Министерство образования и науки Украины

Харьковский национальный университет радиоэлектроники

Кафедра Программной инженерии

КУРСОВАЯ РАБОТА

ОБЪЯСНИТЕЛЬНАЯ ЗАПИСКА

По дисциплине “Объектно-ориентированное программирование”

«Генеалогическое дерево»

Руководитель: Бондарев В. М.

Студент гр. ПЗПИ-19-2 Куприянов М.А

Комиссия:

Ст. Преп. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Черепанова Ю. Ю.

Проф. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Бондарев В. М.

Доц. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Побеженко И. А.

Харьков 2020

**ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ РАДИОЭЛЕКТРОНИКИ**

Кафедра: ***Программной инженерии***

Дисциплина: ***Объектно-ориентированное программирование***

Специальность: ***121 Инженерия программного обеспечения***

Образовательная программа: ***Программная инженерия***

Курс 1 . Группа ПЗПИ-19-2 . Семестр 2 .

***ЗАДАНИЕ***

***на курсовой проект студента***

***Куприянова Михаила Александровича***

1. Тема проекта: *«Генеалогическое дерево»*
2. Термин сдачи студентом законченного проекта: ***“31” – мая – 2020 г.***
3. Исходные данные к проекту:

*Паспортные данные членов некоторого родового клана; ссылки на детей (или родителей). Поиск всех потомков или всех предков для конкретного члена клана. Иерархическое отображение генеалогического дерева выбранной человека.*

1. Содержание расчетно-пояснительной записки:

*Реферат, содержание, вступление, спецификация программы, проектная спецификация, инструкция пользователя, тестирование данных, выводы.*

1. Перечень графического материала:

*Рисунки, примеры экранных форм.*

**КАЛЕНДАРНЫЙ ПЛАН**

|  |  |  |
| --- | --- | --- |
| *№* | *Название этапа* | *Сроки выполнения* |
| 1 | Выдача, согласование и утверждение темы. | 30-03-2020 р. |
| 2 | Составление спецификации | 31-03-2020 – 06-04-2020 г. |
| 3 | Проектирование | 07-04-2020 – 13-04-2020 г. |
| 4 | Программирование модели | 14-04-2020 – 06-05-2020 г. |
| 5 | Программирование пользовательского интерфейса | 07-05-2020 – 20-05-2020 г. |
| 6 | Тестирование и работа над деталями | 21-05-2020 – 26-05-2020 г. |
| 7 | Написание пояснительной записки | 27-05-2020 – 31-05-2020 г. |
| 8 | Защита | 01-06-2020 – 01-07-2020 г. |

Студент Куприянов Михаил Александрович

Руководитель Бондарев Владимир Михайлович

*« 30 » марта 2020 года*

**РЕФЕРАТ**

Пояснительная записка: 28 с, 45 рис, 1 приложение, 5 источников.

ВЕДОМОСТЬ, КЛАСС, ЯЗЫК ПРОГРАММИРОВАНИЯ C#, НАКЛАДНАЯ, ОБЬЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ, ПРОГРАММА, ПЛАТФОРМА .NET, ГЕНЕАЛОГИЧЕСКОЕ ДЕРЕВО.

Целью данной работы является разработка программы “Генеалогическое дерево” на основе объектно-ориентированного подхода.

При реализации курсового проекта был использован язык С#, среда разработки Microsoft Visual Studio 2019 на платформе .NET Framework, 4.7.2 тип приложения: Windows Forms.

В результате разработки была создана программа под названием «Генеалогическое дерево», которая позволяет хранить информацию о членах некоторого родового клана: паспортные данные, имена, фамилии, отчества; искать всех потомков и предков для конкретного человека. Пользователю доступен большой спектр инструментов для работы с определенным родовым кланом.

СОДЕРЖАНИЕ

Вступление 6

[1](#_Toc260271314) Спецификация программы 7

[1.1 Описание функций программ](#_Toc260271315)ы 7

[1.2 Интерфейс](#_Toc260271316) пользователя 7

[2 Проектная](#_Toc260271318) спецификация 14

[2.1 Объектная модель программ](#_Toc260271319)ы 14

[2.2 Реализация функций программ](#_Toc260271320)ы 17

2.3 Формат данных 19

[3 Инструкция](#_Toc260271324) пользователю 21

[3.1 Установка программ](#_Toc260271320)ы 21

3.2 Работа с программой 21

Вывод 26

[Источники 27](#_Toc260271326)

Приложение А: Ссылка на код программы 28

**ВСТУПЛЕНИЕ**

В данном курсовом проекте разрабатывалось оконное приложение, основанное на принципах объектно-ориентированного программирования.

