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

по лабораторной работе № 2

«Разработка бизнес-логики приложения» по дисциплине

«Новые технологии в программировании»

Студент гр. 586-1

А. Ахметов

« » \_ 2019 г.

Руководитель доцент каф. КСУП

А.Е. Горяинов

« » \_ 2019 г.

2019

На рисунке 1.1 представлена UML-диаграмма классов

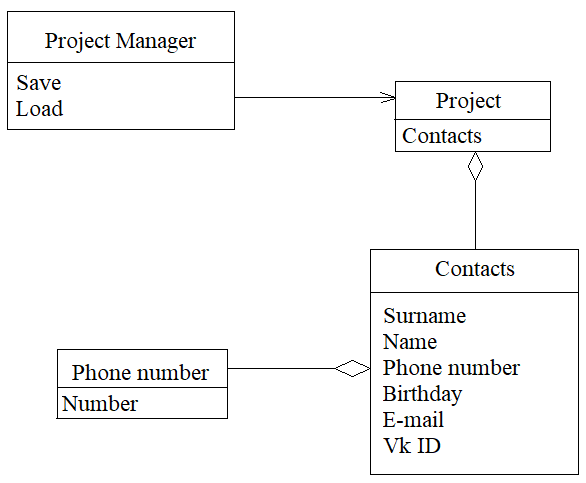


Рисунок 1.1 – UML-диаграмма классов

Класс «Менеджер проекта»

namespace ContactsAppUI

{

/// <summary>

/// Класс "Менеджер проекта". Позволяет сохранять и загружать даннные.

/// </summary>

public static class ProjectManager

{

/// <summary>

/// Метод сохранения

/// </summary>

/// <param name="data">Переменная, которая хранит данные контактов и сохраняет их в текстовый файл</param>

/// <param name="filename">Именование файла</param>

public static void SaveToFile(Project data, string filename)

{

Newtonsoft.Json.JsonSerializer serializer = new Newtonsoft.Json.JsonSerializer();

string documents = Environment.GetFolderPath(Environment.SpecialFolder.Personal);

using (StreamWriter sw = new StreamWriter(documents+@"\json.txt"))

using(Newtonsoft.Json.JsonWriter writer = new Newtonsoft.Json.JsonTextWriter(sw))

{

serializer.Serialize(writer, data);

}

}

/// <summary>

/// Метод загрузки

/// </summary>

/// <param name="filename">Именование файла</param>

private static Project Load(string filename)

{

Project project = null;

Newtonsoft.Json.JsonSerializer serializer = new Newtonsoft.Json.JsonSerializer();

string documents = Environment.GetFolderPath(Environment.SpecialFolder.Personal);

using (StreamReader sr = new StreamReader(documents + @"\json.txt"))

using(Newtonsoft.Json.JsonReader reader = new Newtonsoft.Json.JsonTextReader(sr))

{

project = (Project)serializer.Deserialize<Project>(reader);

}

return project;

}

}

}

Класс «Проект»

namespace ContactsAppUI

{

/// <summary>

/// Класс "Проект". Представляет собой список контактов.

/// </summary>

public class Project

{

/// <summary>

/// Объявление списка.

/// </summary>

public List<Contact> Contacts = new List<Contact>();

}

}

Класс «Контакты»

namespace ContactsAppUI

{

/// <summary>

/// Класс "Контакт". Содержит фамилию, имя, дату рождения, e-mail, id вконтакте.

/// </summary>

public class Contact

{

/// <summary>

/// Поле "Фамилия".

/// </summary>

private string \_surname;

/// <summary>

/// Поле "Имя".

/// </summary>

private string \_name;

/// <summary>

/// Поле "День рождение".

/// </summary>

private DateTime \_birthDay;

/// <summary>

/// Поле "Электронная почта".

/// </summary>

private string \_email;

/// <summary>

/// Поле "ID Вконтакте".

/// </summary>

private string \_idvk;

/// <summary>

/// Свойство поля "Фамилия". Содержит не более 50 символов.

/// </summary>

public string Surname

{

get { return \_surname; }

set

{

if (value.Length > 50)

{

throw new ArgumentException("Длина фамилии не должна быть больше 50 символов");

}

else

\_surname = value;

}

}

/// <summary>

/// Свойство поля "Имя". Содержит не более 50 символов.

/// </summary>

public string Name

{

get { return \_name; }

set

{

if (value.Length > 50)

{

throw new ArgumentException("Длина имени не должна быть больше 50 символов");

}

else

\_name = value;

}

}

/// <summary>

/// Поле "Номер телефона". Реализация представлена в отдельном файле.

/// </summary>

private PhoneNumber PhoneNumber { get; set; }

/// <summary>

/// Свойство поля "День рождения". Содержит информацию о дне, месяце, годе рождения владельца контакта.

