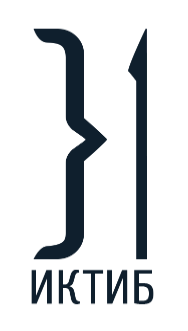
МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное учреждения высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт компьютерных технологий и информационной безопасности

Кафедра математического обеспечения и применения ЭВМ



**ЛАБОРАТОРНАЯ РАБОТА №5**

по дисциплине

**«Объектно-ориентированное программирование»**

на тему:

**«Классы, объекты, наследование в С#»**

Вариант №8

Выполнили:

Студенты группы

КТбо2-6

|  |  |  |
| --- | --- | --- |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Беликова Ю. А. |
|  | *подпись* |  |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Чепурная Т.В. |
|  | *подпись* |  |

Проверил:

ассистент кафедры

МОП ЭВМ

|  |  |  |
| --- | --- | --- |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Чичерина К. С. |
|  | *подпись* |  |

Оценка

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

# Задание

Создать класс Function (функция) с методами вычисления значения функции y = f(x) в заданной точке х и вывода результата на консоль. На его основе определить классы Ellipse, Hiperbola и Parabola, в которых реализуются соответствующие математические зависимости. В дополнительном классе Series все три функции должны вызываться для заданного интервала изменения x с выводом результатов.

# Спецификация класса

В программе присутствует 7 классов в пространстве имен Lab5.

1. Базовый класс Function

Поля класса:

protected double a = 0; - 1-й коэффициент

protected double x = 0; - значение x

protected double y = 0; - значение y

Конструкторы:

public Function(); - конструктор по умолчанию

public Function(double \_a, double \_x); - конструктор с параметрами

Метод:

abstract public void FunctionValue(); - абстрактный метод;

1. Производный класс Ellipse

Поля класса:

double b; - 2- ой коэффициент

Конструктор:

public Ellipse(double \_a, double \_x, double \_b) : base (\_a, \_x) - конструктор с параметрами

Метод:

public override void FunctionValue() - метод вычисления функции Эллипса

1. Производный класс Hiperbola

Конструктор:

public Hiperbola(double \_a, double \_x) : base(\_a, \_x) { } - конструктор с параметрами

Метод:

public override void FunctionValue() - метод вычисления функции Гиперболы

1. Производный класс Parabola

Поля класса:

double c = 0; - 3- ий коэффициент

double b = 0; - 2- ой коэффициент

Конструктор:

public Parabola(double \_a, double \_x, double \_b, double \_c) :base (\_a, \_x) - конструктор с параметрами

Метод:

public override void FunctionValue(); - метод вычисления функции Параболы

1. Дополнительный класс Series

Поля класса:

int \_start; - интервал

int \_finish; - интервал

Конструктор:

public Series(int start, int finish); - конструктор с параметрами

1. Класс ConsoleInteractor

public void Run();

# Используемые математические зависимости и алгоритмы

Вычисление значения функции гиперболы в заданной точке x, где a – коэффициент:

y = a / x;

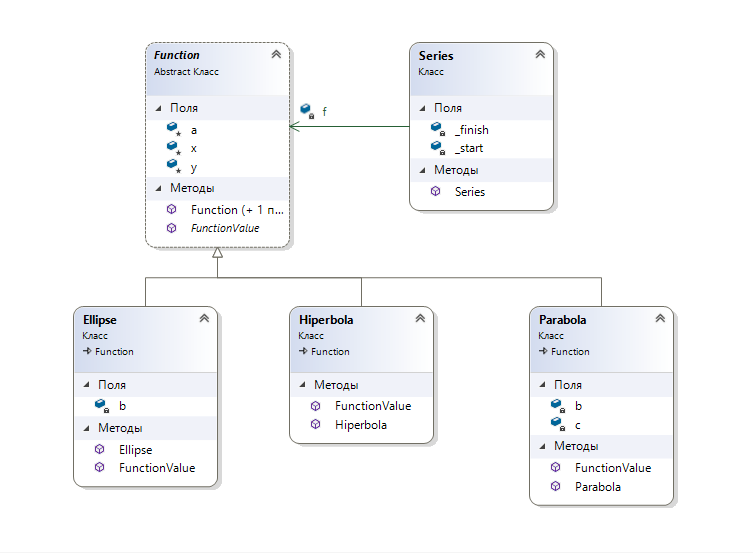
Вычисление значения функции параболы в заданной точке x, где a, b, c ­– коэффициенты:

y = a \* x \* x + b \* x + c;

Вычисление значения функции эллипс в заданной точке x, где a, b ­– коэффициенты:

y = (b \* (sqrt(abs(1 - ((x \* x) / (a \* a))))));

# Диаграмма классов



1. Диаграмма

# Листинг программы

**Function.cs**

namespace lab5

{

abstract class Function

{

protected double a = 0;

protected double x = 0;

protected double y = 0;

public Function() { } // Нужен для создания массива в классе Serias

public Function(double \_a, double \_x) // Конструктор

{

a = \_a;

x = \_x;