Темой курсового проекта является “Генеалогическое дерево”, программа предоставляет доступ к базе членов некого родового клана в удобном формате. Пользователь может обращаться к данным, а также взаимодействовать с ними: добавлять, удалять, редактировать данные в базе, и кроме того осуществлять по ним поиск среди элементов базы.

Программа, разработанная в данной курсовой работе, облегчает процесс хранения информации о семейном клана и дает возможность пользователям полноценно работать с данными. Программа предоставляет свободный и комфортный доступ к информации, позволяет полностью адаптировать содержание базы под собственные нужды. Пользователь может редактировать любой из видов информации о членах клана.

Приложение было разработано с учетом всех аспектов создания оконных приложений на языке C # и разработки в среде .NET [1-5]. Разработка велась по четко составленному плану.

Использование объектно-ориентированного подхода при создании программы такого вида упрощает ее разработку и позволяет в дальнейшем совершенствовать приложение путем минимальных вмешательств в исходный код [1].

1. **СПЕЦИФИКАЦИЯ ПРОГРАММЫ**

1.1 Описание функций программы:

Программа «Генеалогическое дерево» предоставляет пользователю возможность структурированно хранить и просматривать информацию о некотором родовом клане, просматривать генеалогическое дерево для конкретного человека, редактировать внесенные данные.

Среди основных функций программного обеспечения следует выделить следующие:

* Просмотр всех членов клана
* Добавление нового человека в базу
* Редактирование данных
* Удаление конкретного человека
* Отображение генеалогического дерева для конкретного человека
* Создание нового клана
* Просмотр данных про конкретный клан
* Редактирование конкретного клана

1.2 Описание пользовательского интерфейса программы

При включении программы, отображается форма, в которой можно выбрать что пользователь желает сделать в первую очередь (см. Рис. 1.1).

Меню осуществляет доступ к таким функциям программы, как: «Открыть», «Создать», «Сохранить» и «Закрыть» (см. Рис. 1.2).

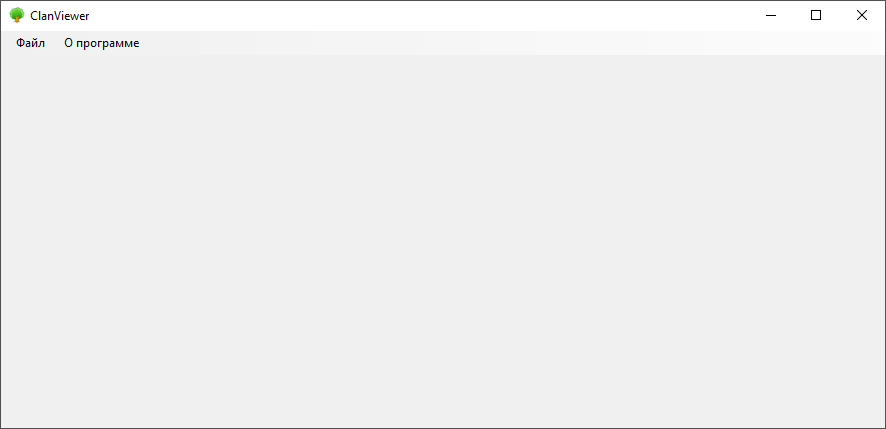


Рисунок 1.1 – Стартовая форма

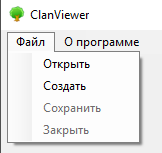


Рисунок 1.2 – Выбор последующих действий: открыть уже существующий клан или создать новый

Если у вас уже есть созданный семейный клан, необходимо нажать кнопку «Открыть» и найти сохраненный файл с данными.

В противном случае выбираем кнопку «Создать», после чего необходимо ввести название клана и пароль, зная который можно будет потом редактировать данные (см. Рис. 1.3).

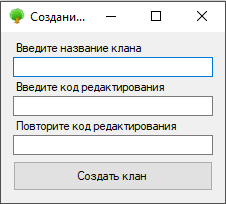
Когда введены необходимые исходные данные, необходимо нажать на кнопку «Создать клан», тогда открывается главная форма (см. Рис. 1.4).

