

Лабораторная работа №3

Полиморфизм

Полиморфизм — это свойство системы использовать объекты с одинаковым интерфейсом без информации о типе и внутренней структуре объекта.

При использовании термина «полиморфизм» в сообществе ООП подразумевается полиморфизм подтипов; а использование параметрического полиморфизма называют обобщённым программированием.

Задание 1

На основе лабораторной работы №2 создать консольное приложение, которое будет содержать классы следующих фигур: прямоугольник, круг, квадрат, треугольник, трапеция, ромб, параллелограмм, правильный пятиугольник и правильный десятиугольник. В каждый класс добавить координаты самой фигуры, её цвет и функцию определения координат центра фигуры. Нарисовать фигуры на форме и внутри каждой фигуры отобразить её площадь.

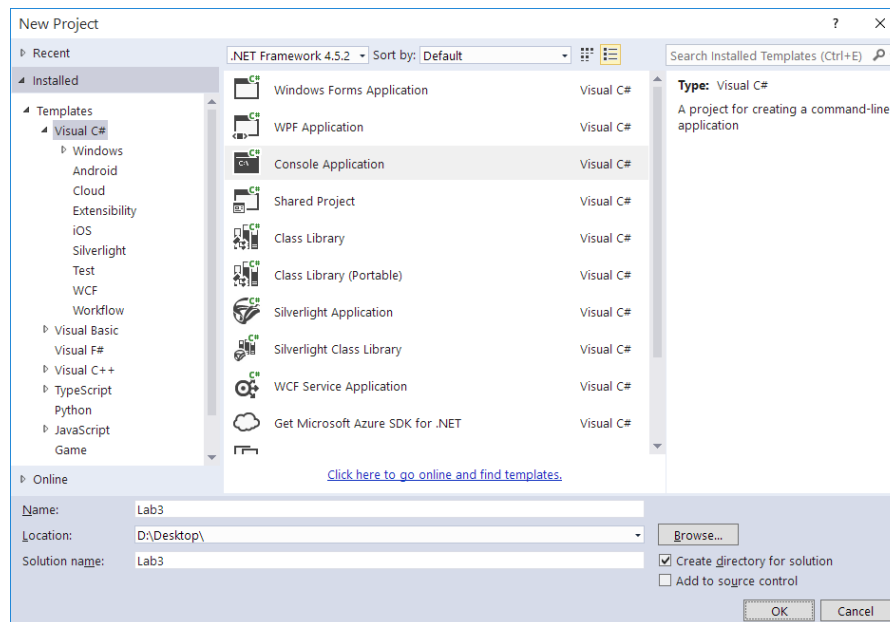
При этом будет применен один из главных принципов ООП - полиморфизм.

1. Запускаем VisualStudio



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

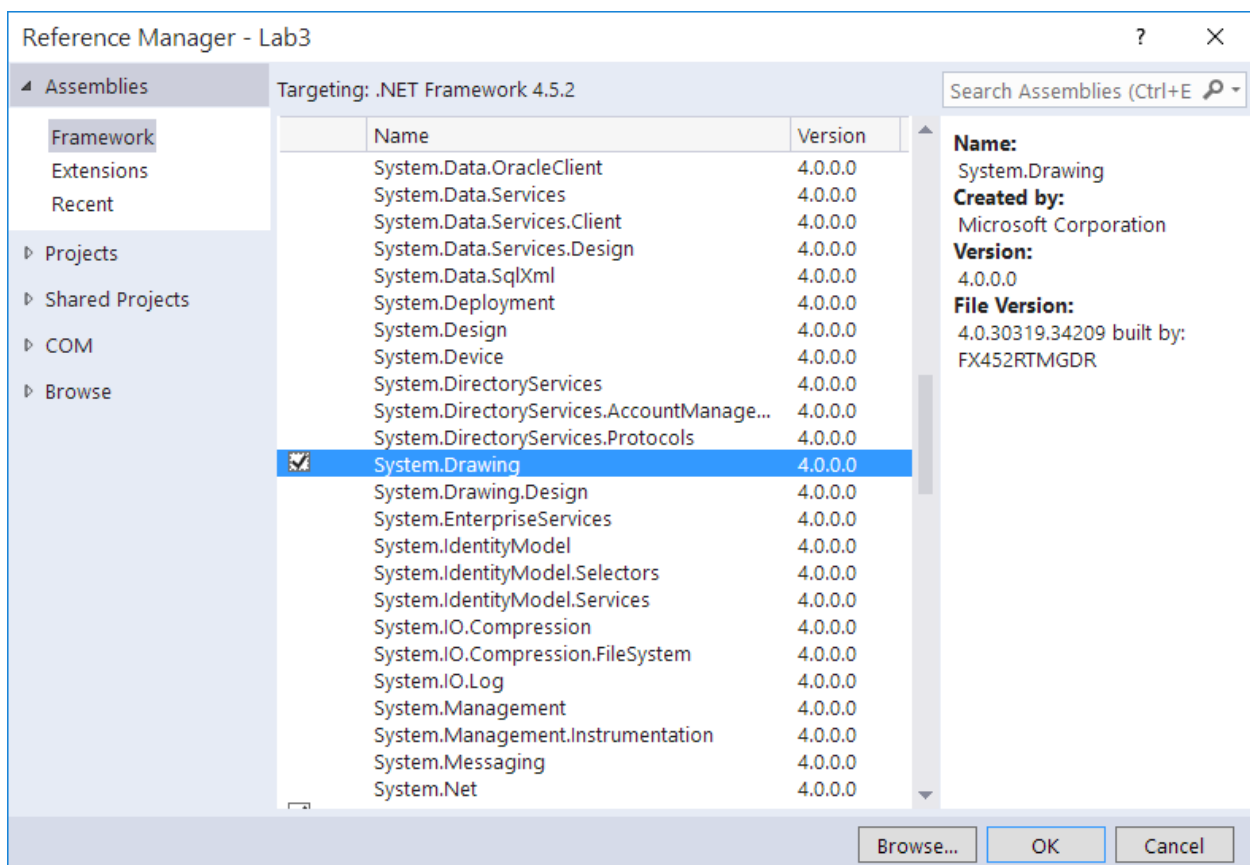
Язык - C#, тип проекта - Console Application, имя проекта – Lab3.



