«Колледж информатики и программирования при Финансовом Университете

при Правительстве Российской Федерации»

**Практические работы №5 и №7**

**По предмету «Технологии разработки программного обеспечения»**

Выполнил студент

Группы 3ПКС-116

Зайцев Н.В.

Москва 2019

**Практическая работа №5**

Создайте новое консольное приложение. Откройте файл проекта с исходным кодом. После открывающей фигурной скобки пространства имён создайте перечисление с именем Figures. Добавьте в это перечисление несколько именованных констант, например, Romb, Circle, Rect и т.д.

Ниже создайте структуру с именем Fdata. Поместите в структуру следующие данные:

public int x0,y0; // координаты центра тяжести

public int color; // цвет фигуры

public Figures type; // тип фигуры

В методе Main() создайте и инициализируйте структурную переменную типа Fdata:

// вариант 1

Fdata fd;

fd.x0=1; fd.y0=1;

fd.color=0x2727FF;

fd.type=Figures.Romb;

// или вариант 2

Fdata fd=new Fdata() { x0=1, y0=1, color=0x2727FF, type=Figures.Romb };

Добавьте код для вывода всех полей структурной переменной на экран. Запустите программу.

**Код**

using System;  
using System.ComponentModel;  
using System.Collections;  
using System.Collections.ObjectModel;  
using System.Linq;  
using System.IO;  
  
  
namespace ConsoleApp1  
{  
 class Program  
 {  
 struct Fdata  
 {  
 public int x0,y0;   
 public int color;   
 public Figures type;  
  
 public Fdata(int \_x0, int \_y0, int \_color, Figures \_type)  
 {  
 x0 = \_x0;  
 y0 = \_y0;  
 color = \_color;  
 type = \_type;  
 }  
  
 }  
 enum Figures  
 {  
 Romb, Circle, Rect, Quad  
 }  
  
 static void ShowFdataObj(Fdata obj)  
 {  
 Console.WriteLine($"Type: {obj.type}");  
 Console.WriteLine($"Coords: x0 - {obj.x0}, y0 - {obj.y0}");  
 Console.WriteLine($"Color: {obj.color}");  
 }  
 static void Main()  
 {  
 Fdata fd = new Fdata(1, 1,0x2727FF,Figures.Romb);  
 ShowFdataObj(fd);  
   
 }  
 }  
}

**Практическая работа №7**

**Задание 1**

Создайте новое консольное приложение. Откройте файл проекта с исходным кодом. Добавьте в класс новый статический метод с именем «ffactorial». Метод должен возвращать тип long и иметь один входной параметр «int n». В теле метода объявите переменную «res» с типом «long». Реализуйте цикл по «i» от 1 до «n», а в теле цикла вычисляйте результат по формуле «res \*= i». В конце метода верните значение переменной «res».

Добавьте в класс новый статический метод «rfactorial». Метод должен возвращать тип long и иметь один входной параметр «int n». В теле метода добавьте оператор:

return n<1 ? 1 : n\*rfactorial(n-1);

В методе Main() вызовите оба созданных метода и распечатайте результат. Запустите программу

**Код**

using System;  
using System.ComponentModel;  
using System.Collections;  
using System.Collections.ObjectModel;  
using System.Linq;  
using System.IO;  
  
  
namespace ConsoleApp1  
{  
 class Program  
 {  
 static long ffactorial(int n)  
 {  
 long res = 1;  
 for (int i = 1; i < n; i++)  
 {  
 res \*= i;  
 }  
  
 return res;  
 }  
  
 static long rfactorial(int n)  
 {  
 return n < 1 ? 1 : n \* rfactorial(n - 1);  
 }  
  
 static void Main()  
 {  
 Console.WriteLine(ffactorial(5));  
 Console.WriteLine(rfactorial(5));  
 }  
 }  
}

**Задание 2**

Откройте приложение Geometry из каталога mod2. Откройте класс Program и создайте в нём новый статический метод «area», который возвращает «double» и получает на вход один параметр «Fdata fd».

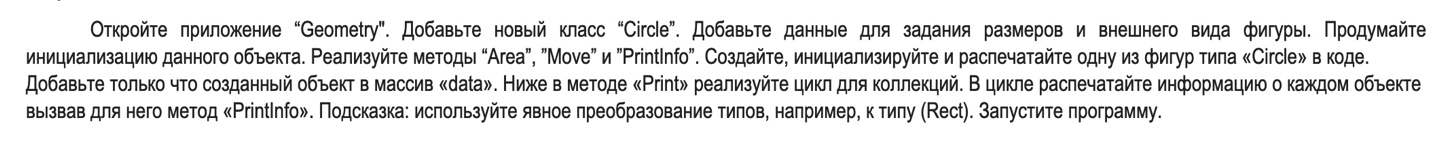
Объявите в теле этого метода переменную «res» с типом «double» и нулевым начальным значением. Ниже добавьте оператор «switch» по типу геометрической фигуры. Для каждого типа фигуры из перечисления Figures реализуйте вычисление площади. Верните результат.