Рисунок 1.3 – Форма для создания нового клана

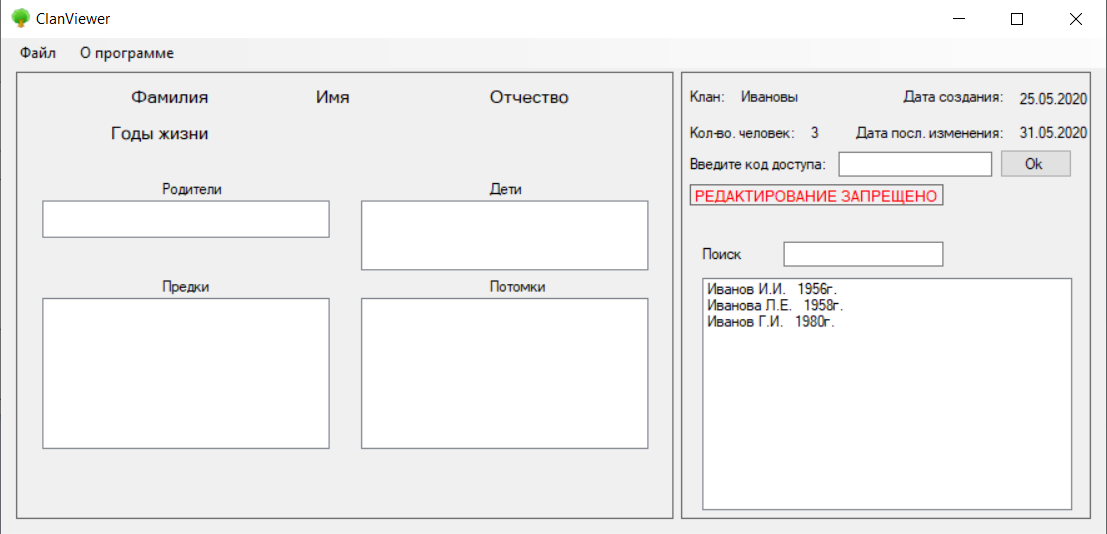


Рисунок 1.4 – Главная форма

В главной форме вы можете просмотреть генеалогическое дерево любого человека, который уже внесена в базу данных. Для этого необходимо в правой части формы найти окно, где находится перечень людей данного родового клана, после чего необходимо дважды нажать на нужного человека и левая часть формы сразу заполнится необходимыми данными (см. Рис. 1.5).

Для редактирования любого члена семьи нужно ввести код доступа, тогда надпись «Редактирование запрещено» изменится на «Редактирование разрешено». Теперь необходимо дважды нажать на нужную лицо и в нижней части формы появятся кнопки «Редактировать» и «Удалить» (см. Рис. 1.6). Если нажать кнопку «Удалить» данные о человеке исчезнут из родового клана. После нажатия кнопки «Редактировать» появится новая форма. В этой форме вы увидите поля для редактирования данных и кнопку «Сохранить» (см. Рис. 1.7)

