



Практическая работа № 1

*Курс по основам программирования пройден.
Пришла пора познакомиться с архитектурой!
Мало уметь писать код. Нужно научиться
разумно разбивать своё приложение на части.*

Итак, Вы перешли на второй курс ИКИТе и стали настолько крутыми разработчиками, что Вам поручили разработать новейшую систему «DecanatPRO». Несмотря на то, что система «новейшая» в качестве инструмента Вам предложили использовать WinForm и консоль. Исходные требования к системе на начальном этапе не большие, но в дальнейшем предполагается расширение функционала, поэтому предлагается организовать архитектуру Вашего приложения следующим образом (рис. 1):

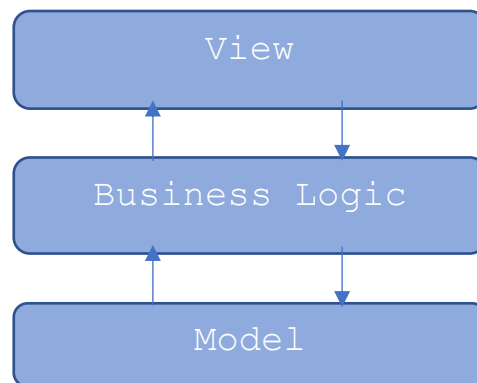


Рис. 1

Уровень модели представляет собой отдельную библиотеку (Model.dll), включающую в себя доменный класс Student со следующими членами класса:

```
public class Student
{
    Ссылка: 0
    public string Name { set; get; }
    Ссылка: 0
    public string Speciality { set; get; }
    Ссылка: 0
    public string Group { set; get; }
}
```

Рис. 2

Функционал системы находится в библиотеке (BusinessLogic.dll) в классе `Logic`, и состоит из следующих функций:

- 1) Добавить нового студента (добавление осуществляется через новое окно, в котором необходимо ввести информацию о ФИО студента, его группе, направлении подготовки)
- 2) Удалить студента
- 3) Вывести весь список в таблицу
- 4) Вывести гистограмму: распределение студентов по специальностям.

Примерная структура класса `Logic` представлена ниже:

```

public class Logic
{
    Ссылка: 0
    public List<Student> students { set; get; }
    = new List<Student>();

    Ссылка: 0
    public void AddStudent(string name, string speciality, string group) {...}
    Ссылка: 0
    public void DeleteStudent(string name, string speciality, string group) {...}
}

```

Рис. 3

Уровень представления реализован двумя вариантами, каждый из которых обращается к одному и тому же функционалу, реализованному в классе `Logic`:

- Многооконное приложение WinForm;
- Консольное приложение.

Помните, что Вам необходимо избегать обращение к модели из уровня представления. То есть уровень view должен взаимодействовать с объектами доменных классов только через слой бизнес-логики.