В методе Main() снимите комментарии и запустите программу.

**Код**

using System;  
using System.ComponentModel;  
using System.Collections;  
using System.Collections.ObjectModel;  
using System.Linq;  
using System.IO;  
  
  
namespace ConsoleApp1  
{  
 class Program  
 {  
 struct Fdata  
 {  
 public int x0,y0;   
 public int color;   
 public Figures type;  
  
 public Fdata(int \_x0, int \_y0, int \_color, Figures \_type)  
 {  
 x0 = \_x0;  
 y0 = \_y0;  
 color = \_color;  
 type = \_type;  
 }  
  
 }  
 enum Figures  
 {  
 Square,  
 Rectangle,  
 Triangle,  
 Parallelogram,  
 Trapeze,  
 Circle,  
 Rhombus,   
 }  
  
 static void ShowFdataObj(Fdata obj)  
 {  
 Console.WriteLine($"Type: {obj.type}");  
 Console.WriteLine($"Coords: x0 - {obj.x0}, y0 - {obj.y0}");  
 Console.WriteLine($"Color: {obj.color}");  
 }  
  
 static double Area(Fdata obj)  
 {  
 double res;  
 switch (obj.type)  
 {  
 case Figures.Trapeze:  
 {  
 int a, b, h;  
 Console.WriteLine("Введите длину верхнего основания трапеции:");  
 a = Convert.ToInt32(Console.ReadLine());  
 Console.WriteLine("Введите длину нижнего основания трапеции:");  
 b = Convert.ToInt32(Console.ReadLine());  
 Console.WriteLine("Введите длину высоты трапеции:");  
 h = Convert.ToInt32(Console.ReadLine());  
 res = 0.5 \* h \* (a + b);  
 break;  
 }  
 case Figures.Parallelogram:  
 {  
 int a, h;  
 Console.WriteLine("Введите длину основания параллелограмма:");  
 a = Convert.ToInt32(Console.ReadLine());  
 Console.WriteLine("Введите высоту параллелограмма:");  
 h = Convert.ToInt32(Console.ReadLine());  
 res = a \* h;  
 break;  
 }  
 case Figures.Triangle:  
 {  
 int b, h;  
 Console.WriteLine("Введите длину основания треугольника:");  
 b = Convert.ToInt32(Console.ReadLine());  
 Console.WriteLine("Введите высоту ромба:");  
 h = Convert.ToInt32(Console.ReadLine());  
 res = 0.5 \* b \* h;  
 break;  
 }  
 case Figures.Square:  
 {  
 int a;  
 Console.WriteLine("Введите длину стороны квадрата:");  
 a = Convert.ToInt32(Console.ReadLine());  
 res = a \* a;  
 break;  
 }  
 case Figures.Rhombus:  
 {  
 int a, h;  
 Console.WriteLine("Введите длину основание ромба:");  
 a = Convert.ToInt32(Console.ReadLine());  
 Console.WriteLine("Введите высоту ромба:");  
 h = Convert.ToInt32(Console.ReadLine());  
 res = a \* h;  
 break;  
 }  
 case Figures.Rectangle:  
 {  
 int a, b;  
 Console.WriteLine("Введите 1 сторону прямоугольника:");  
 a = Convert.ToInt32(Console.ReadLine());  
 Console.WriteLine("Введите 2 сторону прямоугольника:");  
 b = Convert.ToInt32(Console.ReadLine());  
 res = a \* b;  
 break;  
 }  
 case Figures.Circle:  
 {  
 int r;  
 Console.WriteLine("Введите радиус круга:");  
 r = Convert.ToInt32(Console.ReadLine());  
 res = Math.PI \* (r \* r);  
 break;  
 }  
 default:  
 {  
 res = -1;  
 break;  
 }  
 }  
  
 return res;  
 }  
 static void Main()  
 {  
 Fdata fd1 = new Fdata(1, 1,0x2727FF,Figures.Rhombus);  
 Fdata fd2 = new Fdata(1, 1,0x2727FF,Figures.Circle);  
 ShowFdataObj(fd1);  
 ShowFdataObj(fd2);  
 Console.WriteLine(Area(fd1));  
 Console.WriteLine(Area(fd2));  
 }  
 }  
}