/// </summary>

public DateTime BirthDay

{

get { return \_birthDay; }

set

{

if (value>DateTime.Today && value<DateTime.FromBinary(1900))

{

throw new ArgumentException("Дата рождения не может быть раньше 1900 года и позже текущего");

}

else

\_birthDay = value;

}

}

/// <summary>

/// Свойство поля "Электронная почта". Содержит не более 50 символов.

/// </summary>

public string Email

{

get { return \_email; }

set

{

if (value.Length > 50)

{

throw new ArgumentException("Длина e-mail не должна быть больше 50 символов");

}

else

\_email = value;

}

}

/// <summary>

/// Свойство поля "ID Вконтакте". Содержит не более 15 символов.

/// </summary>

public string IDvk

{

get { return \_idvk; }

set

{

if (value.Length > 15)

{

throw new ArgumentException("Длина id не должна быть больше 15 символов");

}

else

\_idvk = value;

}

}

}

}

Класс «Номер телефона»

namespace ContactsAppUI

{

/// <summary>

/// Отдельный класс "Номер телефона".

/// </summary>

public class PhoneNumber

{

/// <summary>

/// Поле "Номер".

/// </summary>

private long \_number;

/// <summary>

/// Свойство поля "Номер". Начинается с 7 и состоит из 11 цифр.

/// </summary>

private long Number

{

get { return \_number; }

set

{

if (value < 70000000000 && value > 70000000000)

{

throw new ArgumentException("Номер телефона должен состоять из 11 цифр и начинаться с 7");

}

else

\_number = value;

}

}

}

}

На рисунке 1.2 представлен репозиторий текущей ветки на GitHub

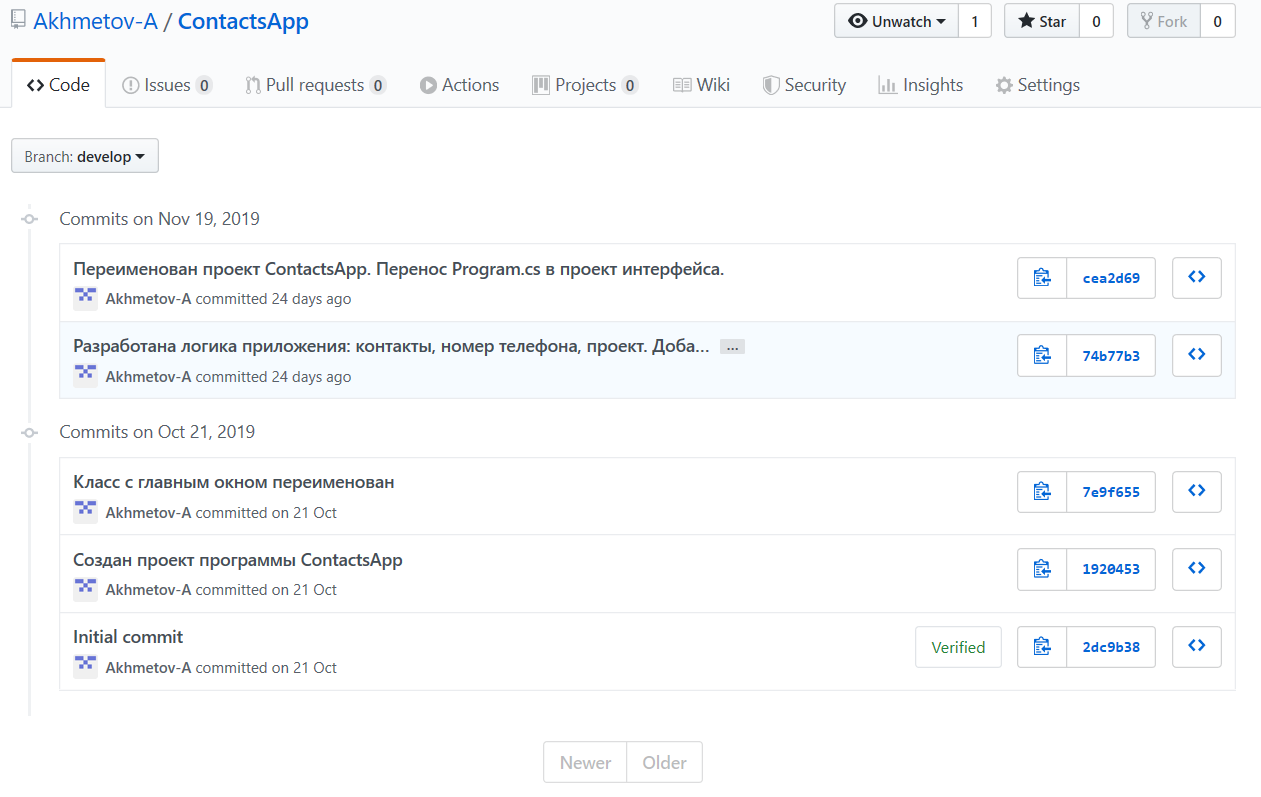


Рисунок 1.2 – репозиторий текущей ветки на GitHub