

Министерство науки и высшего образования Российской Федерации  
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ  
И РАДИОЭЛЕКТРОНИКИ  
(ТУСУР)

Кафедра компьютерных систем в управлении и проектировании

Отчет

Лабораторная работа по курсу новых технологий в  
программировании  
**«Разработка бизнес-логики приложения»**

Преподаватель:

Горяинов А. Е. \_\_\_\_\_

Дата: «    »    20\_\_ г.  
\_\_\_\_\_

Студент гр. 588-1:

Дудик Н. А \_\_\_\_\_

Дата: «    »    20\_\_ г.

**Содержание**

Введение .....	3
Описание и выполнение задания .....	3
Выводы .....	8

## Введение

Целью данной лабораторной работы является изучение типовых требований, предъявляемых к бизнес-логике приложений, а также получение умений разработки логики приложения с обеспечением данных требований.

### 1 Описание и выполнение задания

В рамках данной лабораторной работы требуется изучить требования и процесс разработки приложения, а также разработать классы для работы логики приложения, обеспечив целостность данных этих классов с помощью механизма генерации исключений.

UML-диаграмма классов выглядит следующим образом.

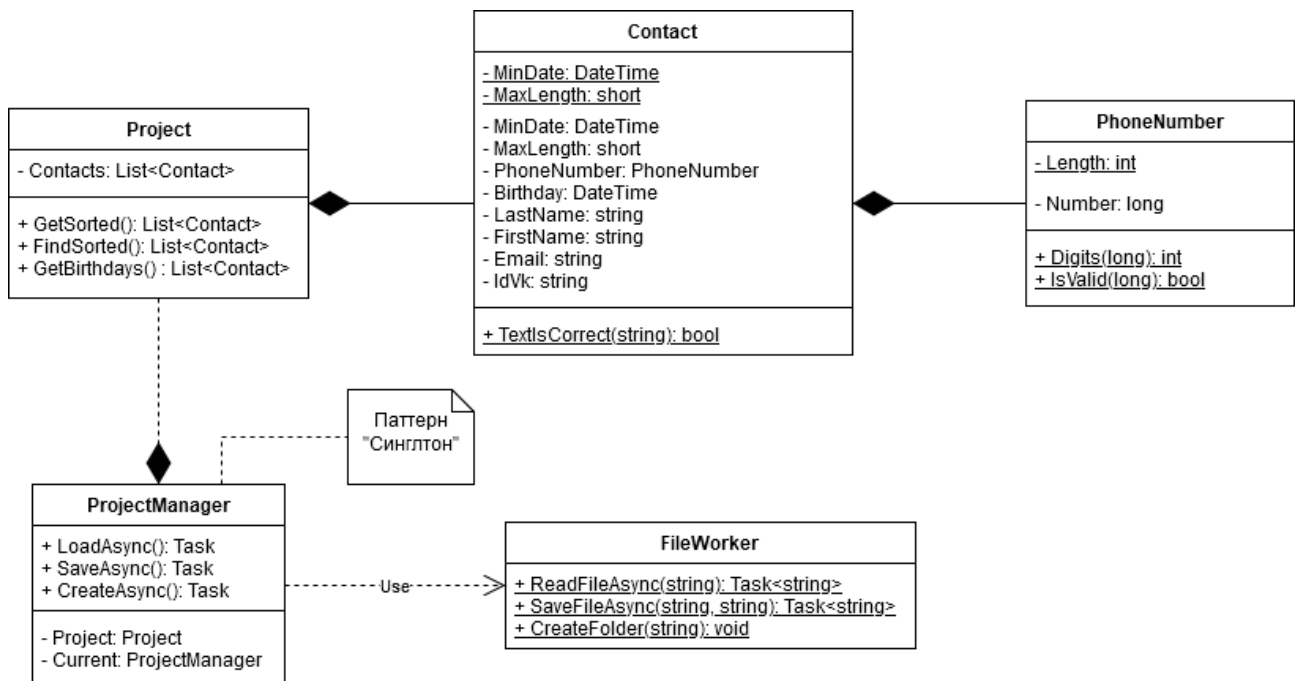


Рисунок 1 UML-диаграмма классов бизнес-логики

Примеры использования класса `Contact` приведены ниже.

```

public Contact Contact { get; private set; }

public EditForm(Contact contact){...}

/// Привязка полей формы к свойствам объекта Contact, а также подписывание событий ...
private void BindControls(){...}

/// Первичная установка значений ...
private void SetValues()
{
    textBox_LastName.Text = Contact.LastName;
    textBox_FirstName.Text = Contact.FirstName;
    dateTimePicker_Birthday.Value = Contact.Birthday;
    maskedTextBox_Phone.Text = PhoneConverter.Mask(Contact.PhoneNumber);
    textBox_Email.Text = Contact.Email;
    textBox_IdVk.Text = Contact.IdVk;
}