3. Выведем в консоль название лабораторной и имя автора:

```
7 namespace Lab3
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13            Console.WriteLine("Лабораторная работа №3 - Полиморфизм");
14            Console.WriteLine("Выполнил - Николай Коперник");
15        }
16    }
17 }
```

4. Подключим (Project/Add Reference) к проекту библиотеку **System.Drawing** для рисования фигур:



5. Создадим в классе (из лабораторной работы №2) **Figure** новую абстрактную функцию **GetCenter**, которая получает значение координат центра фигуры, поле **Position**, которое будет содержать координаты самой фигуры и поле **Color**, содержащее цвет фигуры.

```
7 namespace Lab3
8 {
9     public abstract class Figure
10    {
11        public string Name { get; set; }
12
13        public System.Drawing.Color Color { get; set; }
14
15        public System.Drawing.Point Position { get; set; }
16
17        public abstract double GetArea();
18
19        public abstract System.Drawing.Point GetCenter();
20    }
21 }
```

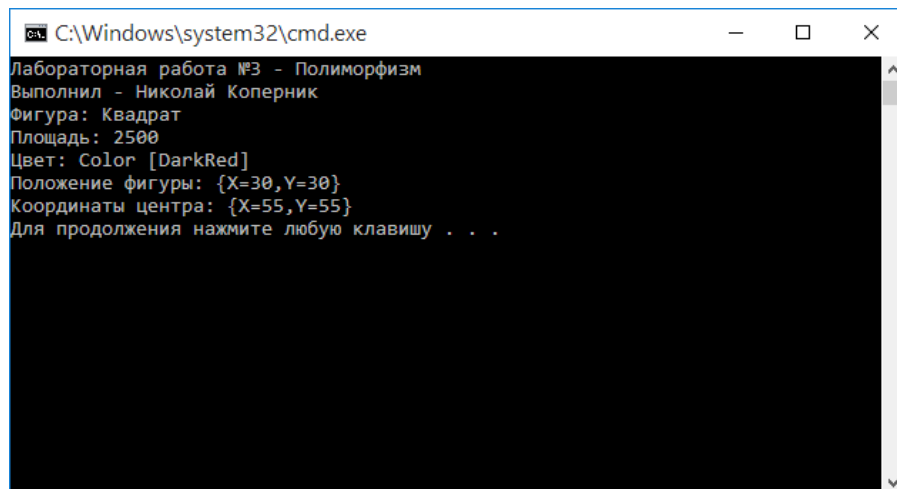
6. Реализуем функцию **GetCenter** для каждой фигуры (здесь только для **Rectangle** в качестве примера):

```
1  using System;
2  using System.Collections.Generic;
3  using System.Drawing;
4  using System.Linq;
5  using System.Text;
6  using System.Threading.Tasks;
7
8  namespace Lab3
9  {
10     class Rectangle : Figure
11     {
12         public double Width { get; set; }
13         public double Height { get; set; }
14
15         public override double GetArea()
16         {
17             return Width * Height;
18         }
19
20         public override Point GetCenter()
21         {
22             return new Point((int)(Position.X + Width / 2), (int)(Position.Y + Height / 2));
23         }
24     }
25 }
```

