```

```
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
8     /// </summary>
9     public partial class DeleteDialog
10    {
11        #region Properties
12        /// <summary>
13        /// Ответ пользователя
14        /// </summary>
15        public bool Result { get; private set; }
16        #endregion
17        #region Constructors
18        /// <summary>
19        /// Основной конструктор
20        /// </summary>
21        public DeleteDialog()
22        {
23            InitializeComponent();
24        }
25        #endregion
26        #region Methods
27        /// <summary>
28        /// Закрывает окно с положительным ответом
29        /// </summary>
30        private void OkButton_OnClick(object sender, RoutedEventArgs e)
31        {
32            Result = true;
33            Close();
34        }
35        /// <summary>
36        /// Закрывает окно с отрицательным ответом
37        /// </summary>
38        private void CancelButton_OnClick(object sender, RoutedEventArgs e)
39        {
40            Result = false;
41            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-
7          namespace:MahApps.Metro.Controls;assembly=MahApps.Metro"
8      xmlns:interpolDatabaseProject="clr-namespace:InterpolDatabaseProject"
9      mc:Ignorable="d"
10     Title="Interpol database" MinHeight="600" Height="600" MinWidth="800"
11     Width="800"
12     Icon="Storage/Files/icon.png"
13     GlowBrush="{DynamicResource AccentColorBrush}"
14     WindowStartupLocation="CenterScreen"
15     Closing="MainWindow_OnClosing">
16     <Controls:MetroWindow.Flyouts>
17         <Controls:FlyoutsControl>
18             <Controls:Flyout Name="AddNewCriminalFlyout" Header="New criminal"
19                 Position="Right"
20                 Width="600"
21                 IsOpenChanged="Flyout_OnIsOpenChanged">
22                 <ScrollViewer>
23                     <Grid>
24                         <Grid.Resources>
25                             <Style TargetType="{x:Type Label}">
26                                 <Setter Property="Margin" Value="10" />
27                             </Style>
28                         </Grid.Resources>
29                         <Grid.ColumnDefinitions>
30                             <ColumnDefinition/>
31                             <ColumnDefinition/>
32                         </Grid.ColumnDefinitions>
33                         <Grid.RowDefinitions>
34                             <RowDefinition/>
35                             <RowDefinition/>
36                             <RowDefinition/>
37                             <RowDefinition/>
38                             <RowDefinition/>
39                             <RowDefinition/>
40                             <RowDefinition/>
41                             <RowDefinition/>
42                             <RowDefinition/>
43                             <RowDefinition/>
44                             <RowDefinition/>
45                             <RowDefinition/>
46                             <RowDefinition/>
47                             <RowDefinition/>
48                             <RowDefinition/>
49                         </Grid.RowDefinitions>
50                         <Label Grid.Column="0" Grid.Row="0">Forename</Label>

```

```

51         <Label Grid.Column="0" Grid.Row="1">Codename</Label>
52         <Label Grid.Column="0" Grid.Row="2">Lastname</Label>
53         <Label Grid.Column="0" Grid.Row="3">Height</Label>
54         <Label Grid.Column="0" Grid.Row="4">Color of eye</
Label>
55         <Label Grid.Column="0" Grid.Row="5">Color of hair</
Label>
56         <Label Grid.Column="0" Grid.Row="6">Sex</Label>
57         <Label Grid.Column="0" Grid.Row="7">Special Signs</
Label>
58         <Label Grid.Column="0" Grid.Row="8">Citizenship</
Label>
59         <Label Grid.Column="0" Grid.Row="9">Country of birth</
Label>
60         <Label Grid.Column="0" Grid.Row="10">Place of birth</
Label>
61         <Label Grid.Column="0" Grid.Row="11">Date of birth</
Label>
62         <Label Grid.Column="0" Grid.Row="12">Last lived in
(Country)</Label>
63         <Label Grid.Column="0" Grid.Row="13">Last lived in
(Place)</Label>
64         <Label Grid.Column="0" Grid.Row="14">Languages</Label>
65         <Label Grid.Column="0" Grid.Row="15">Current state</
Label>
66         <Label Grid.Column="0" Grid.Row="16">Photo</Label>
67         <Label Grid.Column="0" Grid.Row="18">Criminal group</
Label>
68         <Label Grid.Column="0" Grid.Row="19">Charges</Label>
69
70
71         <TextBox Name="AddNewCriminal_ForenameTextBox"
72             Tooltip="Leave empty if it is unknown"
73             Grid.Column="1" Grid.Row="0"
HorizontalAlignment="Stretch"
VerticalAlignment="Center">
74
75         </TextBox>
76         <TextBox Name="AddNewCriminal_CodenameTextBox"
77             Tooltip="Leave empty if it is unknown"
78             Grid.Column="1" Grid.Row="1"
79             HorizontalAlignment="Stretch"
80             VerticalAlignment="Center">
81
82         </TextBox>
83         <TextBox Name="AddNewCriminal_LastnameTextBox"
84             Tooltip="Leave empty if it is unknown"
85             Grid.Column="1" Grid.Row="2"
86             HorizontalAlignment="Stretch"
87             VerticalAlignment="Center">
88
89         </TextBox>
90         <StackPanel Grid.Column="1" Grid.Row="3"
Orientation="Horizontal">
<CheckBox
Name="AddNewCriminal_IsHeightKnownCheckBox"
IsChecked="True"/>
<Slider IsEnabled="{Binding
ElementName=AddNewCriminal_IsHeightKnownCheckBox,

```