```

Рисунок 2 Чтение значений из свойств Contact

```

/// Свойство "IdVk" хранит в себе VK ID контакта. ...
public string IdVk
{
    get => _idVk;
    set
    {
        if (!TextIsCorrect(value))
            throw new ArgumentException( message: "ID ВКонтакте не должен быть длинее "
                                           + MaxTextLength
                                           + " символов");
        _idVk = value;
    }
}

/// Проверяет корректность текстового значения: оно не должно быть null или длинее макс
private static bool TextIsCorrect(string text)
{
    return text != null && text.Length <= MaxTextLength;
}

```

Рисунок 3 Проверка корректности текста методом TextIsCorrect класса Contact

Ниже приведен класс Project, инициализирующий список из Contact, а также имеющий методы сортировки и выборки контактов.

```

public List<Contact> Contacts { get; } = new List<Contact>();

public List<Contact> GetSorted() => Contacts.OrderBy(i :Contact => i.LastName).ToList();

public List<Contact> FindSorted(string query) => Contacts
    .Where(i :Contact => i.LastName.Contains(query)) //IEnumerable<Contact>
    .OrderBy(j :Contact => j.LastName) //IOrderedEnumerable<Contact>
    .ToList(); //List<Contact>

public List<Contact> GetBirthdays(DateTime date) => Contacts
    .Where(i :Contact => i.Birthday.Day == date.Day && i.Birthday.Month == date.Month) //
    .OrderBy(j :Contact => j.LastName) //IOrderedEnumerable<Contact>
    .ToList(); //List<Contact>

```

Рисунок 4 Методы класса Project

```

/// <summary>
/// Обновление списка контактов в контроле ListBox
/// </summary>
private void UpdateData()
{
    // Поиск контактов по введённой строке
    _contacts = new BindingList<Contact>(_contactsViewModel.Find(_query));
    // Повторная привязка DataSource
    listBox_Contacts.DataSource = _contacts;
    if (listBox_Contacts.Items.Count == 0)
    |   SelectContact( index: -1);
}

```

Рисунок 5 Поиск

по контактам для отображения списка на экране

Класс FileWorker, используемый для чтения и записи файлов, используется классом ProjectManager для чтения и записи конкретно экземпляра класса Project.

```

public async Task<Project> Load()
{
    // Создание папки приложения в случае её отсутствия
    FileWorker.CreateFolder(Settings.AppDirectory);
    if (!File.Exists(Settings.ContactsDataFile))
    {
        await Create();
    }
    // Асинхронное чтение файла
    string data = await FileWorker.ReadFileAsync(Settings.ContactsDataFile);
    // Десериализация полученных данных
    Project project = JsonConvert.DeserializeObject<Project>(data);
    return project;
}

public async Task Save(Project project)
{
    // Создание папки приложения в случае её отсутствия
    FileWorker.CreateFolder(Settings.AppDirectory);
    // Сериализация данных
    string data = JsonConvert.SerializeObject(project);
    // Асинхронная запись
    await FileWorker.SaveFileAsync(data, path: Settings.ContactsDataFile);
}

public async Task Create()
{
    await Save(new Project());
}

```

Рисунок 6 Чтение и запись Project через FileWorker

История коммитов в ветке develop на момент написания данного отчёта выглядела следующим образом.

Commits on Jan 22, 2022		
<ProjectManager.cs>:Создан класс менеджера проекта, написаны методы д...	199e582	<>
boratbir committed on 22 Jan		
<Contact.cs, ProjectManager.cs>:Исправления.	18e69f9	<>
boratbir committed on 22 Jan		
<Project.cs>:Создан класс Project, добавлено свойство списка контакто...	caab48d	<>
boratbir committed on 22 Jan		
<Project.cs>:Создан класс Project.	fbaee0c	<>
boratbir committed on 22 Jan		
<Contact.cs>:Добавлено свойство номера телефона.	9c28bd	<>
boratbir committed on 22 Jan		
<PhoneNumber.cs>:Создан класс номера телефона.	cec2d73	<>
boratbir committed on 22 Jan		
<Contact.cs>:Добавлено свойство даты рождения.	85b5746	<>
boratbir committed on 22 Jan		
<Contact.cs>:Создан класс контакта.	4ef9731	<>
boratbir committed on 22 Jan		

Рисунок 7 Последние коммиты

## **2 Выводы**

В результате выполнения данной лабораторной работы была разработана бизнес-логика для приложения, а также построена UML-диаграмма классов.