7. Создадим экземпляр класса **Rectangle** в файле **Program.cs** и выведем всю информацию о нем.

```
7 namespace Lab3
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13            Console.WriteLine("Лабораторная работа №3 - Полиморфизм");
14            Console.WriteLine("Выполнил - Николай Коперник");
15
16            Figure f = new Rectangle()
17            {
18                Name = "Квадрат",
19                Color = System.Drawing.Color.DarkRed,
20                Position = new System.Drawing.Point(30, 30),
21                Width = 50, Height = 50
22            };
23
24            Console.WriteLine("Фигура: " + f.Name);
25            Console.WriteLine("Площадь: " + f.GetArea());
26            Console.WriteLine("Цвет: " + f.Color);
27            Console.WriteLine("Положение фигуры: " + f.Position);
28            Console.WriteLine("Координаты центра: " + f.GetCenter());
29        }
30    }
31 }
```

И получим:



```
C:\Windows\system32\cmd.exe
Лабораторная работа №3 - Полиморфизм
Выполнил - Николай Коперник
Фигура: Квадрат
Площадь: 2500
Цвет: Color [DarkRed]
Положение фигуры: {X=30,Y=30}
Координаты центра: {X=55,Y=55}
Для продолжения нажмите любую клавишу . . .
```

8. Создадим форму и нарисуем на ней несколько прямоугольников, для этого:

- подключим библиотеку **System.Windows.Forms** и добавим соответствующее пространство имен.

```
6 using System.Windows.Forms;
```

- Создадим новую форму с заголовком **Лабораторная №3 – Полиморфизм**, размером **800x600**, и расположим её в **центре** экрана.

```
Form frm = new Form()
{
    Text = "Лабораторная №3 - Полиморфизм",
    Size = new System.Drawing.Size(800, 600),
    StartPosition = FormStartPosition.CenterScreen
};

Application.Run(frm);
```

- Для того, чтобы нарисовать фигуру на форме, нужно добавить в класс **Figure** абстрактную функцию **Draw** и определить эту функцию в классе **Rectangle**.

```
public override void Draw(Graphics gr)
{
    // Рисуем прямоугольник
    gr.DrawRectangle(new Pen(Color), Position.X, Position.Y, (int)Width, (int)Height);

    // Рисуем информацию о координатах его центра
    gr.DrawString(GetCenter().ToString(), new Font("Arial", 9), Brushes.Black, GetCenter());
}
```

- Создадим массив фигур и разместим в нем несколько разных прямоугольников.