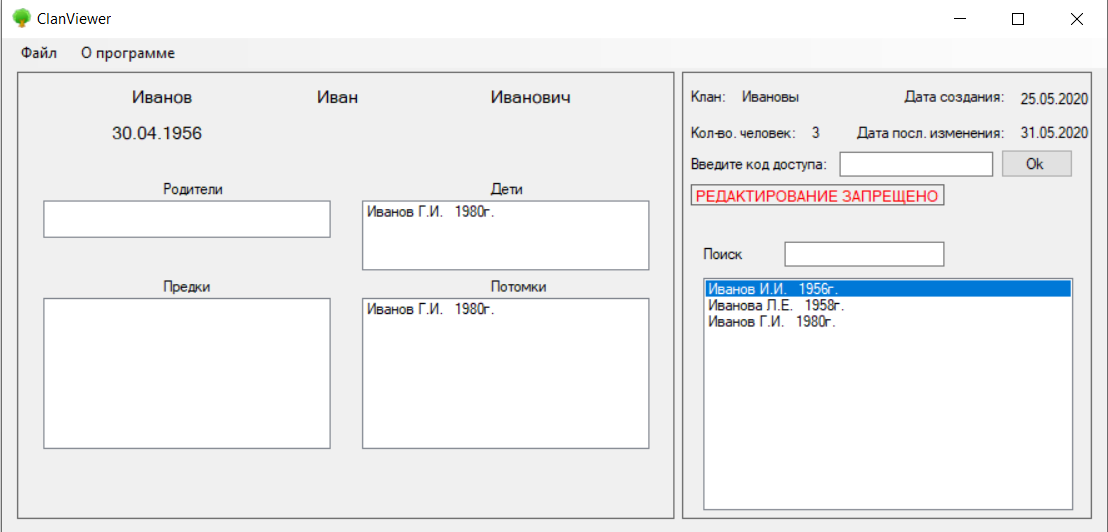


Рисунок 1.5 – Форма просмотра данных о человеке

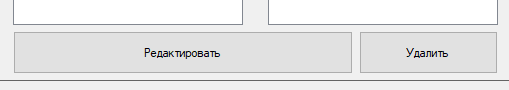


Рисунок 1.6 – Клавиши редактирования и удаления

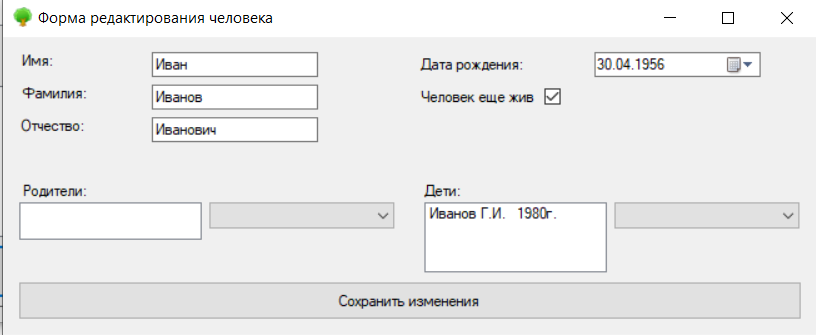


Рисунок 1.7 – Форма редактирования

Поиск нужного человека осуществляется путем ввода данных в поле "Поиск". В этом поле можно вводить как год рождения, так и фамилию или инициалы (см. Рис. 1.8).

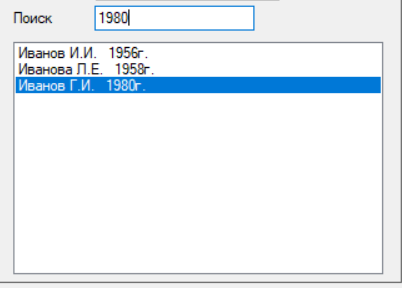


Рисунок 1.8 – Поиск

Код доступа позволяет добавлять новых людей к родовому клану, для этого необходимо нажать кнопку «Добавить человека» (см. Рис. 1.9).

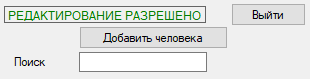


Рисунок 1.9 – Демонстрация кнопки «Добавить человека»

Нажатие этой кнопки открывает форму добавления человека к родовому клану, которая выглядит почти как форма редактирования (см. Рис. 1.10).

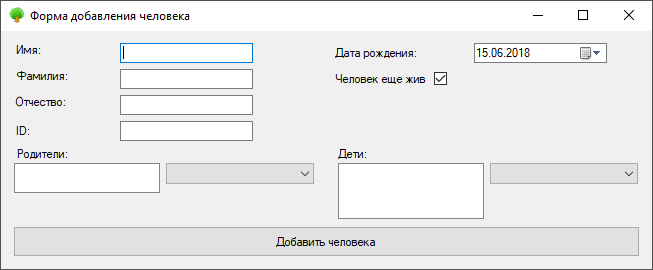


Рисунок 1.10 – «Форма добавления человека»

После ввода имени, фамилии, отчества, ID и даты рождения можно выбрать родителей и детей, если таковые имеются (см. Рис. 1.11).

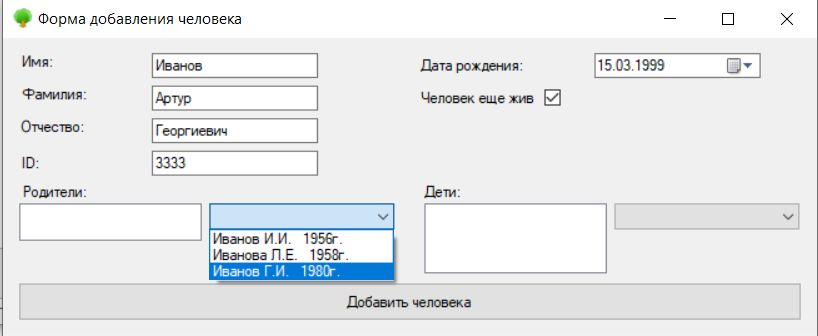
В системе предусмотрено, что родители не могут быть моложе детей, поэтому выбирать родителей можно только из тех людей, кто старше создаваемого человека, а детей только из тех, кто моложе. ID может состоять только из цифр, а имя, фамилия и отчество - с букв. Если человек уже умер, нужно убрать галочку с поля «Человек еще жив», тогда можно будет выбрать дату смерти.

Рисунок 1.11 – Выбор родителей и детей

**2 ПРОЕКТНА СПЕЦИФИКАЦИЯ**

2.1 Объектная модель программы

Программа представляет из себя набор классов, которые взаимодействуют между собой (см. рис. 2.1).

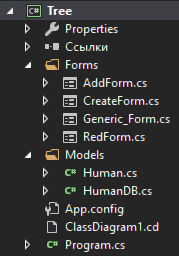


Рисунок 2.1 – Архитектура проекта

Папка «Forms» содержит в себе все формы, которые используются в проекте. [3]

На рисунке 2.2 изображены поля, которые реализуют всю необходимую информацию о людях:

1. Имя
2. Фамилия
3. Отчество
4. Дата рождения
5. Жизнь
6. Дата смерти

public class Human

{

public string Name { get; set; }

public string SName { get; set; }

public string FName { get; set; }

public DateTime BornDate{ get; set; }

public bool Life{ get; set; }

public DateTime DeathDate{ get; set; }

}

Рисунок 2.2 – Поля класса Human

Кроме этого класс содержит также множество методов, направленных на сохранение информации о человеке, загрузку ее из базы, редактирование, удаление и прочее.

