

Лабораторная работа №2

Наследование

Наследование — механизм языка, позволяющий описать новый класс на основе уже существующего (родительского, базового) класса или интерфейса. Потомок может добавить собственные методы и свойства, а также пользоваться родительскими методами и свойствами.

Класс, от которого произошло наследование, называется **базовым** или **родительским** (англ. base class). Классы, которые произошли от базового, называются **потомками, наследниками** или **производными классами** (англ. derived class).

Задание 1

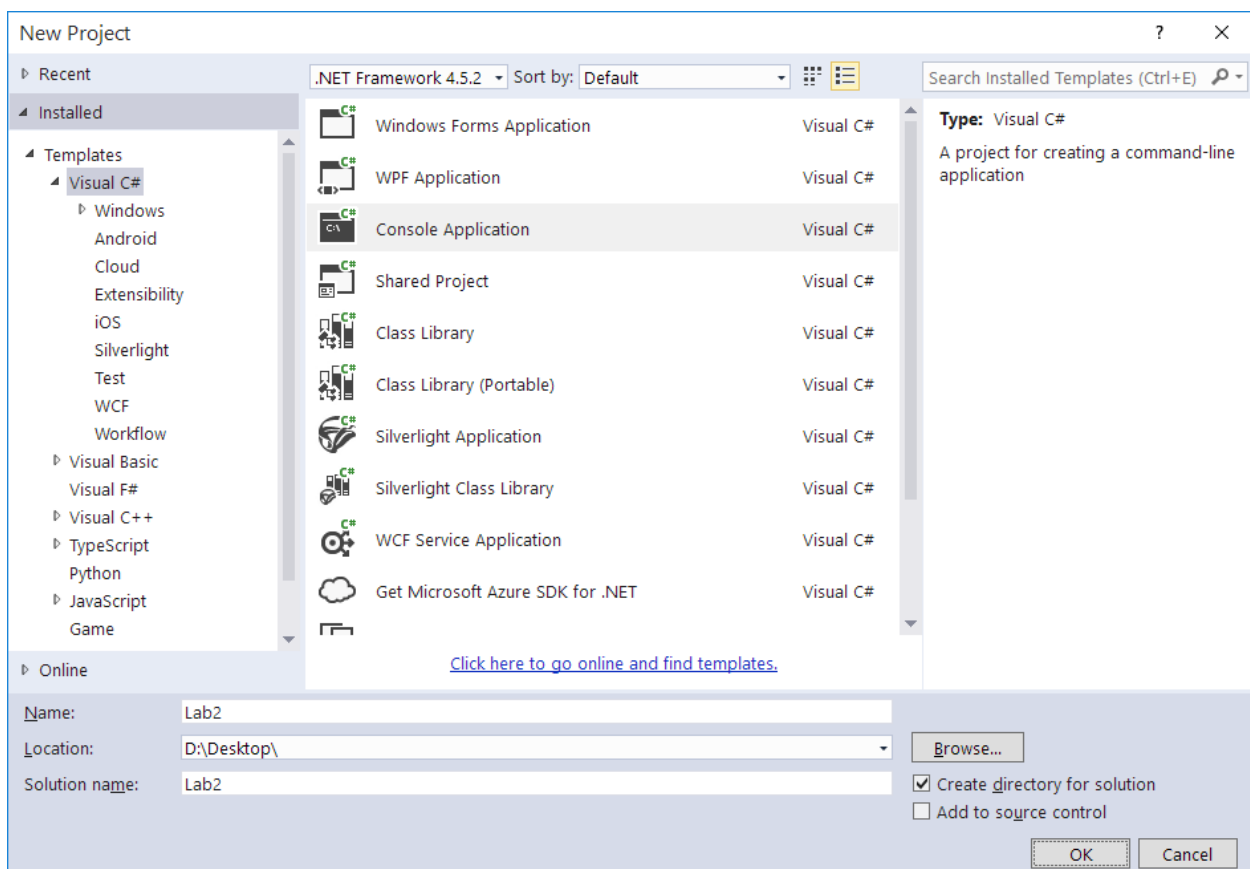
Создать консольное приложение C#, в котором описать иерархию классов, представляющих различные геометрические фигуры. При этом будет применен один из главных принципов ООП - наследование.