**Задание 3**



**Код**

using System;  
using System.ComponentModel;  
using System.Collections;  
using System.Collections.ObjectModel;  
using System.Linq;  
using System.IO;  
  
  
namespace ConsoleApp1  
{  
 class Cirlce  
 {  
 public int x0, y0, Radius;  
 public string Color;  
  
 public Cirlce(int \_x0, int \_y0, int \_Radius, string \_Color)  
 {  
 x0 = \_x0;  
 y0 = \_y0;  
 Radius = \_Radius;  
 Color = \_Color;  
 }  
  
 public void PrintInfo()  
 {  
 Console.WriteLine($"Type: Circle-class");  
 Console.WriteLine($"Coords: x0 - {x0}, y0 - {y0}");  
 Console.WriteLine($"Color: {Color}");  
 Console.WriteLine($"Radius: {Radius}");  
 }  
  
 public void Area()  
 {  
 Console.WriteLine($"Area of the circle - {Math.PI \* (Radius \* Radius)}");  
 }  
  
 public void Move(int new\_x0, int new\_y0)  
 {  
 Console.WriteLine($"Moving circle from {x0} {y0} to {new\_x0} {new\_y0}");  
 x0 = new\_x0;  
 y0 = new\_y0;  
 }  
 }  
  
 class Program  
 {  
 struct Fdata  
 {  
 public int x0, y0;  
 public int color;  
 public Figures type;  
  
 public Fdata(int \_x0, int \_y0, int \_color, Figures \_type)  
 {  
 x0 = \_x0;  
 y0 = \_y0;  
 color = \_color;  
 type = \_type;  
 }  
 }  
  
 enum Figures  
 {  
 Square,  
 Rectangle,  
 Triangle,  
 Parallelogram,  
 Trapeze,  
 Circle,  
 Rhombus,  
 }  
  
 static void ShowFdataObj(Fdata obj)  
 {  
 Console.WriteLine($"Type: {obj.type}");  
 Console.WriteLine($"Coords: x0 - {obj.x0}, y0 - {obj.y0}");  
 Console.WriteLine($"Color: {obj.color}");  
 }  
  
 static double Area(Fdata obj)  
 {  
 double res;  
 switch (obj.type)  
 {  
 case Figures.Trapeze:  
 {  
 int a, b, h;  
 Console.WriteLine("Введите длину верхнего основания трапеции:");  
 a = Convert.ToInt32(Console.ReadLine());  
 Console.WriteLine("Введите длину нижнего основания трапеции:");  
 b = Convert.ToInt32(Console.ReadLine());  
 Console.WriteLine("Введите длину высоты трапеции:");  
 h = Convert.ToInt32(Console.ReadLine());  
 res = 0.5 \* h \* (a + b);  
 break;  
 }  
 case Figures.Parallelogram:  
 {  
 int a, h;  
 Console.WriteLine("Введите длину основания параллелограмма:");  
 a = Convert.ToInt32(Console.ReadLine());  
 Console.WriteLine("Введите высоту параллелограмма:");  
 h = Convert.ToInt32(Console.ReadLine());  
 res = a \* h;  
 break;  
 }  
 case Figures.Triangle:  
 {  
 int b, h;  
 Console.WriteLine("Введите длину основания треугольника:");  
 b = Convert.ToInt32(Console.ReadLine());  
 Console.WriteLine("Введите высоту ромба:");  
 h = Convert.ToInt32(Console.ReadLine());  
 res = 0.5 \* b \* h;  
 break;  
 }  
 case Figures.Square:  
 {  
 int a;  
 Console.WriteLine("Введите длину стороны квадрата:");  
 a = Convert.ToInt32(Console.ReadLine());  
 res = a \* a;  
 break;  
 }  
 case Figures.Rhombus:  
 {  
 int a, h;  
 Console.WriteLine("Введите длину основание ромба:");  
 a = Convert.ToInt32(Console.ReadLine());  
 Console.WriteLine("Введите высоту ромба:");  
 h = Convert.ToInt32(Console.ReadLine());  
 res = a \* h;  
 break;  
 }  
 case Figures.Rectangle:  
 {  
 int a, b;  
 Console.WriteLine("Введите 1 сторону прямоугольника:");  
 a = Convert.ToInt32(Console.ReadLine());  
 Console.WriteLine("Введите 2 сторону прямоугольника:");  
 b = Convert.ToInt32(Console.ReadLine());  
 res = a \* b;  
 break;  
 }  
 case Figures.Circle:  
 {  
 int r;  
 Console.WriteLine("Введите радиус круга:");  
 r = Convert.ToInt32(Console.ReadLine());  
 res = Math.PI \* (r \* r);  
 break;  
 }  
 default:  
 {  
 res = -1;  
 break;  
 }  
 }  
  
 return res;  
 }  
  
 static void Main()  
 {  
 Fdata fd1 = new Fdata(1, 1, 0x2727FF, Figures.Rhombus);  
 Fdata fd2 = new Fdata(1, 1, 0x2727FF, Figures.Circle);  
 ShowFdataObj(fd1);  
 ShowFdataObj(fd2);  
 Console.WriteLine(Area(fd1));  
 Console.WriteLine(Area(fd2));  
 Cirlce cr = new Cirlce(5, 10, 30, "Red");  
 cr.PrintInfo();  
 Cirlce[] data = new Cirlce[5];  
 Random rnd = new Random();  
 for (int i = 0; i < 5; i++)  
 {  
 data[i] = new Cirlce(rnd.Next(1,6), rnd.Next(5,54), rnd.Next(5,20), "Red");  
 data[i].PrintInfo();  
 }  
 }  
 }  
}