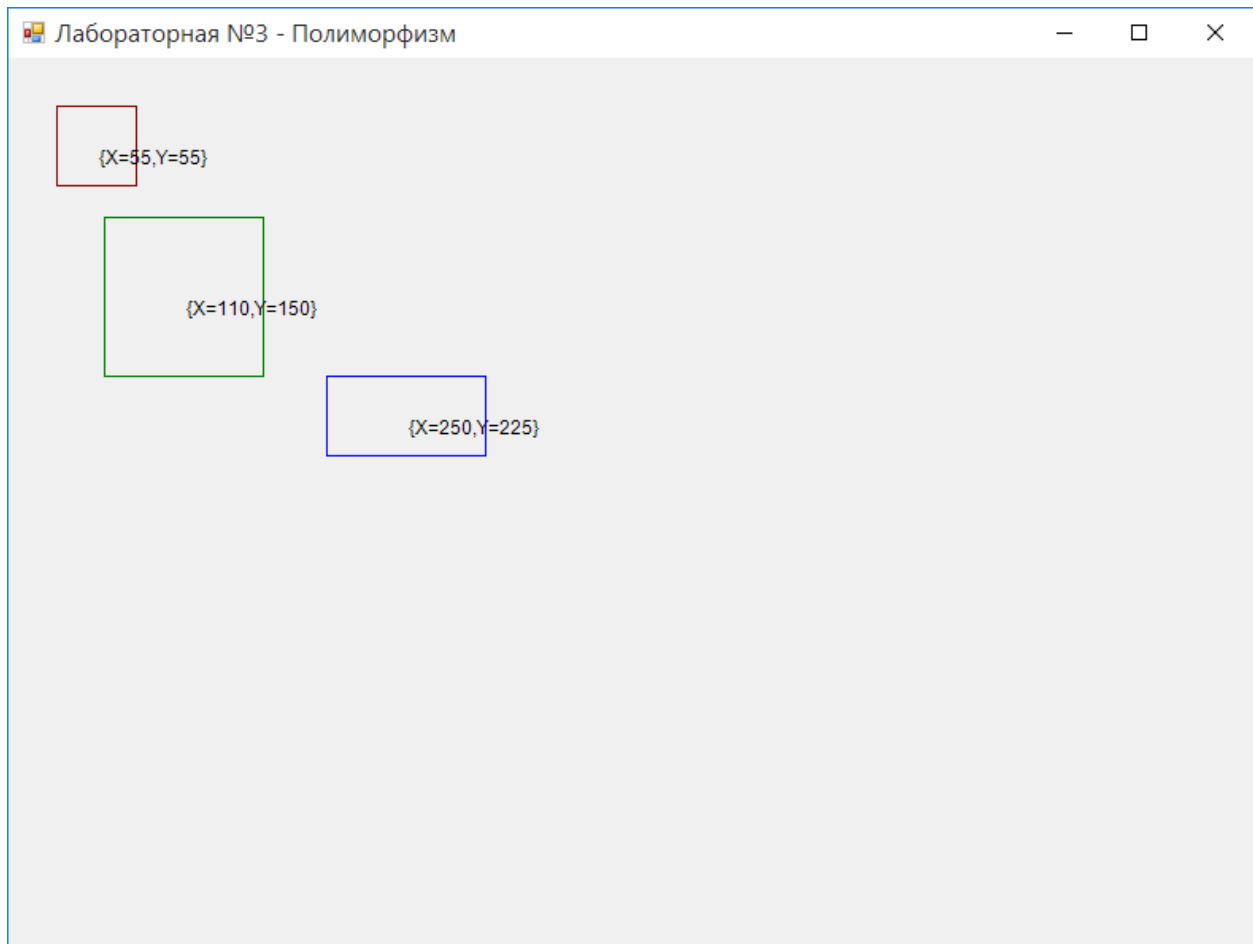
```
10 class Program
11 {
12     public static Figure[] figures = {
13         new Rectangle()
14         {
15             Name = "Квадрат #1",
16             Color = System.Drawing.Color.DarkRed,
17             Position = new System.Drawing.Point(30, 30),
18             Width = 50, Height = 50
19         },
20
21         new Rectangle()
22         {
23             Name = "Квадрат #2",
24             Color = System.Drawing.Color.Green,
25             Position = new System.Drawing.Point(60, 100),
26             Width = 100, Height = 100
27         },
28
29         new Rectangle()
30         {
31             Name = "Прямоугольник #1",
32             Color = System.Drawing.Color.Blue,
33             Position = new System.Drawing.Point(200, 200),
34             Width = 100, Height = 50
35         }
36     };
37 }
```


- Чтобы отобразить фигуру на форме и воспользоваться созданной функцией **Draw** нужно задействовать событие **Paint**.

```
37 static void Main(string[] args)
38 {
39     Console.WriteLine("Лабораторная работа №3 - Полиморфизм");
40     Console.WriteLine("Выполнил - Николай Коперник");
41
42     Form frm = new Form()
43     {
44         Text = "Лабораторная №3 - Полиморфизм",
45         Size = new System.Drawing.Size(800, 600),
46         StartPosition = FormStartPosition.CenterScreen
47     };
48
49     frm.Paint += Frm_Paint;
50
51     Application.Run(frm);
52 }
53
54 private static void Frm_Paint(object sender, PaintEventArgs e)
55 {
56     foreach (Figure f in figures)
57     {
58         f.Draw(e.Graphics);
59     }
60 }
61
62 }
```

- В событии **Paint** с помощью цикла **foreach** перебираем все фигуры из массива **figures** и вызываем функцию **Draw**, для отображения фигур на форме. В этот момент как раз и происходит реализация **полиморфизма** (в массиве **figures** могут быть не только прямоугольники, а все производные от класса **Figure**).

- В результате получим:



- **Осталось только реализовать приведенные выше функции для всех фигур и вывести их на форму!**

Чтобы в полной мере осознать содеянное, вернемся к определению.

Полиморфизм — это свойство системы использовать объекты с одинаковым интерфейсом без информации о типе и внутренней структуре объекта. Таким образом мы задействовали объекты с общим интерфейсом, объявленным в классе **Figure** и отображали на форме эти фигуры, используя функцию **Draw**. При этом не важно какие именно фигуры используются: прямоугольники, окружности, треугольники. Главное, чтобы они имели общего предка **Figure**.