1. Запускаем VisualStudio



2. Создаем новый проект (File/New/Project)

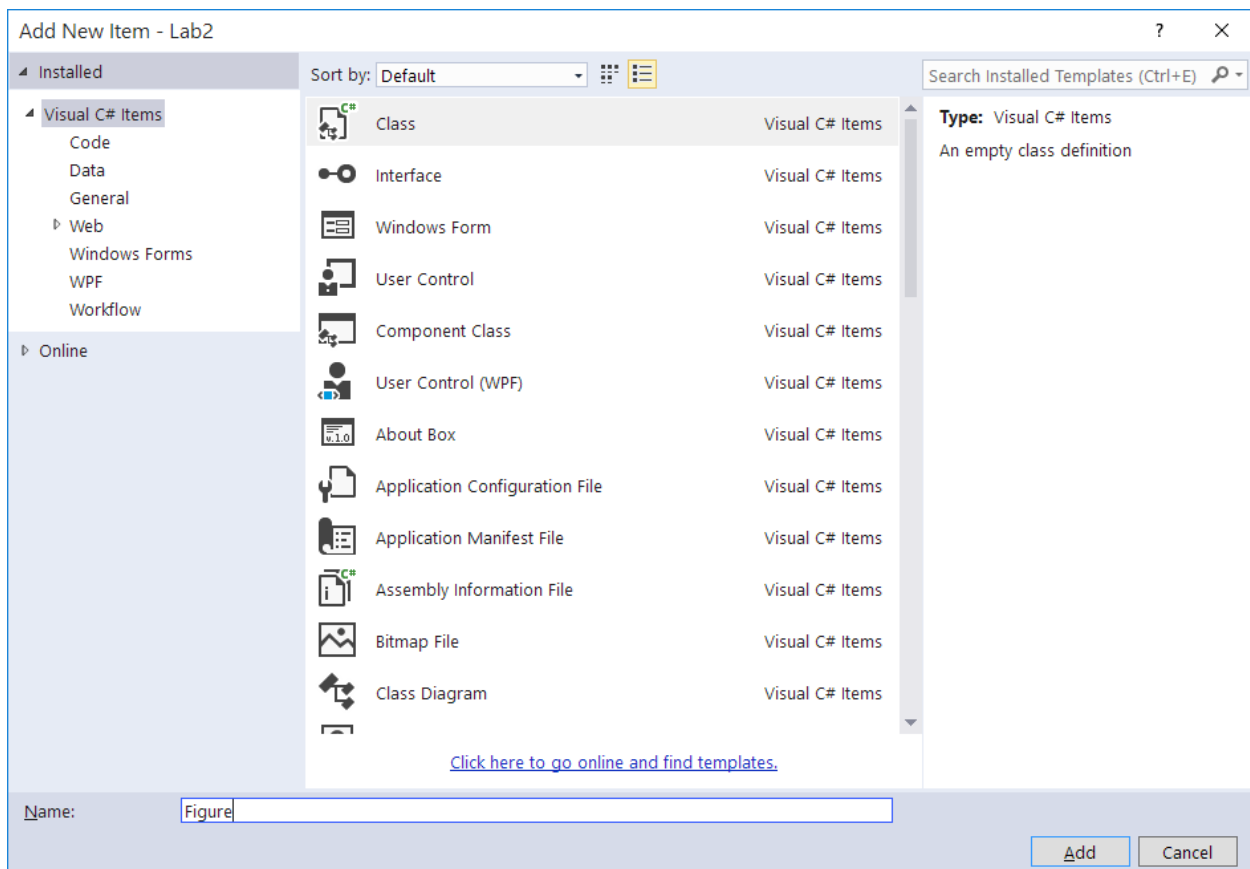
Язык - C#, тип проекта - Console Application, имя проекта - Lab2.



3. Выведем в консоль название лабораторной и имя автора:

```
7 namespace Lab2
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13            Console.WriteLine("Лабораторная №2 - Наследование");
14            Console.WriteLine("Выполнил - Исаак Ньютон");
15        }
16    }
17 }
```

4. Теперь создадим новый класс (Project/Add Class) и назовем его Figure:



5. Создадим публичное свойство **Name** типа **String** для хранения названия фигуры. Также создадим функцию **GetArea** для расчета площади фигуры. Функцию **GetArea** как и сам класс **Figure** пометим как **abstract** (абстрактный), так как мы описываем не конкретно какую-то фигуру, а её абстрактное представление. Таким образом, мы создали **абстрактный** класс, который будем использовать для порождения **классов-потомков** реальных фигур.

```
7 namespace Lab2
8 {
9     public abstract class Figure
10    {
11        public string Name { get; set; }
12
13        public abstract double GetArea();
14    }
15 }
```

6. Создадим новый класс (Project/Add Class) **Rectangle** - прямоугольник. Родителем данного класса будет являться класс **Figure**. Таким образом, свойство **Name** и функция **GetArea** перейдут по наследству и будут также находиться внутри класса **Rectangle**.
7. Добавим публичные свойства **Width** и **Height** (ширину и высоту прямоугольника) типа **double**. А также реализуем расчет его площади. (Слово **override** говорит о том, что мы **переопределяем** функцию из родительского класса)