На рисунке 2.3 показано, как объекты классов взаимодействуют между собой.

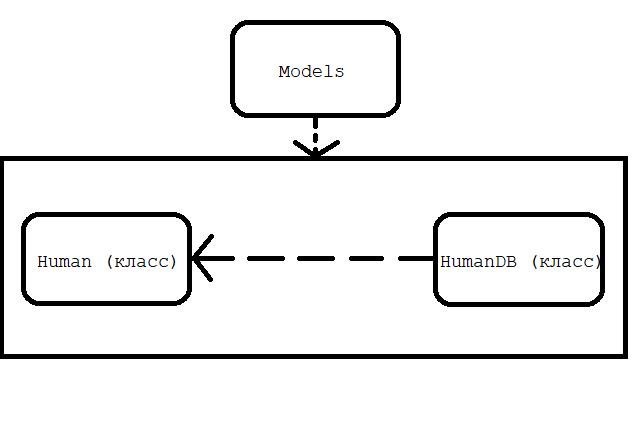


Рисунок 2.3 – Иллюстрация взаимодействия классов в программе.

Класс HumanDB (див. рис. 2.4) является основним класом для сохранения и загрузки информации из файлов, заполнения таблицы информацией.

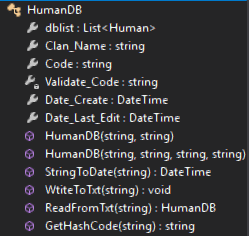


Рисунок 2.4 – Класс HumanDB и его методы

Класс«Human»представлен на рисунке 2.5.

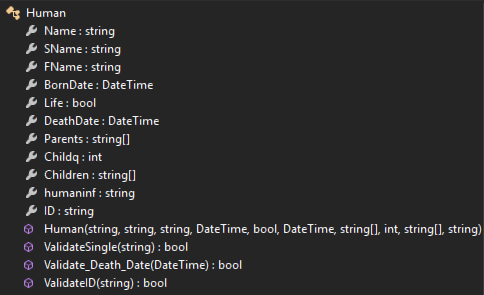


Рисунок 2.5 – Класс Human

2.2 Реализация функций программы

Рассмотрим метод поиска людей в базе данных. Данный метод реализует поиск по определенному запросу среди данных, введенных в соответствующем поле (см. Рис. 2.6).

private void SearchBox\_TextChanged(object sender, EventArgs e)

{

int index = HumanSearchByInfo(SearchBox.Text, humanlist);

if (index != -1)

AllHumansBox.SelectedIndex = index;

}

Рисунок 2.6 – Поиск среди данных, введенных в соответствующем поле

Класс «Human» - основной класс программы. У этого класcа есть конструктор «Human», который конвертирует данные людей. Класс содержит ряд свойств и полей, необходимых для роботы с элементами данного класса.

Класс «HumanDB» содержит основные методы, среди которых проверка введенных данных на валидность:

1. Данные при создании нового клана (см. Рис. 2.7).
2. Правильность введенной даты (см. Рис. 2.8).
3. Шифрование ключей доступа (см. Рис. 2.9).

public HumanDB(string clan\_name, string code)

{

if (clan\_name != "") Clan\_Name = clan\_name;

else throw new FormatException(Properties.Resources.Error\_Nothing\_ClanName);

if (code != "") Code = GetHashCode(code);

else throw new FormatException(Properties.Resources.Error\_Nothing\_Code);

Date\_Create = DateTime.Now.Date;

Date\_Last\_Edit = DateTime.Now.Date;

dblist = new List<Human>();

}

Рисунок 2.7 – Проверка на наличие информации в полях

public static DateTime StringToDate(string str)

{

try

{

int day = int.Parse(str[0].ToString() + str[1].ToString());

int mounth = int.Parse(str[3].ToString() + str[4].ToString());

int year = int.Parse(str[6].ToString() + str[7].ToString() + str[8].ToString() + str[9].ToString());

DateTime res = new DateTime(year, mounth, day);

return res;

}

catch { throw new FormatException(Properties.Resources.Error\_Validate\_Str\_Date); }

}

Рисунок 2.8 – Проверка правильности введенной даты

public static string GetHashCode(string input)

{

MD5 md5 = MD5.Create();

byte[] data = md5.ComputeHash(Encoding.UTF8.GetBytes(input));

StringBuilder sBuilder = new StringBuilder();

for (int i = 0; i < data.Length; i++)

{

sBuilder.Append(data[i].ToString("x2"));

}

return sBuilder.ToString();