```

91         Path=IsChecked}" Width="100"
Name="AddNewCriminal_HeightSlider"
92         HorizontalAlignment="Stretch"
VerticalAlignment="Center" Minimum="0"
93         Maximum="250" SmallChange="1"
LargeChange="5" Value="100" IsSnapToTickEnabled="True"
94         TickFrequency="1">
95     </Slider>
96     <TextBox IsEnabled="{Binding
ElementName=AddNewCriminal_IsHeightKnownCheckBox,
Path=IsChecked}"
97         HorizontalAlignment="Stretch"
VerticalAlignment="Center" IsReadOnly="True"
98         Text="{Binding
ElementName=AddNewCriminal_HeightSlider, Path=Value}"/>
99     </StackPanel>
100     <ComboBox Name="AddNewCriminal_EyeColorComboBox"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
Grid.Column="1" SelectedIndex="0" Grid.Row="4"/>
101     <ComboBox Name="AddNewCriminal_HairColorComboBox"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
Grid.Column="1" SelectedIndex="0" Grid.Row="5"/>
102     <ComboBox Name="AddNewCriminal_SexComboBox"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
Grid.Column="1" Grid.Row="6" SelectedIndex="0"/>
103     <TextBox Name="AddNewCriminal_SpecialSignsTextBox"
ToolTip="Leave empty if it is unknown"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
Grid.Column="1" Grid.Row="7" TextWrapping="Wrap"></
TextBox>
104     <ComboBox Name="AddNewCriminal_CitizenshipComboBox"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
Grid.Column="1" Grid.Row="8" SelectedIndex="0"/>
105     <ComboBox Name="AddNewCriminal_BirthCountryComboBox"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
Grid.Column="1" Grid.Row="9" SelectedIndex="0"/>
106     <TextBox Name="AddNewCriminal_BirthplaceTextBox"
ToolTip="Leave empty if it is unknown" Grid.Column="1"
Grid.Row="10" HorizontalAlignment="Stretch"
VerticalAlignment="Center"></TextBox>
107     <DatePicker Name="AddNewCriminal_BirthdateDatePicker"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
Grid.Column="1" Grid.Row="11" ></DatePicker>
108     <ComboBox
Name="AddNewCriminal_LastLivingCountryComboBox"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
Grid.Column="1" Grid.Row="12" SelectedIndex="0"/>
109     <TextBox Name="AddNewCriminal_LastLivingPlaceTextBox"
ToolTip="Leave empty if it is unknown" Grid.Column="1"
Grid.Row="13" HorizontalAlignment="Stretch"
VerticalAlignment="Center"></TextBox>
110     <ScrollViewer Grid.Column="1" Grid.Row="14"
MaxHeight="100">
111     <ListBox Name="AddNewCriminal_LanguagesListBox"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
SelectionMode="Multiple"/>

```

```

112         </ScrollViewer>
113         <ComboBox Name="AddNewCriminal_CurrentStateComboBox"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
Grid.Column="1" Grid.Row="15" SelectedIndex="0"/>
114         <Button Click="AddNewCriminal_PickButton_Click"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
Grid.Column="1" Grid.Row="16">Pick file</Button>
115         <TextBlock Grid.Column="1" Grid.Row="17"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
Name="AddNewCriminal_PhotoFilePath"></TextBlock>
116         <ComboBox Name="AddNewCriminal_CriminalGroupComboBox"
DisplayMemberPath="Value.Name" ToolTip="Leave empty if
criminal is singleton" HorizontalAlignment="Stretch"
VerticalAlignment="Center" Grid.Column="1" Grid.Row="18"/>
117         <ScrollViewer Grid.Column="1" Grid.Row="19"
MaxHeight="100">
118             <ListBox Name="AddNewCriminal_ChargesListBox"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
SelectionMode="Multiple"/>
119         </ScrollViewer>
120         <Button Grid.Column="0" Grid.Row="20"
Grid.ColumnSpan="2"
Click="AddNewCriminal_SubmitButton_Click">SUBMIT</Button>
121     </Grid>
122 </ScrollViewer>
123 </Controls:Flyout>
124 <Controls:Flyout Name="AddNewCriminalGroupFlyout" Header="New
criminal group" Position="Left" Width="400"
IsOpenChanged="Flyout_OnIsOpenChanged">
125     <StackPanel Width="300">
126         <Label>Name</Label>
127         <TextBox Name="AddNewCriminalGroup_NameTextBox"
HorizontalAlignment="Stretch"
VerticalAlignment="Center"></TextBox>
128         <Label>Additional data</Label>
129         <TextBox Name="AddNewCriminalGroup_AdditionalDataTextBox"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
TextWrapping="Wrap"></TextBox>
130         <Button Name="AddNewCriminalGroup_SubmitButton"
Click="AddNewCriminalGroup_SubmitButton_OnClick">Submit</
Button>
131     </StackPanel>
132 </Controls:Flyout>
133 <Controls:Flyout Name="ChangeCriminalGroupFlyout" Header="Change
criminal group" Position="Left" Width="400"
IsOpenChanged="Flyout_OnIsOpenChanged">
134     <ScrollViewer>
135         <StackPanel Width="300">
136             <Label>Name</Label>
137             <TextBox Name="ChangeCriminalGroupFlyout_NameTextBox"
HorizontalAlignment="Stretch" VerticalAlignment="Center"/>
138             <Label>Additional data</Label>
139             <TextBox
Name="ChangeCriminalGroupFlyout_AdditionalDataTextBox"
HorizontalAlignment="Stretch" VerticalAlignment="Center"
TextWrapping="Wrap"/>

```



```

140         <Button Name="ChangeCriminalGroupFlyout_SubmitButton"
Click="ChangeCriminalGroupFlyout_SubmitButton_OnClick">Sub
mit</Button>
141     </StackPanel>
142 </ScrollViewer>
143 </Controls:Flyout>
144 </Controls:FlyoutsControl>
145 </Controls:MetroWindow.Flyouts>
146 <Controls:MetroWindow.RightWindowCommands>
147     <Controls:WindowCommands>
148         <Button Name="RightWindowCommands_SaveButton"
Click="RightWindowCommands_SaveButton_OnClick" Content="save" />
149         <Button Name="RightWindowCommands_RestoreButton"
Click="RightWindowCommands_RestoreButton_OnClick"
Content="restore" />
150         <Button Name="RightWindowCommands_AboutButton"
Click="RightWindowCommands_AboutButton_OnClick"
Content="about" />
151     </Controls:WindowCommands>
152 </Controls:MetroWindow.RightWindowCommands>
153 <Grid Name="MainGrid">
154     <Grid.ColumnDefinitions>
155         <ColumnDefinition Width="3*"></ColumnDefinition>
156         <ColumnDefinition Width="7*"></ColumnDefinition>
157     </Grid.ColumnDefinitions>
158     <Grid.RowDefinitions>
159         <RowDefinition Height="9*"></RowDefinition>
160         <RowDefinition Height="1*"></RowDefinition>
161     </Grid.RowDefinitions>
162     <ScrollViewer Grid.Row="0" Grid.Column="0" Grid.RowSpan="2">
163         <StackPanel>
164             <Expander Header="Current state">
165                 <ListBox Name="Filter_StateListBox"
SelectionChanged="Filter_ControlValueChanged"
SelectionMode="Multiple" MaxHeight="100"/>
166             </Expander>
167             <Expander Header="Information">
168                 <StackPanel>
169                     <Label Content="Lastname"/>
170                     <TextBox Name="Filter_LastnameTextBox"
TextChanged="Filter_ControlValueChanged" Text=""/>
171                     <Label Content="Forename"/>
172                     <TextBox Name="Filter_ForenameTextBox"
TextChanged="Filter_ControlValueChanged" Text=""/>
173                     <Label Content="Codename"/>
174                     <TextBox Name="Filter_CodenameTextBox"
TextChanged="Filter_ControlValueChanged" Text=""/>
175                 </StackPanel>
176             </Expander>
177             <Expander Header="Characteristics">
178                 <StackPanel>
179                     <Label Content="Age"/>
180                     <StackPanel Orientation="Horizontal">
181                         <CheckBox Name="Filter_IsAgeKnown"
IsChecked="True" Unchecked="Filter_ControlValueChanged"
Checked="Filter_ControlValueChanged"/>

```

