|  |  |  |
| --- | --- | --- |
| Защищено:  Гапанюк Ю. Е.  "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |  | Демонстрация:  Гапанюк Ю. Е.  "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |

**Отчет по лабораторной работе №2 по курсу**

**«Парадигмы и конструкции языков программирования»**

#### Тема работы: "Изучение языка C#"

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: | Агапова Анна Денисовна |
| студент группы  ИУ5Ц-52Б | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | (подпись) |
| Гапанюк Ю.Е. | "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |

Москва, МГТУ 2024

**СОДЕРЖАНИЕ**

[Цель лабораторной работы 3](#_Toc177564394)

[Практическое задание 3](#_Toc177564395)

[Листинг программы 4](#_Toc177564396)

[Результат работы программы 6](#_Toc177564397)

[Ответы на контрольные вопросы 7](#_Toc177564398)

[1) В чем разница между ключевыми словами «override» и «new» при переопределении виртуального метода? 7](#_Toc177564399)

[2) Как переопределить виртуальный метод? 7](#_Toc177564400)

[3) Как реализуется наследование класса от класса? 7](#_Toc177564401)

[4) Как объявить конструктор класса в С#? 7](#_Toc177564402)

[5) Как из конструктора класса вызвать конструктор базового класса? 7](#_Toc177564403)

[6) Что такое свойства и для чего они используются? 8](#_Toc177564404)

[7) Что такое опорная переменная свойства? 8](#_Toc177564405)

[8) Что такое абстрактный класс? 8](#_Toc177564406)

## Цель лабораторной работы

Получение практических навыков по C#

## Практическое задание

## Листинг программы

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Linq;

using System.Text;

namespace LR\_1

{

interface IPrint

{

void Print();

}

abstract class Figure // Класс фигура

{

public string Type { get; protected set; } // Тип фигуры (объявление автоопределяемым свойством)

string \_Type;

public abstract double Area(); // Вычисление площади

public override string ToString()

{

return this.Type + " площадью " + this.Area().ToString("f2");

}

}

class Rectangle : Figure, IPrint

{

double height; //высота

double width; //ширина

public double Width

{

get => this.width;

set => this.width = value;

}

public double Height

{

get => this.height;

set => this.height = value;

}

public Rectangle(double h, double w)

{

this.height = h;

this.width = w;

this.Type = "Прямоугольник";

}

public override double Area()

{

double result = this.height\* this.width;

return result;

}

public void Print()

{

Console.WriteLine(this.ToString());

}

}

class Square : Rectangle, IPrint

{

public Square(double d) : base(d,d)

{

this.Type = "Квадрат";

}

public void Print()

{

Console.WriteLine(this.ToString());

}

}

class Circle : Figure, IPrint

{

double radius;

public Circle(double r)

{

this.radius = r;

this.Type = "Круг";

}

public override double Area()

{

double result = Math.PI\* this.radius \* this.radius;

return result;

}

public void Print()

{

Console.WriteLine(this.ToString());

}

}

class Program

{

static void Main(string[] args)

{

Rectangle rectangle = new Rectangle(2, 3);

Square square = new Square(4);

Circle circle = new Circle(5);

rectangle.Print();

square.Print();

circle.Print();

}

}

}

## Результат работы программы

Прямоугольник площадью 6,00

Квадрат площадью 16,00

Круг площадью 78,54

## Ответы на контрольные вопросы

### В чем разница между ключевыми словами «override» и «new» при переопределении виртуального метода?

ключевое слово override используется для переопределения виртуального метода.

Таким образом, компилятор требует, чтобы программист явно указал

ключевое слово override, если он считает, что данный метод следует

использовать как виртуальный. Если программист хочет показать что

данный метод не должен применяться как виртуальный, но имеет имя,

совпадающее с именем виртуального метода, то он должен указать

ключевое слово new.

### Как переопределить виртуальный метод?

Поэтому в языке C# виртуальные методы должны быть явно

объявлены с помощью ключевых слов virtual или abstract (абстрактные

методы – аналог чистых виртуальных методов в С++).

### Как реализуется наследование класса от класса?

В языке C# наследовать класс можно только от одного класса.

Синтаксис наследования в языке C# аналогичен синтаксису языка

C++. Для обозначения наследования используется символ двоеточия при

объявлении класса, а затем указывается имя базового класса «class

ExtendedClass1 : BaseClass».

### Как объявить конструктор класса в С#?

public BaseClass(int param) { this.i = param; }

Имя конструктора совпадает с именем класса. Конструктор принимает

один параметр и присваивает его переменной класса, доступ к которой

выполняется с помощью ключевого слова this.

### Как из конструктора класса вызвать конструктор базового класса?

Наследуемый класс имеет доступ к protected полям базового

класса, но не имеет доступа к private полям, их можно инициализировать с помощью вызова конструктора базового класса:

public ExtendedClass1(int pi, int pi2) : base(pi)

{

i2 = pi2;

}

Ключевое слово base обозначает вызов конструктора базового класса.

### Что такое свойства и для чего они используются?

Свойство - это специальный вид конструкции языка.

Пример объявления простого свойства:

//private-значение, которое хранит данные для свойства

private int \_property1 = 0;

//объявление свойства

public int property1

{

//возвращаемое значение

get { return \_property1; }

//установка значения, value - ключевое слово

set { \_property1 = value; }

}

### Что такое опорная переменная свойства?

Переменная \_property1 является закрытой (private) и содержит данные

для свойства. Такую переменную принято называть опорной переменной

свойства. Как правило, опорные переменные всегда имеют область

видимости private или protected, чтобы к ним не было внешнего доступа.

Как задаются области видимости для свойств и аксессоров?

прописываются внутри свойства перед аксессором

### Что такое абстрактный класс?

abstract class Figure

{

public string Type { get; set; }

public abstract double Area();

public override string ToString()

{

return this.Type + " площадью " + this.Area().ToString();

}

}