}

Рисунок 2.9 – Шифрование данных

2.3 Формат данных

Запись данных в базу осуществляется следующим образом: (см. Рис. 2.10).

public void WriteToTxt(string path)

{

if (this != null)

{

string[] to\_write\_info = { Clan\_Name, Date\_Create.ToShortDateString(), Date\_Last\_Edit.ToShortDateString(), Code, Validate\_Code };

File.WriteAllLines(path, to\_write\_info);

if (dblist != null)

{

string[] to\_write\_db = new string[dblist.Count\*3];

int j = 0;

for (int i = 0; i < dblist.Count\*3; i+=3)

{

string towrite1 = "{" + dblist[j].Name + "}{" + dblist[j].SName + "}{" + dblist[j].FName + "}{"

+ dblist[j].BornDate + "}{" + dblist[j].Life + "}{" + dblist[j].DeathDate + "}{" + dblist[j].Childq +

"}{" + dblist[j].ID + "}" ;

string towrite2 = "°";

foreach (string id in dblist[j].Parents ) {

towrite2 += "{" + id + "}"; }

string towrite3 = "°";

foreach (string id in dblist[j].Children)

{

towrite3 += "{" + id + "}";

}

to\_write\_db[i] = towrite1;

to\_write\_db[i + 1] = towrite2;

to\_write\_db[i + 2] = towrite3;

j++;

}

File.AppendAllLines(path, to\_write\_db);

}

}

}

Рисунок 2.10 – Запись данных в базу

Данные хранятся в txt-файле, их можно просмотреть и отредактировать информацию о родовом клане (см. Рис. 2.11).

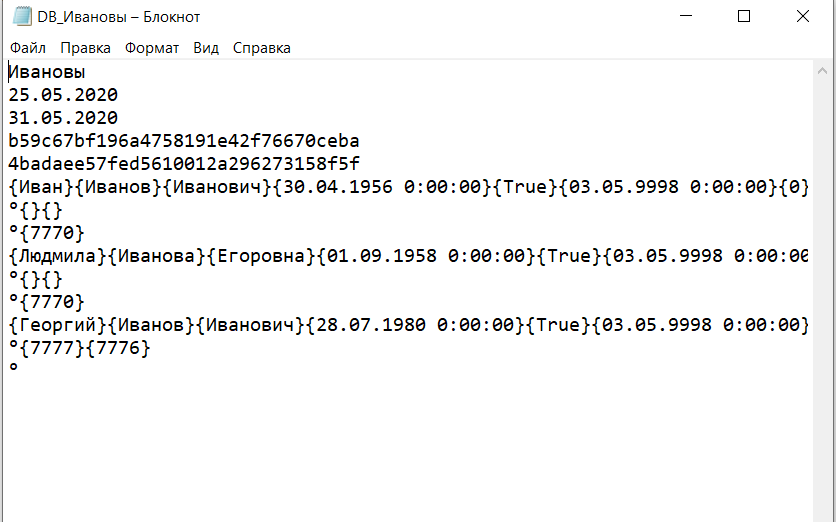


Рисунок 2.11 – Демонстрация хранения информации в файле.

## **3 РУКОВОДСТВО ДЛЯ ПОЛЬЗОВАТЕЛЯ**

* 1. Установка програми:

Установка программы происходит следующим образом:

* открываем носитель с программным продуктом;
* копируем папку «Family tree»;
* запускаем файл «Tree.exe» из корня папки для начала работы.

3.2 Работа с программой

Запускаем программу и открываем семью, созданную для примера работы программы(находится в корне папки Family Tree). Главное окно представлено на рисунке 3.1