```

182         <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>
183         </StackPanel>
184         <Label Content="Height"/>
185         <StackPanel Orientation="Horizontal">
186             <CheckBox Name="Filter_IsHeightKnown"
               Checked="Filter_ControlValueChanged"
               Unchecked="Filter_ControlValueChanged" IsChecked="True"/>
187             <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>
188             </StackPanel>
189             <Label Content="Color of eye"/>
190             <ComboBox Name="Filter_EyeColorCombobox"
               SelectionChanged="Filter_ControlValueChanged"
               MaxHeight="100" IsEditable="True"/>
191             <Label Content="Color of hair"/>
192             <ComboBox Name="Filter_HairColorCombobox"
               SelectionChanged="Filter_ControlValueChanged"
               MaxHeight="100" IsEditable="True"/>
193             <Label Content="Sex"/>
194             <ComboBox Name="Filter_SexCombobox"
               SelectionChanged="Filter_ControlValueChanged"
               MaxHeight="100" IsEditable="True"/>
195             <Label Content="Special signs"/>
196             <TextBox Name="Filter_SpecialSignsTextbox"
               TextChanged="Filter_ControlValueChanged"/>
197             <Label Content="Languages"/>
198             <ListBox Name="Filter_LanguagesListbox"
               SelectionChanged="Filter_ControlValueChanged"
               SelectionMode="Multiple" MaxHeight="100"/>
199         </StackPanel>
200     </Expander>
201     <Expander Header="Locations">
202         <StackPanel>
203             <Label Content="Citizenship"/>
204             <ComboBox Name="Filter_CitizenshipComboBox"
               SelectionChanged="Filter_ControlValueChanged"
               MaxHeight="100" IsEditable="True"/>
205             <Label Content="Country of birth"/>
206             <ComboBox Name="Filter_BirthCountryComboBox"
               SelectionChanged="Filter_ControlValueChanged"
               MaxHeight="100" IsEditable="True"/>
207             <Label Content="Place of birth"/>

```

```

208         <TextBox Name="Filter_PlaceOfBirthTextBox"
TextChanged="Filter_ControlValueChanged"/>
209         <Label Content="Last living country"/>
210         <ComboBox Name="Filter_LastLivingCountryComboBox"
SelectionChanged="Filter_ControlValueChanged"
MaxHeight="100" IsEditable="True"/>
211         <Label Content="Last living place"/>
212         <TextBox Name="Filter_LastLivingPlaceTextBox"
TextChanged="Filter_ControlValueChanged"/>
213     </StackPanel>
214 </Expander>
215 <Expander Header="Criminal groups">
216     <StackPanel>
217         <Button Name="Filter_ShowAllCriminals"
Click="Filter_ShowAllCriminals_OnClick">Criminals from all
groups</Button>
218         <Grid>
219             <Grid.ColumnDefinitions>
220                 <ColumnDefinition></ColumnDefinition>
221                 <ColumnDefinition></ColumnDefinition>
222             </Grid.ColumnDefinitions>
223             <Button
Name="CriminalGroupsListBox_AddGroupButton"
Click="CriminalGroupsListBox_AddGroupButton_OnClick"
Grid.Column="0" Style="{StaticResource
AccentedSquareButtonStyle}">New group</Button>
224             <Button
Name="CriminalGroupsListBox_DeleteGroupButton"
IsEnabled="False"
Click="CriminalGroupsListBox_DeleteGroupButton_OnClick"
Grid.Column="1" Style="{StaticResource
AccentedSquareButtonStyle}">Delete group</Button>
225         </Grid>
226         <ListBox Name="CriminalGroupsListBox"
DisplayMemberPath="Value.Name"
SelectionChanged="CriminalGroupsListBox_OnSelectionChanged"
MaxHeight="100"/>
227     </StackPanel>
228 </Expander>
229 </StackPanel>
230 </ScrollView>
231 <ScrollView Grid.Row="0" Grid.Column="1"
HorizontalScrollBarVisibility="Hidden">
232     <StackPanel>
233         <Grid Name="CriminalGroupInformationGrid" MaxHeight="200"
Visibility="Collapsed">
234             <Grid.RowDefinitions>
235                 <RowDefinition></RowDefinition>
236                 <RowDefinition></RowDefinition>
237                 <RowDefinition></RowDefinition>
238             </Grid.RowDefinitions>
239             <Label Name="CriminalGroupInformationGrid_Name"
Grid.Row="0" HorizontalAlignment="Center"
FontSize="21.333" FontWeight="Bold">
240                 <Label.Effect>
241                     <DropShadowEffect ShadowDepth="0.1"/>

```