}

abstract public void FunctionValue();

}

}

**Ellipse.cs**

using System;

namespace lab5

{

class Ellipse: Function

{

double b;

public Ellipse(double \_a, double \_x, double \_b) : base (\_a, \_x)

{

b = \_b;

}

public override void FunctionValue()

{

try

{

if (a == 0)

{

throw new Exception ("На 0 делить нельзя!");

}

y = (b \* (Math.Sqrt(Math.Abs(1 - ((x \* x) / (a \* a))))));

Console.WriteLine($"Эллипс: x= {x} y= {y}");

}

catch (Exception exc)

{

Console.WriteLine($"Эллипс: {exc.Message}");

}

}

}

}

**Parabola.cs**

using System;

namespace lab5

{

class Parabola : Function

{

double c = 0;

double b = 0;

public Parabola(double \_a, double \_x, double \_b, double \_c) :base (\_a, \_x)

{

b = \_b;

c = \_c;

}

public override void FunctionValue()

{

try

{

if (a == 0)

{

throw new Exception("Первый коэффициент не может равняться 0 !");

}

y = a \* x \* x + b \* x + c;

Console.WriteLine($"Парабола: x= {x} y= {y}");

}

catch (Exception exc)

{

Console.WriteLine($"Парабола: {exc.Message}");

}

}

}

}

**Hiperbola.cs**

using System;

namespace lab5

{

class Hiperbola : Function

{

public Hiperbola(double \_a, double \_x) : base(\_a, \_x) { }

public override void FunctionValue()

{

try

{

if (x == 0)

{

throw new Exception("На 0 делить нельзя!");

}

y = a / x;

Console.WriteLine($"Гипербола: x= {x} y= {y}");

}

catch (Exception exc)

{

Console.WriteLine($"Гипербола: {exc.Message}");

}

}

}

}

**ConsoleInteractor.cs**

using System;

namespace lab5

{

class ConsoleInteractor

{

public void Run()

{

int start, finish, flag = 1;

while (flag == 1)

{

Console.WriteLine("Введите интервал");

start = Convert.ToInt32(Console.ReadLine());

finish = Convert.ToInt32(Console.ReadLine());

if (finish < start)

{

int temp1 = finish;

finish = start;

start = temp1;

}

else if (finish == start)

{

Console.WriteLine("Некорректный интервал. Если хотите ввести интервал заново нажмите 1. Если завершить - 0.");

flag = Convert.ToInt32(Console.ReadLine());

}

if (finish > start)

{

Series s = new(start, finish);

Console.WriteLine("Если хотите продолжить нажмите 1. Если завершить - 0.");

flag = Convert.ToInt32(Console.ReadLine());

}

}

}

}

}

**Program.cs**

namespace lab5

{

class Program

{

static void Main(string[] args)

{

ConsoleInteractor unit = new ConsoleInteractor();

unit.Run();

}

}

}

**Series.cs**

using System;

namespace lab5

{

class Series

{

int \_start;

int \_finish;

Function[] f;

public Series(int start, int finish)

{

\_start = start;

\_finish = finish;

f = new Function[(finish - start + 1) \* 3];//массив указателей на обЪекты Function

int a, b, c, a1, b1, a2; // коэффициенты

Console.WriteLine("Введите коэффициенты для фунции Эллипс:");

a1 = Convert.ToInt32(Console.ReadLine());

b1 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine($"y={b1}\*sqrt(1-(x^2/{a1}^2)");

Console.WriteLine("Введите коэффициенты для фунции Парабола:");

a2 = Convert.ToInt32(Console.ReadLine());

b = Convert.ToInt32(Console.ReadLine());

c = Convert.ToInt32(Console.ReadLine());

Console.WriteLine($"y={a2}x^2 + {b}x + {c}");

Console.WriteLine("Введите коэффициент для фунции Гипербола:");

a = Convert.ToInt32(Console.ReadLine());

Console.WriteLine($"y={a}/x \n");

for (int i = 0; i<(finish - start + 1) \* 3; i++)

{

Console.WriteLine();

f[i] = new Ellipse(a1, \_start, b1);

f[i].FunctionValue();

i++;

f[i] = new Parabola(a2, \_start, b, c);

f[i].FunctionValue();

i++;

f[i] = new Hiperbola(a, \_start);

f[i].FunctionValue();

Console.WriteLine();

\_start++;

}

}

}

}

# Набор тестов, охватывающих все режимы работы программы

Изображение выглядит как текст

Автоматически созданное описание

1. Тест 1

Изображение выглядит как текст

Автоматически созданное описание

1. Тест 2

Изображение выглядит как текст

Автоматически созданное описание

1. Тест 3

Изображение выглядит как текст

Автоматически созданное описание

1. Тест 4

Изображение выглядит как текст

Автоматически созданное описание

1. Тест 5