Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

ОТЧЕТ

К лабораторной работе №2 (Разработка бизнес логики приложения)

по дисциплине «Системное программное обеспечение»

Выполнил студент

группы 538:

\_\_\_\_\_\_\_\_Маматкулов Ф.С.

Проверил

доцент кафедры КСУП:

\_\_\_\_\_\_\_\_ Горяинов А.Е.

Дата: «\_\_\_»\_\_\_\_\_\_\_\_2021 г.

Томск 2021

**Оглавление**

[**1 Цели и задачи лабораторной работы** 3](#_Toc72248633)

[**2 Основная часть** 4](#_Toc72248634)

[1. UML-диаграмма классов проекта логики с пояснением назначения классов. 4](#_Toc72248635)

[2. Пример исходного кода, демонстрирующий создание и инициализацию экземпляров каждого класса программы, а также вызов их методов. 5](#_Toc72248636)

[3. Текущая история коммитов ветки develop 7](#_Toc72248637)

[**3 Заключение** 8](#_Toc72248638)

# **1 Цели и задачи лабораторной работы**

Цель работы: изучить типовые требования, предъявляемые к бизнес-логике приложения, получить умения разработки логики приложения с обеспечением данных требований.

Задачи:

* Изучить требования и процесс разработки логики приложения.
* Повторить синтаксис языка C# для разработки объектно-ориентированных программ.
* Разработать классы, необходимые для работы логики приложения.
* Обеспечить целостность данных классов с помощью свойств и механизма генерации исключений.

# **2 Основная часть**

1. UML-диаграмма классов проекта логики с пояснением назначения классов.

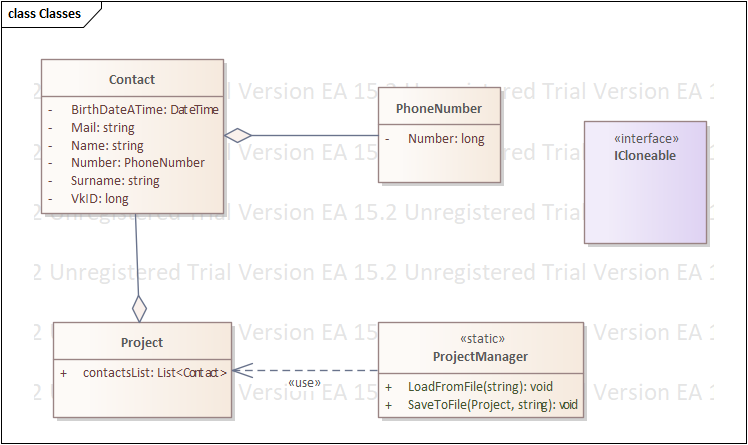


Рисунок 2.1 - UML-диаграмма классов проекта логики

Contact - Класс контакта, хранящий информацию о его имени, фамилии, номере телефона, почте, дне рождения и id вк.

PhoneNumber - Класс, хранящий номер телефона

Project - Класс, хранящий список контактов

ProjectManager - Класс, реализующий сериализацию и десиреализацию списка контактов

ICloneable – Интерфейс

1. Пример исходного кода, демонстрирующий создание и инициализацию экземпляров каждого класса программы, а также вызов их методов.

class Program

{

static void Main(string[] args)

{

Contact one = new Contact("Volkov", "Mark",new PhoneNumber(71118888899), new DateTime(2018, 03, 02), "mail", 2231);

Contact two = new Contact();

two.Surname = "Udashkin";

two.Name = "Uriy";

two.Number = new PhoneNumber(78888888899);

two.BirthDateTime = new DateTime(2018, 03, 02);

two.Mail = "Udav@mail.ru";

two.VkID = 3331;

Console.WriteLine($"Первый контакт: Surname: {one.Surname} Name: {one.Name} Number: {one.Number.Number} Mail: {one.Mail}");

Console.WriteLine($"Второй контакт: Surname: {two.Surname} Name: {two.Name} Number: {two.Number.Number} Mail: {two.Mail}");

Project prog = new Project();

Project prog2 = new Project();

prog.contactsList.Add(one);

prog.contactsList.Add(two);

Console.WriteLine($"Второй контакт из списка контактов: Surname: {prog.contactsList[1].Surname} Name: {prog.contactsList[1].Name} Number: {prog.contactsList[1].Number.Number} Mail: {prog.contactsList[1].Mail}");

ProjectManager.SaveToFile(prog, "ContactsApp.notes");

prog2 = ProjectManager.LoadFromFile("ContactsApp.notes");

Console.WriteLine($"Второй контакт из загруженного из файла списка: Surname: {prog2.contactsList[1].Surname} Name: {prog2.contactsList[1].Name} Number: {prog2.contactsList[1].Number.Number} Mail: {prog2.contactsList[1].Mail}");