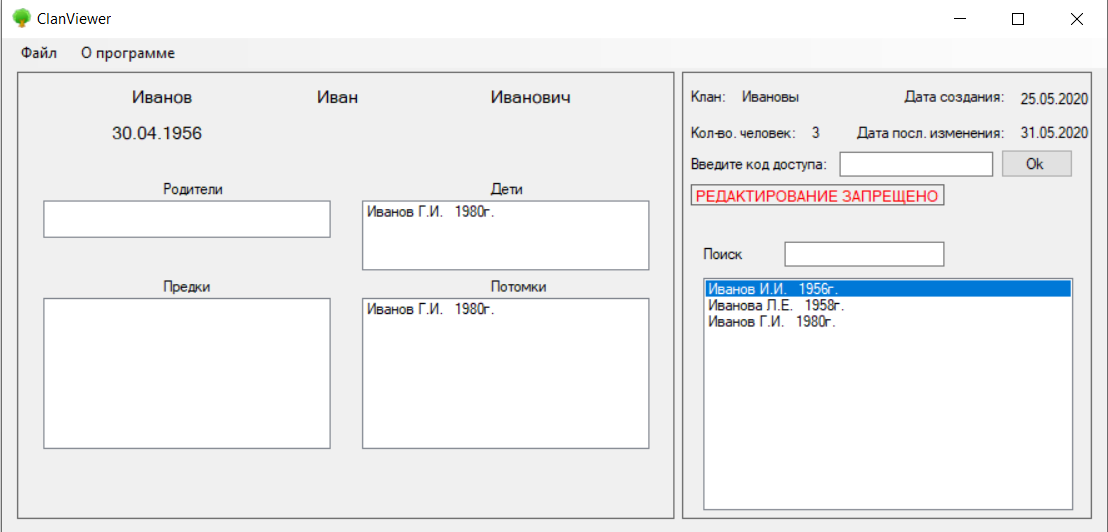


Рисунок 3.1 – Главное окно

Для доступа к редактированию информации любого из базовых родовых кланов необходимо ввести код доступу: «1111».

Чтобы перейти к просмотру генеалогического дерева интересующего нас человека, необходимо дважды нажать на его имя

Поиск:

* для того чтобы воспользоваться поиском, нужно ввести искомые данные в соответствующую строку;

Создание нового родового клана:

* необходимо выбрать в выпадающем меню «Файл» кнопку «Создать»;
* в новом окне вводим название клана и придумываем код, необходимый для редактирования информации(см. Рис. 3.2);
* откроется главное окно, где вы можете добавлять новых людей в клан

Использование уже существующего родового клана:

* необходимо выбрать в выпадающем меню «Файл» кнопку «Создать»;
* находим в файловой системе файл с данными искомого родового клана;
* нажимаем на кнопку «Открыть»;

Сохранение родового клана:

* необходимо выбрать в выпадающем меню «Файл» кнопку «Сохранить»;

Добавление нового человека в родовой клан:

* в специальном окне нужно ввести имя, фамилию, отчество и ID человека, причем первые три должны состоять исключительно из букв, а последнее исключительно из цифр
* необходимо ввести дату рождения человека, которую можно выбрать в календаре или ввести самостоятельно;
* если в базе данных уже есть родители/дети человека, которого добавляют, можно выбрать их в соответствующих полях из выпадающего списка (см. Рис. 3.3);

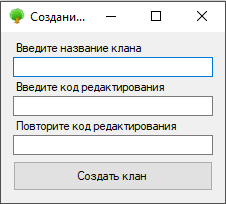


Рисунок 3.2 – Форма для создания нового клана

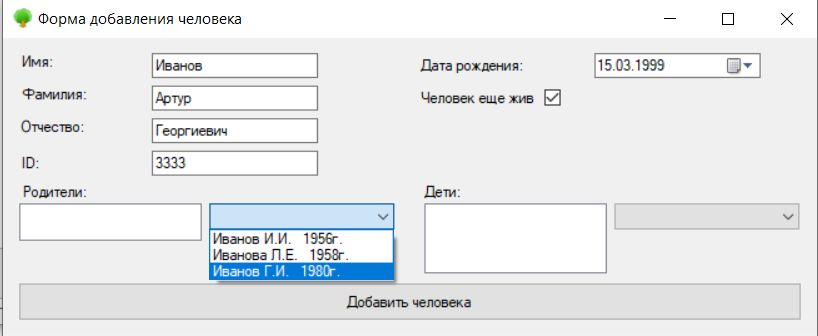


Рисунок 3.3 – Форма добавления человека

Генеалогическое дерево:

* необходимо выбрать человека, генеалогическое дерево которой хотите просмотреть
* вся информация о человеке появится в соответсвующих полях с левой части экрана

Удаление человека:

* після введення коду доступу, зліва внизу з’являються дві нові кнопки: «Редактировать» та «Удалить» (див. рис. 3.4);
* вибрати потрібну людину та натиснути «Удалить»;
* після цього людина автоматичне зникне з родового клану;

Редагування інформації про людину в програмі:

* після введення коду доступу, зліва внизу з’являються дві нові кнопки: «Редактировать» та «Удалить» (див. рис. 3.4);
* вибрати потрібну людину та натиснути «Редактировать»;
* на екрані з’явиться нова форма, де знаходиться вся інформація щодо вибраної людини, яку можна редагувати (див. рис. 3.5);