```

242         </Label.Effect>
243     </Label>
244     <ScrollView Grid.Row="1">
245         <TextBlock
246             Name="CriminalGroupInformationGrid_Information"
247             MaxWidth="500" TextWrapping="Wrap"
248             HorizontalAlignment="Center"/>
249     </ScrollView>
250     <Button
251         Name="CriminalGroupInformationGrid_ChangeDataButton"
252         Grid.Row="2"
253         Click="CriminalGroupInformationGrid_ChangeDataButton_OnCli
254         ck">Change criminal group data</Button>
255 </Grid>
256 <ListBox x:Name="CriminalsListBox"
257     SelectionChanged="CriminalsListBox_OnSelectionChanged"
258     BorderThickness="0" ItemsSource="{Binding CriminalsToShow}"
259     Tag="{Binding Id}">
260     <ListBox.ItemTemplate>
261         <DataTemplate>
262             <Grid Height="150">
263                 <Grid.ColumnDefinitions>
264                     <ColumnDefinition Width="1*"></
265                     ColumnDefinition>
266                     <ColumnDefinition Width="5*"></
267                     ColumnDefinition>
268                 </Grid.ColumnDefinitions>
269                 <StackPanel Grid.Column="1">
270                     <Label Content="{Binding TotalName}"
271                         FontSize="18"/>
272                     <Label Content="{Binding Citizenship}"
273                         FontSize="16"/>
274                 </StackPanel>
275                 <Image Grid.Column="0" Width="150"
276                     Source="{Binding Image}" Stretch="Uniform"/>
277                 <Label Content="{Binding State}"
278                     HorizontalAlignment="Center" VerticalAlignment="Bottom"
279                     FontFamily="Courier New" FontWeight="Bold" FontSize="20"
280                     RenderTransformOrigin="0.5,0.5" Foreground="{Binding
281                     TextColor}">
282                     <Label.Effect>
283                         <DropShadowEffect/>
284                     </Label.Effect>
285                 </Label>
286             </Grid>
287         </DataTemplate>
288     </ListBox.ItemTemplate>
289 </ListBox>
290 </StackPanel>
291 </ScrollView>
292 <StackPanel Name="CriminalActionsStackPanel" Grid.Column="1"
293     Grid.Row="1" Orientation="Horizontal">
294     <Button Name="CriminalActions_AddCriminalButton"
295         Click="CriminalActions_AddCriminalButton_OnClick"
296         Style="{StaticResource AccentedSquareButtonStyle}" Margin="5"
297         VerticalAlignment="Center" HorizontalAlignment="Center"

```

```
275         Width="100">Add criminal</Button>
        <Button Name="CriminalActions_OpenProfileButton"
        Click="CriminalActions_OpenProfileButton_OnClick"
        IsEnabled="False" Style="{StaticResource
        AccentedSquareButtonStyle}" Margin="5"
        VerticalAlignment="Center" HorizontalAlignment="Center"
        Width="100">Open profile</Button>
276     <Button Name="CriminalActions_DeleteCriminalButton"
        IsEnabled="False"
        Click="CriminalActions_DeleteCriminalButton_OnClick"
        Style="{StaticResource AccentedSquareButtonStyle}" Margin="5"
        VerticalAlignment="Center" HorizontalAlignment="Center"
        Width="100">Delete criminal</Button>
277     </StackPanel>
278 </Grid>
279 </Controls:MetroWindow>
280
```

```
1 using System;
2 using System.Collections;
3 using System.Collections.Generic;
4 using System.Collections.ObjectModel;
5 using System.ComponentModel;
6 using System.IO;
7 using System.Linq;
8 using System.Windows;
9 using System.Windows.Controls;
10 using System.Windows.Media;
11 using System.Windows.Media.Imaging;
12 using MahApps.Metro.Controls.Dialogs;
13 using InterpolDatabaseProject.Model;
14
15 namespace InterpolDatabaseProject
16 {
17     /// <summary>
18     /// Логика взаимодействия для MainWindow.xaml
19     /// </summary>
20     public partial class MainWindow
21     {
22         #region Constructors
23         /// <summary>
24         /// Основной конструктор
25         /// </summary>
26         public MainWindow()
27         {
28             InitializeComponent();
29
30             Database.RestoreData();
31             InitializeComponent();
32             SetDataSources();
33             Reload_CriminalsListBox();
34             Reload_CriminalGroups();
35         }
36         #endregion
37
38         #region Properties
39         /// <summary>
40         /// Отфильтрованная коллекция преступников
41         /// </summary>
42         public List<Criminal> FilteredCriminals => ApplyFiltersToCriminals  ➤
43             (Database.Criminals);
44
45         /// <summary>
46         /// Коллекция объектов CriminalToShow, являющаяся источником данных  ➤
47         /// при отображении списка преступников
48         /// </summary>
49         public List<CriminalsListboxItemData> CriminalsToShow
50         {
51             get
52             {
53                 var result =
54                     FilteredCriminals.Select(
55                         criminal =>
56                             new CriminalsListboxItemData(criminal.Id,  ➤
```

```

        criminal.Forename, criminal.CodeName,
55         criminal.Lastname, criminal.Age,
        criminal.Citizenship.ToString(),
56         criminal.PhotoFileName ?? "default.png",
        criminal.State)).ToList();
57     return result;
58 }
59 }
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<Criminal> ApplyFiltersToCriminals(ReadOnlyDictionary<int,
        Criminal> criminals)
70     {
71         var result = CriminalGroupsListBox.SelectedItems.Count == 0 ? new
            List<Criminal>(criminals.Values) : new List<Criminal>
            (((KeyValuePair<int, CriminalGroup>)
72             CriminalGroupsListBox.SelectedItem).Value.Members.Values);
73
74         if (Filter_StateListBox.SelectedItems.Count > 0) result =
            Filter.FilterByState(result, Filter_StateListBox.SelectedItems);
75
76         if (Filter_ForenameTextBox.Text != "") result =
            Filter.FilterByForename(result, Filter_ForenameTextBox.Text);
77         if (Filter_CodenameTextBox.Text != "") result =
            Filter.FilterByCodename(result, Filter_CodenameTextBox.Text);
78         if (Filter_LastnameTextBox.Text != "") result =
            Filter.FilterByLastname(result, Filter_LastnameTextBox.Text);
79
80         if (!Filter_IsAgeKnown.IsChecked.Value)
81             result = Filter.FilterByAge(result);
82         else if (Filter_AgeRangeSlider.LowerValue >
            Filter_AgeRangeSlider.Minimum ||
            Filter_AgeRangeSlider.UpperValue <
            Filter_AgeRangeSlider.Maximum)
83             result = Filter.FilterByAge(result, (int)
            Filter_AgeRangeSlider.LowerValue, (int)
            Filter_AgeRangeSlider.UpperValue);
84
85         if (!Filter_IsHeightKnown.IsChecked.Value)
86             result = Filter.FilterByHeight(result);
87         else if (Filter_HeightRangeSlider.LowerValue >
            Filter_HeightRangeSlider.Minimum ||
            Filter_HeightRangeSlider.UpperValue <
            Filter_HeightRangeSlider.Maximum)
88             result = Filter.FilterByHeight(result, (int)
            Filter_HeightRangeSlider.LowerValue, (int)
            Filter_HeightRangeSlider.UpperValue);
89
90         if (Filter_EyeColorComboBox.SelectedIndex != -1)

```

