C#
Краткая информация

Структура класса

```
class имя_класса
{
    уровень_доступа тип имя_переменной1;
    уровень_доступа тип имя_переменной2;

    уровень_доступа тип_возвращаемого_значения имя_метода1(параметры)
    {
        //тело метода
    }

    уровень_доступа тип_возвращаемого_значения имя_метода2(параметры)
    {
        //тело метода
    }
}
```

Пример класса

```
class Student
{
    private string name;
    private int age;

    public Student(string _name, int _age)
    {
        age = _age;
        name = _name;
    }

    public string Result()
    {
        return name + " " + Convert.ToString(age);
    }
}
```

Структура простой программы на C#

```
// Используемые типы в пространстве имен
using System;

// Пространство имён YourNamespace
namespace YourNamespace
{
    // Класс с названием YourMainClass, содержащий метод Main
    class YourMainClass
    {
        // Метод Main (точка входа в программу)
        static void Main(string[] args)
        {
            //Начало кода программы...
        }
    }

    // Класс(-ы) написанный(-ые) пользователем
    class YourClass
    {
        //Код написанный пользователем...
    }
}
```

Пояснения:

1) Директива **using** позволяет в коде вместо «длинных» команд писать «короткие».
2) Ключевое слово **namespace** (пространство имён) используется для объявления области, которая содержит набор связанных объектов. Разные строки кода (в разных папках / файлах, или на разных компьютерах, или у разных программистов), объединённые одним пространством имён, будут считаться одним проектом (одной программой).
3) Ключевое слово **class.** Класс представляет собой шаблон («описание»), по которому определяется реализация объекта. В классе указываются данные и методы, которые будут оперировать этими данными (т.е. *что* обрабатывать - данные, и *как* обрабатывать - методы). На основе класса в дальнейшем создаются объекты (экземпляры класса) с реальными значениями параметров. *Примечание: если на основе класса с именем* ***ClassName*** *создать объект, то у данного объекта будет тип* ***ClassName***

Пример программы

Программа выводит в консоль имя и возраст студента:

```
using System;

namespace Example
{
    class Program
    {
        static void Main(string[] args)
        {
            // Объявление переменной типа "строка"
            string name1 = "Aleks";
            // Объявление переменной типа "целое число"
            int age1 = 20;

            // Создание объекта с именем "student1"
            // и типом "Student", и передача объекту
            // аргументов "name1" и "age1"
            Student student1 = new Student(name1, age1);
            // вызов у объекта "student1" метода "Result"
            // и вывод возвращаемого значения в консоль
            Console.WriteLine(student1.Result());
            // Ожидание консолью нажатия кнопки
            Console.ReadKey();
        }
    }

    // Описание класса с именем "Student"
    // (шаблона для всех объектов типа "Student")
    class Student
    {
        // Объявление переменных внутри класса "Student"
        private string name;
        private int age;

        // Метод-конструктор в классе "Student",
        // описывает первоначальную инициализацию
        // состояния нового (вновь создаваемого) объекта.
        // Метод принимает два аргумента с типами "int" и "string"
        public Student(string _name, int _age)
        {
            age = _age;
            name = _name;
        }

        // Метод с именем "Result". Производит требуемые
        // преобразования и возвращает результат
        public string Result()
        {
            return name + " " + Convert.ToString(age);
        }
    }
}
```

Результат выполнения программы:

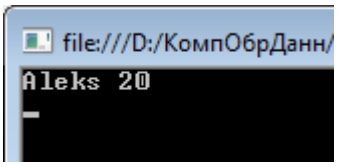
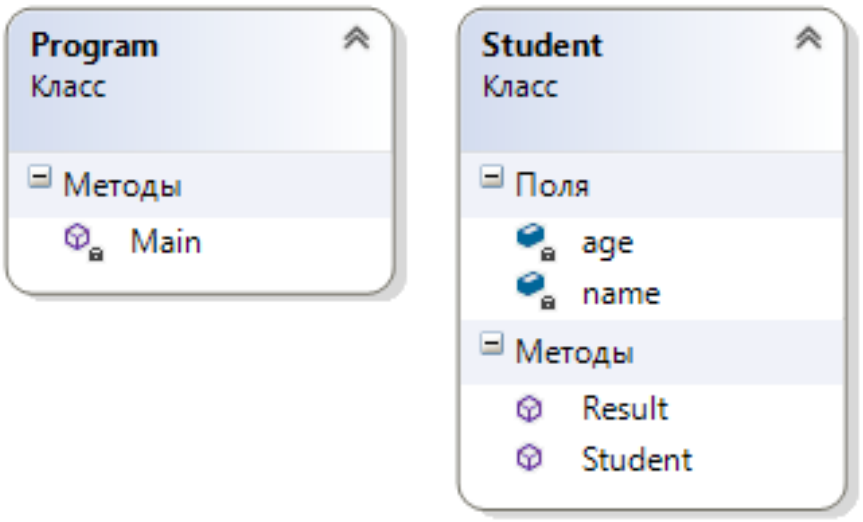


Диаграмма классов:



Описание выполнения программы:

Описание класса **Student** (шаблона для объектов):