Console.ReadLine();

}

}

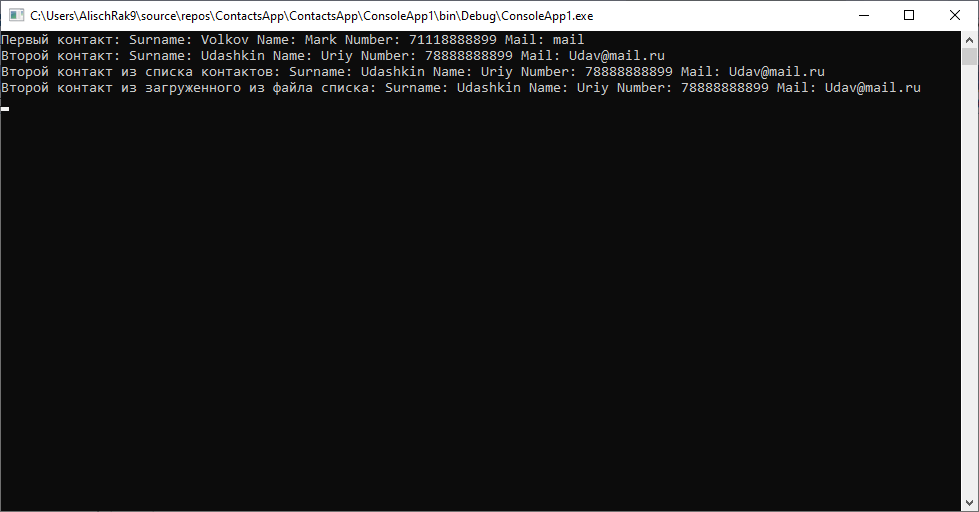


Рисунок 2.2 – Результат создания и инициализации экземпляров каждого класса

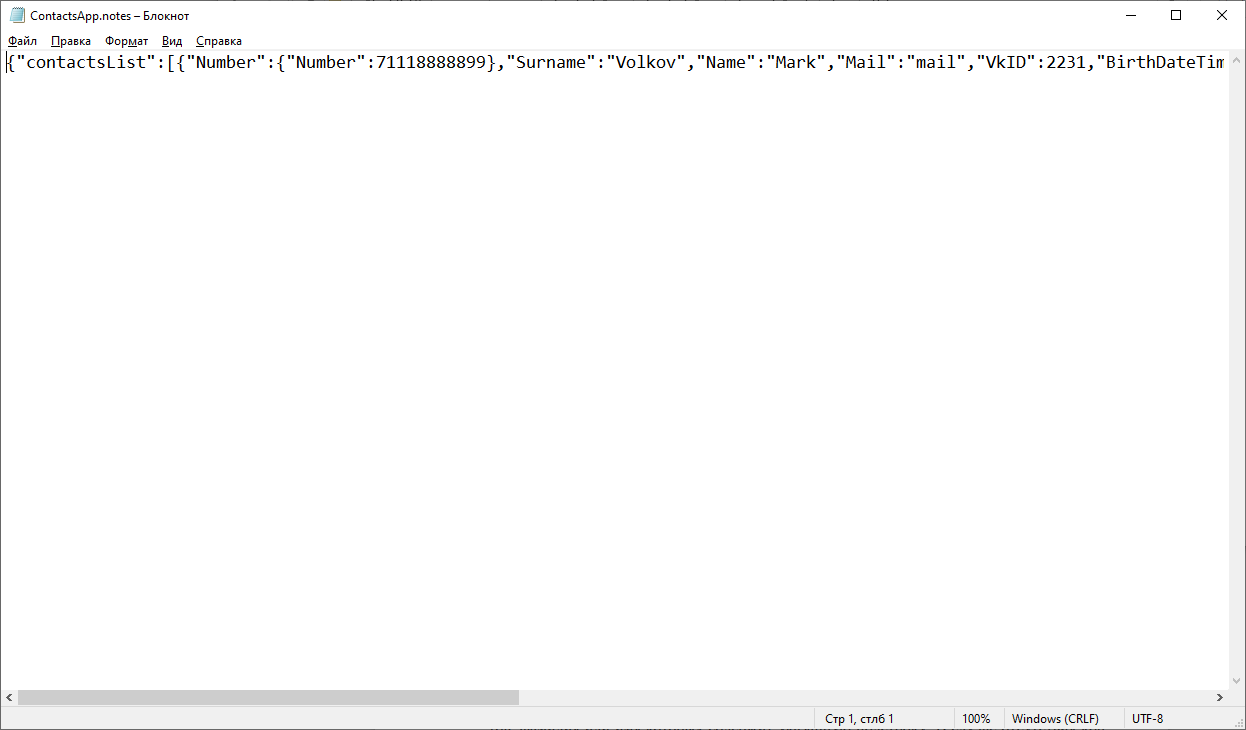


Рисунок 2.3 - Результат вызова методов сериализации

1. Текущая история коммитов ветки develop

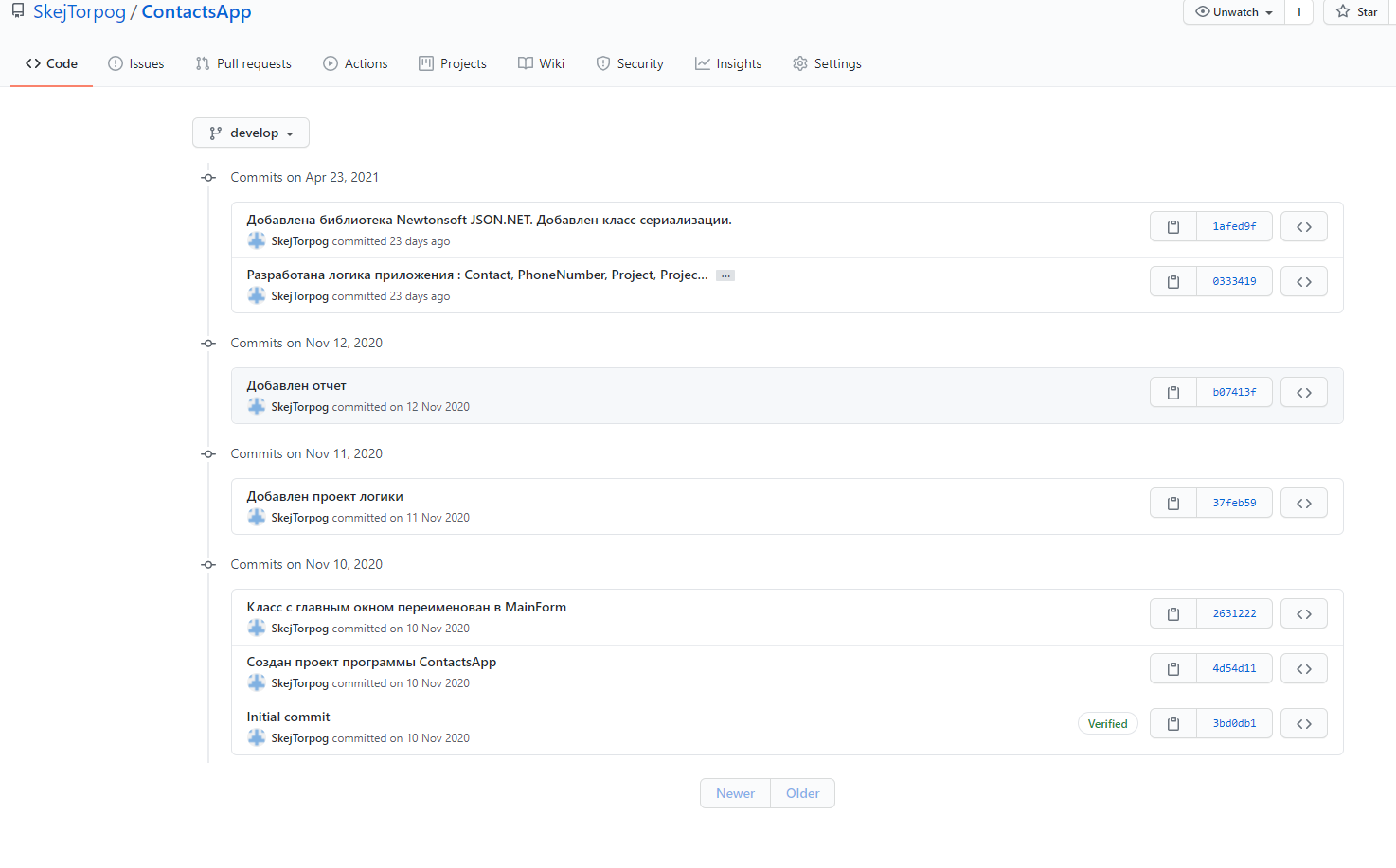


Рисунок 2.4 – Текущая история коммитов ветки develop

# **3 Заключение**

В ходе данной лабораторной работы были изучены требования и процесс разработки логики приложения, разработаны классы, необходимые для работы логики приложения, обеспечена целостность данных классов с помощью свойств и механизма генерации исключений. Также были созданы поля и свойства для этих полей, подключена библиотека JSON.NET и реализована сериализация.