```

90         result = Filter.FilterByEyeColor(result, (EyeColor)
           Filter_EyeColorComboBox.SelectedItem);
91     if (Filter_HairColorComboBox.SelectedIndex != -1)
92         result = Filter.FilterByHairColor(result, (HairColor)
           Filter_HairColorComboBox.SelectedItem);
93     if (Filter_SexComboBox.SelectedIndex != -1)
94         result = Filter.FilterBySex(result, (SexOptions)
           Filter_SexComboBox.SelectedItem);
95     if (Filter_SpecialSignsTextbox.Text != "")
96         result = Filter.FilterBySpecialSigns(result,
           Filter_SpecialSignsTextbox.Text);
97     if (Filter_LanguagesListbox.SelectedItems.Count > 0)
98         result = Filter.FilterByLanguages(result,
           Filter_LanguagesListbox.SelectedItems);
99     if (Filter_CitizenshipComboBox.SelectedIndex != -1)
100         result = Filter.FilterByCitizenship(result, (Country)
           Filter_CitizenshipComboBox.SelectedItem);
101     if (Filter_BirthCountryComboBox.SelectedIndex != -1)
102         result = Filter.FilterByBirthCountry(result, (Country)
           Filter_BirthCountryComboBox.SelectedItem);
103     if (Filter_PlaceOfBirthTextbox.Text != "")
104         result = Filter.FilterByBirthPlace(result,
           Filter_PlaceOfBirthTextbox.Text);
105     if (Filter_LastLivingCountryComboBox.SelectedIndex != -1)
106         result = Filter.FilterByLastLivingCountry(result, (Country)
           Filter_LastLivingCountryComboBox.SelectedItem);
107     if (Filter_LastLivingPlaceTextbox.Text != "")
108         result = Filter.FilterByLastLivingPlace(result,
           Filter_LastLivingPlaceTextbox.Text);
109
110     return result;
111 }
112
113 /// <summary>
114 /// Устанавливает источники неизменяемых в программе данных для
           соответствующих элементов управления
115 /// </summary>
116 private void SetDataSources()
117 {
118     AddNewCriminal_EyeColorComboBox.ItemsSource =
119         Enum.GetValues(typeof(EyeColor)).Cast<EyeColor>();
120     AddNewCriminal_HairColorComboBox.ItemsSource =
121         Enum.GetValues(typeof(HairColor)).Cast<HairColor>();
122     AddNewCriminal_SexComboBox.ItemsSource =
123         Enum.GetValues(typeof(SexOptions)).Cast<SexOptions>();
124     AddNewCriminal_CitizenshipComboBox.ItemsSource =
125         Enum.GetValues(typeof(Country)).Cast<Country>();
126     AddNewCriminal_BirthCountryComboBox.ItemsSource =
127         Enum.GetValues(typeof(Country)).Cast<Country>();
128     AddNewCriminal_LastLivingCountryComboBox.ItemsSource =
129         Enum.GetValues(typeof(Country)).Cast<Country>();
130     AddNewCriminal_CurrentStateComboBox.ItemsSource =
131         Enum.GetValues(typeof(
           CriminalStateOptions)).Cast<CriminalStateOptions>();
132     AddNewCriminal_LanguagesListbox.ItemsSource =
133         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
153              (CriminalStateOptions)).Cast<CriminalStateOptions>();
154
155      /// <summary>
156      /// Перезагружает источники данных в элементах управления,
157      /// отображающих преступников
158      /// </summary>
159      private void Reload_CriminalsListBox()
160      {
161          if (CriminalsListBox != null)
162              CriminalsListBox.ItemsSource = CriminalsToShow;
163      }
164
165      /// <summary>
166      /// Перезагружает источники данных в элементах управления,
167      /// отображающих группировки
168      /// </summary>
169      private void Reload_CriminalGroups()
170      {
171          if (AddNewCriminal_CriminalGroupComboBox == null) return;
172          AddNewCriminal_CriminalGroupComboBox.ItemsSource =
173              Database.CriminalGroups;
174          CriminalGroupsListBox.ItemsSource = Database.CriminalGroups;
175      }
176
177      /// <summary>
178      /// Обработчик события Click для AddNewCriminal_PickButton.
179      /// Выбирает файл с фотографией преступника
180      /// </summary>
181      private void AddNewCriminal_PickButton_Click(object sender,
182          RoutedEventArgs e)
183      {
184          Microsoft.Win32.OpenFileDialog dlg = new
185              Microsoft.Win32.OpenFileDialog
186          {
187              DefaultExt = ".png",
188              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
197 /// основе введенных пользователем данных
198 /// </summary>
199 private void AddNewCriminal_SubmitButton_Click(object sender,
200     RoutedEventArgs e)
201 {
202     List<Language> languages =
203         AddNewCriminal_LanguagesListBox.SelectedItems.Cast<Language>
204         ().ToList();
205     List<Crime> charges =
206         AddNewCriminal_ChargesListBox.SelectedItems.Cast<Crime>().ToList
207         ();
208
209     Database.AddCriminal(new Criminal(
210         AddNewCriminal_LastnameTextBox.Text,
211         AddNewCriminal_ForenameTextBox.Text,
212         AddNewCriminal_CodenameTextBox.Text,
213         AddNewCriminal_IsHeightKnownCheckBox.IsChecked.Value ? (int?)
214         AddNewCriminal_HeightSlider.Value : null,
215         (EyeColor)AddNewCriminal_EyeColorComboBox.SelectedItem,
216         (HairColor)AddNewCriminal_HairColorComboBox.SelectedItem,
217         (SexOptions)AddNewCriminal_SexComboBox.SelectedItem,
218         AddNewCriminal_SpecialSignsTextBox.Text,
219         (Country)AddNewCriminal_CitizenshipComboBox.SelectedItem,
220         (Country)AddNewCriminal_BirthCountryComboBox.SelectedItem,
221         AddNewCriminal_BirthplaceTextBox.Text,
222         AddNewCriminal_BirthdateDatePicker.SelectedDate,
223         (Country)
224         AddNewCriminal_LastLivingCountryComboBox.SelectedItem,
225         AddNewCriminal_LastLivingPlaceTextBox.Text,
226         languages,
227         (CriminalStateOptions)
228         AddNewCriminal_CurrentStateComboBox.SelectedItem,
229         null,
230         (AddNewCriminal_CriminalGroupComboBox.SelectedIndex == -1)
231         ? null
232         : Database.CriminalGroups[((KeyValuePair<int,
233         CriminalGroup>
234         AddNewCriminal_CriminalGroupComboBox.SelectedItem).Key],
235         charges
236     ));

```