Когда мы производим наследование от абстрактного класса, мы должны обязательно реализовать его абстрактные функции. (В нашем случае **GetArea**)

```

7 namespace Lab2
8 {
9     class Rectangle : Figure
10    {
11        public double Width { get; set; }
12        public double Height { get; set; }
13
14        public override double GetArea()
15        {
16            return Width * Height;
17        }
18    }
19 }

```

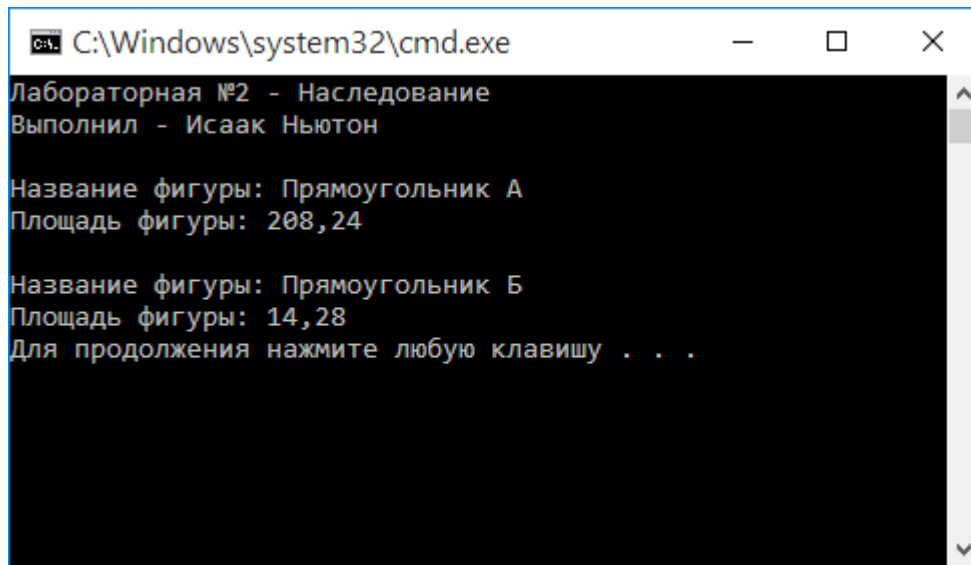
8. В процедуре **main** создадим несколько прямоугольников и рассчитаем их площадь:

```

7 namespace Lab2
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13            Console.WriteLine("Лабораторная №2 - Наследование");
14            Console.WriteLine("Выполнил - Исаак Ньютон");
15
16            // Создаем прямоугольники А и В и на месте зададим их параметры
17            Rectangle a = new Rectangle()
18            {
19                Name = "Прямоугольник А", Width = 15.2, Height = 13.7
20            };
21
22            Rectangle b = new Rectangle()
23            {
24                Name = "Прямоугольник В", Width = 5.1, Height = 2.8
25            };
26
27            // Выведем информацию о прямоугольниках
28            Console.WriteLine();
29            Console.WriteLine("Название фигуры: {0}", a.Name);
30            Console.WriteLine("Площадь фигуры: {0}", a.GetArea());
31            Console.WriteLine();
32            Console.WriteLine("Название фигуры: {0}", b.Name);
33            Console.WriteLine("Площадь фигуры: {0}", b.GetArea());
34        }
35    }
36 }

```

9. Запустим приложение:



```
C:\Windows\system32\cmd.exe
Лабораторная №2 - Наследование
Выполнил - Исаак Ньютон

Название фигуры: Прямоугольник А
Площадь фигуры: 208,24

Название фигуры: Прямоугольник Б
Площадь фигуры: 14,28
Для продолжения нажмите любую клавишу . . .
```

Чтобы в полной мере осознать содеянное, вернемся к определению.

Наследование — механизм языка, позволяющий описать новый класс на основе уже существующего (родительского, базового) класса или интерфейса. Потомок может добавить собственные методы и свойства, а также пользоваться родительскими методами и свойствами.



Поздравляю тебя!

Ты задействовал один из главных принципов объектно-ориентированного программирования - **наследование**. Ты создал класс **Rectangle** на основе существующего класса **Figure**. Добавил в класс потомок собственные свойства (**Width** и **Height**) и воспользовался родительским методом (функцией) **GetArea** (изменив его).

Задание 2

По аналогии с заданием 1 создать классы следующих фигур: прямоугольник (уже готово), круг, квадрат, треугольник, трапеция, ромб, параллелограмм, правильный пятиугольник и правильный десятиугольник. Задать все необходимые свойства каждой геометрической фигуры и рассчитать её площадь. Информацию о каждой фигуре вывести на экран.

Вопросы

1. Что такое наследование?
2. Как в C# наследовать класс A от класса B?
3. Что означает слово `abstract`?
4. Что такое переопределение функции (метода)?
5. Что означает слово `override`?