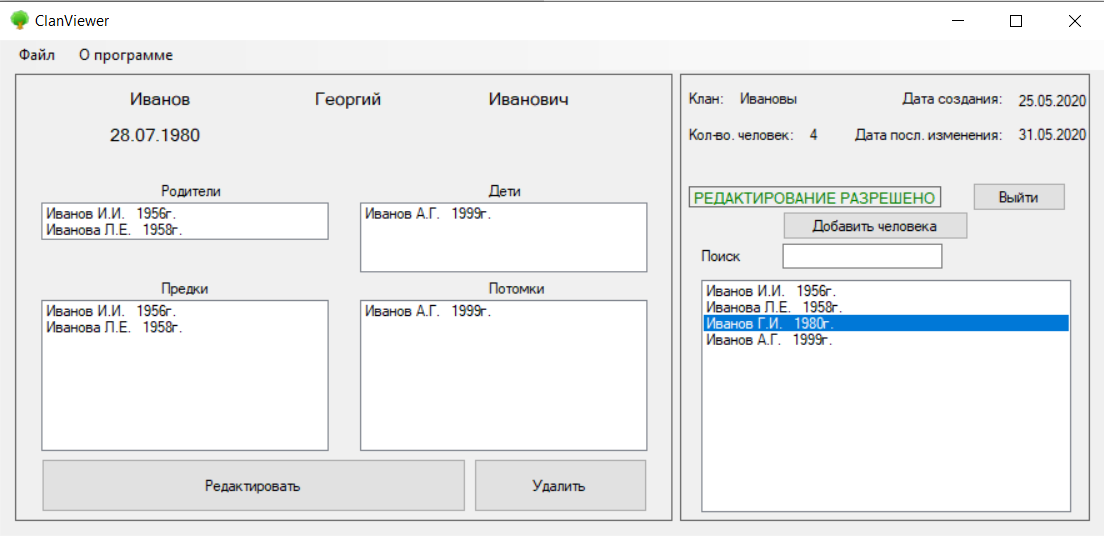


Рисунок 3.4 –Форма просмотра генеалогического дерева человека

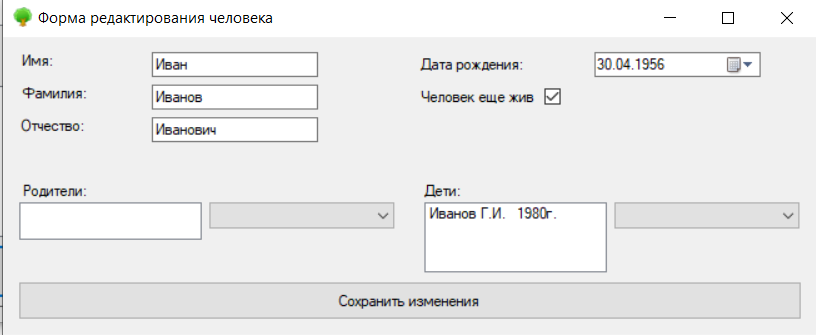


Рисунок 3.5 – Форма редактирования человека

Для завершения работы с конкретным родовым кланом:

* выбрать в выпадающем меня Файл кнопку Закрыть

Для того чтобы просмотреть информацию о программе:

* выберите кнопку О программе (см. Рис. 3.6);

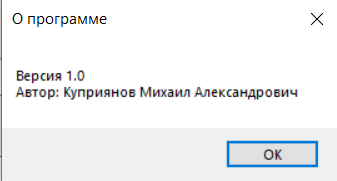


Рисунок 3.6 – Информация о программе

Для завершения работы программы можно нажать на крестик.

# ВЫВОД

В ходе выполнения курсовой работы был разработано приложение генеалогическое дерево, которая позволяет работать с списком членов некоторого родового клана, получать информацию о них, такую как: ФИО, предки, потомки, дата рождения, генеалогическое древо, изменять информацию о существующих членах клана или добавлять новых. Данное приложение может помочь людям, желающим изучить свою или чужую родословную.

В результате написания был создан полноценный программный продукт. Были изучены некоторые аспекты создания оконных приложений на языке C# в среде .NET. Были изучены отдельные фазы разработки программного обеспечения и компоновка их в единое целое. Были не только закреплены знания в дисциплине, но и получен опыт в разработке программного продукта, изучены стандарты оформления отчетной документации. Я получил опыт в разработке программ, планировании хода разработки и изучил стандарты оформления программного обеспечения.

**ИСТОЧНИКИ**

* 1. Бондарєв В. М. Об'єктно-орієнтоване програмування на С#. навч. посіб. Харків : СМІТ, 2009. 221 с.
  2. Макконнелл С. Совершенный код. Санкт-Петербург : Русская Редакция, 2007. 896 с.
  3. Троелсен Е. C# и платформа .NET. Библиотека программиста. Санкт-Петербург : Издательский дом «Уильямс», 2005. 796 с.
  4. Фролов. А. В., Фролов Г. В. Визуальное проектирование приложений C#. Москва: КУДИЦ-ОБРАЗ, 2003. 512 с.
  5. Шилдт Г. Полный справоник по С#. Моксва : Издательский дом «Уильямс», 2004. 752 с.

## **ПРИЛОЖЕНИЕ А**

<https://github.com/kupriyanov-19/Kupriianov-family-tree>