|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

**Отчёт**

***Лабораторная работа №2***

Выполнил студент: \_\_***Сафиуллин Антон Павлович\_\_\_\_\_\_\_\_\_\_***

*фамилия, имя, отчество*

Группа: \_\_\_\_***РТ5-31Б\_\_\_\_\_\_\_\_***

**описание задания**

Лабораторная работа №2

Разработать программу, реализующую работу с классами.

1. Программа должна быть разработана в виде консольного приложения на языке C#.

2. Абстрактный класс «Геометрическая фигура» содержит виртуальный метод для вычисления площади фигуры.

3. Класс «Прямоугольник» наследуется от «Геометрическая фигура». Ширина и высота объявляются как свойства (property). Класс должен содержать конструктор по параметрам «ширина» и «высота».

4. Класс «Квадрат» наследуется от «Прямоугольник». Класс должен содержать конструктор по длине стороны.

5. Класс «Круг» наследуется от «Геометрическая фигура». Радиус объявляется как свойство (property). Класс должен содержать конструктор по параметру «радиус».

6. Для классов «Прямоугольник», «Квадрат», «Круг» переопределить виртуальный метод Object.ToString(), который возвращает в виде строки основные параметры фигуры и ее площадь.

7. Разработать интерфейс IPrint. Интерфейс содержит метод Print(), который не принимает параметров и возвращает void. Для классов «Прямоугольник», «Квадрат», «Круг» реализовать наследование от интерфейса IPrint. Переопределяемый метод Print() выводит на консоль информацию, возвращаемую переопределенным методом ToString().

**текст программы**

using System;

using System.Collections.Specialized;

using System.ComponentModel.DataAnnotations;

using System.Runtime.InteropServices.ComTypes;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Dynamic;

namespace LabRabota\_2

{

class Program

{

public static void cout(string per)

{

Console.WriteLine(per);

}

public static double dcin()

{

int error = 1;

double pev;

while (error == 1)//Вввод переменной

{

try

{

string per = Console.ReadLine();

pev = double.Parse(per);

return pev;

}

catch

{

cout("Введите корректную переменную: ");

}

finally { }

}

return 0;

}

static void Main(string[] args)

{

cout("Программу создал Сафиуллин А.П. РТ5-31Б");

Rectangle KK;

Sqare TT;

TT = new Sqare(4);

KK = new Rectangle(5, 4);

IPrint.Print(TT);

IPrint.Print(KK);

Circle circle1;

circle1 = new Circle(5);

IPrint.Print(circle1);

}

}

interface IPrint

{

private static void cout(string per)

{

Console.WriteLine(per);

}

public static void Print(object T)

{

cout(T.ToString());

}

}

abstract class GeomFigure

{

protected double Area;

protected string Name;

public static void cout(string per)

{

Console.WriteLine(per);

}

public static double dcin()

{

int error = 1;

double pev;

while (error == 1)//Вввод переменной

{

try

{

string per = Console.ReadLine();

pev = double.Parse(per);

return pev;

}

catch

{

cout("Введите корректную переменную: ");

}

finally { }

}

return 0;

}

public GeomFigure()

{

}

}

class Rectangle : GeomFigure,IPrint

{

protected double Width;

protected double Length;

protected Rectangle()

{

Area = Width \* Length;

}

public Rectangle(double W, double L)

{

Width = W;

Length = L;

Area = Width \* Length;

Name = "Прямоугольник";

}

public void Display()

{

cout($"Ширина фигуры {Name} равна:{Width}");

cout($"Длина фигуры {Name} равна:{Length}");

cout($"Площадь фигуры {Name} равна:{Area}");

}

public override string ToString()

{

if (Name == "Прямоугольник") return $"Фигура имеет название: " + Name.ToString() + "\nШирина фигуры: " + Width.ToString() + "\nДлина фигуры: " + Length.ToString() + "\nПлощадь фигуры: " + Area.ToString();

else return "";

}

}

class Sqare: Rectangle, IPrint

{

public Sqare(double W) : base(W,W)

{

Name = "Квадрат";

}

public override string ToString()

{

return $"Фигура имеет название: " + Name.ToString() + "\nШирина && Длина фигуры: " + Width.ToString() + "\nПлощадь фигуры: " + Area.ToString();

}

}

class Circle: GeomFigure, IPrint

{

private double radius;

protected string newArea;

public Circle(double R)

{

radius = R;

newArea = $"{radius \* radius}\*Pi";

Name = "Круг";

}

public double Radius

{

get

{

return radius;

}

}

public string name

{

get

{

return Name;

}

}

public void display()

{

cout($"Радиус фигуры {Name}: {radius}");

}

public override string ToString()

{

return $"Фигура имеет название: " + Name.ToString() + "\nРадиус фигуры: " + Radius.ToString() + "\nПлощадь фигуры: " + newArea.ToString();

}

}

}

**экранные формы с примерами выполнения программы**