```
229
230         if (AddNewCriminal_PhotoFilePath.Text != "")
231             Database.ChangeCriminalsPhoto
                (AddNewCriminal_PhotoFilePath.Text, Criminal.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
246     /// </summary>
247     private void CriminalActions_AddCriminalButton_OnClick(object sender,
        RoutedEventArgs e)
248     {
249         ClearAddCriminalFlyout();
250         AddNewCriminalFlyout.IsOpen = true;
251     }
252
253     /// <summary>
254     /// Метод очистки полей AddCriminalFlyout от данных предыдущих
        использований
255     /// </summary>
256     private void ClearAddCriminalFlyout()
257     {
258         AddNewCriminal_BirthdateDatePicker.SelectedDate = null;
259         AddNewCriminal_LanguagesListBox.UnselectAll();
260         AddNewCriminal_BirthCountryComboBox.SelectedIndex = 0;
261         AddNewCriminal_BirthplaceTextBox.Text = "";
262         AddNewCriminal_ChargesListBox.UnselectAll();
263         AddNewCriminal_CitizenshipComboBox.SelectedIndex = 0;
264         AddNewCriminal_CodenameTextBox.Text = "";
265         AddNewCriminal_CriminalGroupComboBox.SelectedIndex = -1;
266         AddNewCriminal_CurrentStateComboBox.SelectedIndex = 0;
267         AddNewCriminal_EyeColorComboBox.SelectedIndex = 0;
268         AddNewCriminal_HairColorComboBox.SelectedIndex = 0;
269         AddNewCriminal_ForenameTextBox.Text = "";
270         AddNewCriminal_LastnameTextBox.Text = "";
271         AddNewCriminal_SpecialSignsTextBox.Text = "";
272         AddNewCriminal_LastLivingCountryComboBox.SelectedIndex = 0;
273         AddNewCriminal_LastLivingPlaceTextBox.Text = "";
274         AddNewCriminal_SexComboBox.SelectedIndex = 0;
275         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
281     /// </summary>
282     private void Flyout_OnIsOpenChanged(object sender, RoutedEventArgs e)
283         => MainGrid.IsEnabled = !MainGrid.IsEnabled;
284
285     /// <summary>
286     /// Обработчик события Click для CriminalActions_DeleteCriminalButton
287     /// Спрашивает разрешение на удаление преступника и в случае ↗
        /// подтверждения удаляет выбранного преступника
288     /// </summary>
289     private void CriminalActions_DeleteCriminalButton_OnClick(object ↗
        sender, RoutedEventArgs e)
290     {
291         if (!AskPermissionToDelete()) return;
292         Database.DeleteCriminal(((CriminalsListBoxItemData) ↗
        CriminalsListBox.SelectedItem).Id);
293         Reload_CriminalsListBox();
294     }
295
296     /// <summary>
297     /// Обработчик события SelectionChanged для CriminalsListBox
298     /// Блокирует/разблокирует кнопки открытия профиля и удаления ↗
        /// преступника
299     /// </summary>
300     private void CriminalsListBox_OnSelectionChanged(object sender, ↗
        SelectionChangedEventArgs e)
301     {
302         CriminalActions_DeleteCriminalButton.IsEnabled = ↗
        CriminalsListBox.SelectedItems.Count != 0;
303         CriminalActions_OpenProfileButton.IsEnabled = ↗
        CriminalsListBox.SelectedItems.Count != 0;
304     }
305
306     /// <summary>
307     /// Обработчик событий изменения данных для всех элементов управления ↗
        /// фильтра
308     /// Перезагружает список преступников при изменении значений полей ↗
        /// фильтра
309     /// </summary>
310     private void Filter_ControlValueChanged(object sender, object e) => ↗
        Reload_CriminalsListBox();
311
312     /// <summary>
313     /// Обработчик события Click для Filter_ShowAllCriminals
314     /// Сбрасывает выбор группировки
315     /// </summary>
316     private void Filter_ShowAllCriminals_OnClick(object sender, ↗
        RoutedEventArgs e)
317         => CriminalGroupsListBox.UnselectAll();
318
319     /// <summary>
320     /// Обработчик события SelectionChanged для CriminalGroupsListBox
321     /// Отображает/скрывает данные группировке и разблокирует/блокирует ↗
        /// кнопку удаления
322     /// </summary>
```

```
323     private void CriminalGroupsListBox_OnSelectionChanged(object sender,
324         SelectionChangedEventArgs e)
325     {
326         if (CriminalGroupsListBox.SelectedIndex == -1)
327         {
328             CriminalGroupInformationGrid.Visibility =
329                 Visibility.Collapsed;
330             CriminalGroupInformationGrid_Name.Content = "";
331             CriminalGroupInformationGrid_Information.Text = "";
332         }
333         else
334         {
335             var selectedCriminalGroup = ((KeyValuePair<int,
336                 CriminalGroup>)CriminalGroupsListBox.SelectedItem).Value;
337             CriminalGroupInformationGrid.Visibility = Visibility.Visible;
338             CriminalGroupInformationGrid_Name.Content =
339                 selectedCriminalGroup.Name;
340             CriminalGroupInformationGrid_Information.Text =
341                 selectedCriminalGroup.AdditionalData;
342         }
343         CriminalGroupsListBox_DeleteGroupButton.IsEnabled =
344             CriminalGroupsListBox.SelectedIndex != -1;
345         Reload_CriminalsListBox();
346     }
347
348     /// <summary>
349     /// Обработчик события Click для CriminalGroupsListBox_AddGroupButton
350     /// Очищает и открывает Flyout для добавления новой группировки
351     /// </summary>
352     private void CriminalGroupsListBox_AddGroupButton_OnClick(object
353         sender, RoutedEventArgs e)
354     {
355         ClearAddNewCriminalGroupFlyout();
356         AddNewCriminalGroupFlyout.IsOpen = true;
357     }
358
359     /// <summary>
360     /// Очищает значения элементов управления внутри
361     /// AddNewCriminalGroupFlyout
362     /// </summary>
363     private void ClearAddNewCriminalGroupFlyout()
364     {
365         AddNewCriminalGroup_NameTextBox.Text = "";
366         AddNewCriminalGroup_AdditionalDataTextBox.Text = "";
367     }
368
369     /// <summary>
370     /// Обработчик события Click для AddNewCriminalGroup_SubmitButton
371     /// Проверяет заполненные пользователем поля при добавлении
372     /// группировки
373     /// В случае успешного заполнения создает новую группировку
374     /// </summary>
375     private void AddNewCriminalGroup_SubmitButton_OnClick(object sender,
376         RoutedEventArgs e)
377     {
378         if (AddNewCriminalGroup_NameTextBox.Text == "")
```

```
369     {
370         AddNewCriminalGroup_NameTextBox.Text = "Required";
371         return;
372     }
373
374     Database.AddCriminalGroup(new CriminalGroup
375         (AddNewCriminalGroup_NameTextBox.Text,
376         AddNewCriminalGroup_AdditionalDataTextBox.Text));
377     AddNewCriminalGroupFlyout.IsOpen = false;
378     Reload_CriminalGroups();
379 }
380
381 /// <summary>
382 /// Обработчик события Click для
383 /// CriminalGroupsListBox_DeleteGroupButton
384 /// Удаляет выбранную группировку
385 /// </summary>
386 private void CriminalGroupsListBox_DeleteGroupButton_OnClick(object
387     sender, RoutedEventArgs e)
388 {
389     if (!AskPermissionToDelete()) return;
390     Database.DeleteCriminalGroup(((KeyValuePair<int, CriminalGroup>)
391         CriminalGroupsListBox.SelectedItem).Key);
392     Reload_CriminalGroups();
393 }
394
395 /// <summary>
396 /// Обработчик события Click для CriminalActions_OpenProfileButton
397 /// Открывает профиль выбранного преступника
398 /// </summary>
399 private void CriminalActions_OpenProfileButton_OnClick(object sender,
400     RoutedEventArgs e)
401 {
402     CriminalProfile criminalProfile = new CriminalProfile
403         (Database.Criminals[(((CriminalsListBoxItemData)
404             CriminalsListBox.SelectedItem).Id)];
405     criminalProfile.ShowDialog();
406     Reload_CriminalsListBox();
407 }
408
409 /// <summary>
410 /// Обработчик события Click для
411 /// CriminalGroupInformationGrid_ChangeDataButton
412 /// Открывает Flyout для изменения данных о группировке
413 /// </summary>
414 private void CriminalGroupInformationGrid_ChangeDataButton_OnClick
415     (object sender, RoutedEventArgs e)
416 {
417     CriminalGroup selectedCriminalGroup = ((KeyValuePair<int,
418         CriminalGroup>)CriminalGroupsListBox.SelectedItem).Value;
419     ChangeCriminalGroupFlyout_NameTextBox.Text =
420         selectedCriminalGroup.Name;
421     ChangeCriminalGroupFlyout_AdditionalDataTextBox.Text =
422         selectedCriminalGroup.AdditionalData;
423     ChangeCriminalGroupFlyout.IsOpen = true;
```

```
412     }
413
414     /// <summary>
415     /// Обработчик события Click для
416     ChangeCriminalGroupFlyout_SubmitButton
417     /// Подтверждает внесенные в информацию о группировке изменения
418     /// </summary>
419     private void ChangeCriminalGroupFlyout_SubmitButton_OnClick(object sender, RoutedEventArgs e)
420     {
421         if (ChangeCriminalGroupFlyout_NameTextBox.Text == "")
422         {
423             ChangeCriminalGroupFlyout_NameTextBox.Text = "Required";
424             return;
425         }
426         CriminalGroup selectedCriminalGroup = ((KeyValuePair<int, CriminalGroup>)CriminalGroupsListBox.SelectedItem).Value;
427         selectedCriminalGroup.Name =
428             ChangeCriminalGroupFlyout_NameTextBox.Text;
429         selectedCriminalGroup.AdditionalData =
430             ChangeCriminalGroupFlyout_AdditionalDataTextBox.Text;
431         ChangeCriminalGroupFlyout.IsOpen = false;
432         Reload_CriminalGroups();
433         CriminalGroupInformationGrid_Name.Content =
434             selectedCriminalGroup.Name;
435         CriminalGroupInformationGrid_Information.Text =
436             selectedCriminalGroup.AdditionalData;
437     }
438
439     /// <summary>
440     /// Обработчик события Click для RightWindowCommands_AboutButton
441     /// Отображает информацию о данной программе
442     /// </summary>
443     private void RightWindowCommands_AboutButton_OnClick(object sender, RoutedEventArgs e)
444     {
445         this.ShowMessageAsync("About this program",
446             "Курсовой проект по ООП студента группы ПИ-15-1 ХНУРЭ
447             Слупского Данилы\n\n" +
448             "Картотека Интерпола.\n\n" + "Данные по каждому
449             зарегистрированному преступнику: фамилия, имя, кличка, рост,
450             цвет волос, цвет глаз, особые приметы, гражданство, место и
451             дата рождения, последнее место жительства, знание языков,
452             преступная профессия, последнее дело и так далее. Преступные
453             и мафиозные группировки(данные о подельщиках). Выборка по
454             любому подмножеству признаков. Перенос «завязавших» в архив;
455             удаление –только после смерти.",
456             MessageDialogStyle.Affirmative);
457     }
458
459     /// <summary>
460     /// Обработчик события Click для RightWindowCommands_SaveButton
461     /// Сохраняет текущее состояние базы в файл
462     /// </summary>
463     private void RightWindowCommands_SaveButton_OnClick(object sender,
```

```

        RoutedEventArgs e)
            => Database.SaveData();
452
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        Database.RestoreData();
461        Reload_CriminalsListBox();
462        Reload_CriminalGroups();
463    }
464
465    /// <summary>
466    /// Требуется от пользователя разрешение на удаление данных
467    /// </summary>
468    /// <returns>Имеет ли пользователь разрешение на удаление данных</returns>
469    private bool AskPermissionToDelete()
470    {
471        DeleteDialog dialog = new DeleteDialog();
472        dialog.ShowDialog();
473        return dialog.Result;
474    }
475    #endregion
476 }
477
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            Id = id;
498            TotalName = "#" + id + " - " + forename + " \'" + codename + "\' " + lastname + " AGE: " + ((age == null) ? "-" : age.ToString());
499            Citizenship = citizenship;
500            if (!File.Exists("..\..\Storage\Files\" + imageName))
501                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 фильтров</param>
564        /// <param name="selectedItems">Выбранные значения</param>
565        /// <returns>Отфильтрованная коллекция преступников</returns>
566        public static List<Criminal> FilterByState(List<Criminal> result, 7
            IList selectedItems)
567        {
568            for (int i = 0; i < result.Count; i++)
569            {
570                if (selectedItems.Cast<CriminalStateOptions>()
571                    .Any(selectedItem => result[i].State == selectedItem)) 7
                    continue;
572                result.RemoveAt(i);
573                i--;
574            }
575            return result;
576        }
577
578        /// <summary>
579        /// Фильтрация по имени преступника
580        /// </summary>
581        /// <param name="result">Результат предыдущих фильтров</param>
582        /// <param name="text">Выбранное значение</param>
583        /// <returns>Отфильтрованная коллекция преступников</returns>
584        public static List<Criminal> FilterByForename(List<Criminal> result, 7
            string text)
585            => result.Where(r => r.Forename.ToLower().Contains(text.ToLower 7
                ())).ToList();
586
587        /// <summary>
588        /// Фильтрация по кличке преступника
589        /// </summary>
590        /// <param name="result">Результат предыдущих фильтров</param>
591        /// <param name="text">Выбранное значение</param>
592        /// <returns>Отфильтрованная коллекция преступников</returns>
593        public static List<Criminal> FilterByCodename(List<Criminal> result, 7
            string text)
594            => result.Where(r => r.CodeName.ToLower().Contains(text.ToLower 7
                ())).ToList();
595
596        /// <summary>
597        /// Фильтрация по фамилии преступника
598        /// </summary>
599        /// <param name="result">Результат предыдущих фильтров</param>
600        /// <param name="text">Выбранное значение</param>
601        /// <returns>Отфильтрованная коллекция преступников</returns>
602        public static List<Criminal> FilterByLastname(List<Criminal> result, 7
            string text)
603            => result.Where(r => r.Lastname.ToLower().Contains(text.ToLower 7
                ())).ToList();
```

```
604
605     /// <summary>
606     /// Фильтрация по диапазону возрастов преступника (если возраст известен)
607     /// </summary>
608     /// <param name="result">Результат предыдущих фильтров</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<Criminal> FilterByAge(List<Criminal> 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 фильтров</param>
619     /// <returns>Отфильтрованная коллекция преступников</returns>
620     public static List<Criminal> FilterByAge(List<Criminal> result)
621     => result.Where(r => r.Age == null).ToList();
622
623     /// <summary>
624     /// Фильтрация по диапазону роста преступника (если рост известен)
625     /// </summary>
626     /// <param name="result">Результат предыдущих фильтров</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<Criminal> FilterByHeight(List<Criminal> result, int lowerValue, int upperValue)
631     => result.Where(r => r.Height >= lowerValue && r.Height <= upperValue).ToList();
632
633     /// <summary>
634     /// Фильтрация по диапазону роста преступника (если рост неизвестен)
635     /// </summary>
636     /// <param name="result">Результат предыдущих фильтров</param>
637     /// <returns>Отфильтрованная коллекция преступников</returns>
638     public static List<Criminal> FilterByHeight(List<Criminal> 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 фильтров</param>
645     /// <param name="selectedValue">Выбранное значение</param>
646     /// <returns>Отфильтрованная коллекция преступников</returns>
647     public static List<Criminal> FilterByEyeColor(List<Criminal> 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 фильтров</param>
654     /// <param name="selectedValue">Выбранное значение</param>
655     /// <returns>Отфильтрованная коллекция преступников</returns>
656     public static List<Criminal> FilterByHairColor(List<Criminal> 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 фильтров</param>
663     /// <param name="selectedValue">Выбранное значение</param>
664     /// <returns>Отфильтрованная коллекция преступников</returns>
665     public static List<Criminal> FilterBySex(List<Criminal> 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 фильтров</param>
672     /// <param name="text">Выбранное значение</param>
673     /// <returns>Отфильтрованная коллекция преступников</returns>
674     public static List<Criminal> FilterBySpecialSigns(List<Criminal> ↗
        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 фильтров</param>
681     /// <param name="selectedItems">Выбранные значения</param>
682     /// <returns>Отфильтрованная коллекция преступников</returns>
683     public static List<Criminal> FilterByLanguages(List<Criminal> 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 фильтров</param>
699     /// <param name="selectedValue">Выбранное значение</param>
700     /// <returns>Отфильтрованная коллекция преступников</returns>
701     public static List<Criminal> FilterByCitizenship(List<Criminal> ↗
        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 фильтров</param>
708         /// <param name="selectedValue">Выбранное значение</param>
709         /// <returns>Отфильтрованная коллекция преступников</returns>
710         public static List<Criminal> FilterByBirthCountry(List<Criminal> result, Country selectedValue) =>
711             result.Where(r => r.BirthCountry == selectedValue).ToList();
712
713         /// <summary>
714         /// Фильтрация по месту рождения преступника
715         /// </summary>
716         /// <param name="result">Результат предыдущих фильтров</param>
717         /// <param name="text">Выбранное значение</param>
718         /// <returns>Отфильтрованная коллекция преступников</returns>
719         public static List<Criminal> FilterByBirthPlace(List<Criminal> result, string text) =>
720             result.Where(r => r.Birthplace.ToLower().Contains(text.ToLower())).ToList();
721
722         /// <summary>
723         /// Фильтрация по последней зарегистрированной стране проживания преступника
724         /// </summary>
725         /// <param name="result">Результат предыдущих фильтров</param>
726         /// <param name="selectedValue">Выбранное значение</param>
727         /// <returns>Отфильтрованная коллекция преступников</returns>
728         public static List<Criminal> FilterByLastLivingCountry(List<Criminal> result, Country selectedValue) =>
729             result.Where(r => r.LastLivingCountry == selectedValue).ToList();
730
731         /// <summary>
732         /// Фильтрация по последнему зарегистрированному месту проживания преступника
733         /// </summary>
734         /// <param name="result">Результат предыдущих фильтров</param>
735         /// <param name="text">Выбранное значение</param>
736         /// <returns>Отфильтрованная коллекция преступников</returns>
737         public static List<Criminal> FilterByLastLivingPlace(List<Criminal> result, string text) =>
738             result.Where(r => r.LastLivingPlace.ToLower().Contains(text.ToLower())).ToList();
739     }
740 }
```