**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**Учреждение образования**

**«Гомельский государственный технический университет»**

**имени П. О. Сухого**

Кафедра: «**Информатика**»

По курсу: «**Языки программирования высокого уровня**»

Лабораторная работа №2

«**Понятие класса. Основные элементы класса**»

Выполнил: студент группы ИП-21 Ивановский Алексей Михайлович

Проверил: старший преподаватель Романькова Татьяна Леонидовна

Гомель 2021

# **Цель работы**

Получить навыки использования методов класса Math, научиться создавать классы, содержащие поля и методы, конструкторы, свойства, научиться создавать и использовать объекты класса.

# **Задание 1**

Создать класс «**Функция**», описывающий объекты - функции из семейства функций заданного вида (вид функции задан в таблице 1). Класс должен содержать следующие элементы:

* Закрытые поля для хранения значения параметров функции a и b.
* Свойства для доступа к параметрам функции.
* Метод для вычисления значения функции (входной параметр – значение аргумента х).
* Метод для табулирования функции (входные параметры - хнач, хкон, шаг ∆х). При хнач <= хкон шаг прибавлять, при хнач > хкон шаг вычитать.
* Метод **ToString()**, результатом которого является строковое представление функции.
* Статический метод для ввода начального значения аргумента, конечного значения аргумента и шага изменения аргумента функции (выходные параметры - хнач, хкон, шаг ∆х) Ввод продолжать до тех пор пока не будет введен шаг **∆х>= 0**.

Создать класс «**Таблица**», описывающий объекты-таблицы из двух столбцов, содержащий следующие элементы:

* Закрытые поля: заголовок таблицы, заголовки столбцов, ширина первого столбца, ширина второго столбца.
* Конструктор с параметрами.
* Метод для вывода шапки таблицы.
* Метод для вывода строки таблицы (входные параметры – числовые значения, которые выводятся в строке таблицы).
* Метод для вывода низа таблицы.

Разработать программу, которая выполняет следующие действия:

* + Создает два объекта класса «**Функция**» (параметры функций вводятся с клавиатуры).
  + Для каждого объекта вычисляет значения для трех различных значений аргумента.
  + Для каждого объекта–функции выполняет табулирование для ряда значений аргумента. Результат выводить в виде таблицы, в заголовке таблицы приводить вид функции.

|  |  |
| --- | --- |
| 11 |  |

**Код программы:**

**Основной файл:**

using System;

using static System.Console;

namespace Lab\_2\_1

{

class Program

{

static void Main()

{

double xStart = 0;

double xFinish = 0;

double xDelta = 0;

double a = 0;

double b = 0;

Function fun = new Function();

Function fun2 = new Function();

Table table = new Table();

WriteLine("Введите значения для первого объекта первый раз");

Function.InputValues(ref xStart, ref xFinish, ref xDelta, ref a, ref b, ref fun);

WriteLine("Первая таблица");

table.ShowTable(in fun, 20, 45);

ReadKey();

Clear();

WriteLine("Введите значения для первого объекта второй раз");

Function.InputValues(ref xStart, ref xFinish, ref xDelta, ref a, ref b, ref fun);

WriteLine("Вторая таблица");

table.ShowTable(in fun, 20, 45);

ReadKey();

Clear();

WriteLine("Введите значения для первого объекта третий раз");

Function.InputValues(ref xStart, ref xFinish, ref xDelta, ref a, ref b, ref fun);

WriteLine("Третья таблица");

table.ShowTable(in fun, 20, 45);

ReadKey();

Clear();

WriteLine("Введите значения для второго объекта первый раз");

Function.InputValues(ref xStart, ref xFinish, ref xDelta, ref a, ref b, ref fun2);

WriteLine("Первая таблица");

table.ShowTable(in fun2, 20, 45);

ReadKey();

Clear();

WriteLine("Введите значения для второго объекта второй раз");

Function.InputValues(ref xStart, ref xFinish, ref xDelta, ref a, ref b, ref fun2);

WriteLine("Вторая таблица");

table.ShowTable(in fun2, 20, 45);

ReadKey();

Clear();

WriteLine("Введите значения для второго объекта третий раз");

Function.InputValues(ref xStart, ref xFinish, ref xDelta, ref a, ref b, ref fun2);

WriteLine("Третья таблица");

table.ShowTable(in fun2, 20, 45);

}

}

}

**Файл с функциями:**

using System;

using static System.Console;

using static System.Math;

using static System.Convert;

namespace Lab\_2\_1

{

class Function

{

private double a;

private double b;

private double xStart = 0;

private double xFinish = 0;

private double xDelta = 0;

private int howMuchStep = 0;

public double GetA() => a;

public double GetB() => b;

public int GetHowMuchStep() => howMuchStep;

private void SetHowMuchStep(int howMuchStep) => this.howMuchStep = howMuchStep;

private void SetA(double a) => this.a = a;

private void SetB(double b) => this.b = b;

private void SetXstart(double xStart) => this.xStart = xStart;

private void SetXfinish(double xFinish) => this.xFinish = xFinish;

private void SetXdelta(double xDelta) => this.xDelta = xDelta;

private void HowMuchSteps(double xStart, double xFinish, double xDelta)

{

if (xStart > xFinish)

SetHowMuchStep((int)(xStart - xFinish) / (int)xDelta);

if (xStart <= xFinish)

SetHowMuchStep((int)(xFinish - xStart) / (int)xDelta);

}

public double GetFunction(double x)

{

return Pow(Cos(a), 2) + Tan(2 \* x) + Abs(b);

}

public double GetX() => xStart;

public void CountStep()

{

if (xStart <= xFinish)

xStart += xDelta;

if (xStart > xFinish)

xStart -= xDelta;

}

public override string ToString() => $"cos({a:f2})^2 + tg(2x) + |{b:f2}|";

public static void InputValues(ref double xStart, ref double xFinish, ref double xDelta, ref double a, ref double b, ref Function fun)

{

short stage = 1;

while (true)

{

try

{

if (stage == 1)

{

WriteLine("Введите начальное значение X");

xStart = ToDouble(ReadLine());

fun.SetXstart(xStart);

WriteLine("Введите конечное значение X");

xFinish = ToDouble(ReadLine());

if (xStart == xFinish)

throw new MyException("Ошибка: начальное значение X равно конечному");

fun.SetXfinish(xFinish);

stage = 2;

Clear();

}

if (stage == 2)

{

WriteLine("Введите шаг изменения X");

xDelta = ToDouble(ReadLine());

if (xDelta <= 0)

throw new MyException("Ошибка: шаг изменения меньше или равен нулю");

fun.SetXdelta(xDelta);

stage = 3;

Clear();

}

if (stage == 3)

{

WriteLine("Введите значение A");

a = ToDouble(ReadLine());

fun.SetA(a);

stage = 4;

Clear();

}

if (stage == 4)

{

WriteLine("Введите значение B");

b = ToDouble(ReadLine());

fun.SetB(b);

Clear();

}

}

catch (MyException ex)

{

Clear();

WriteLine(ex.Message);

continue;

}

catch (Exception)

{

Clear();

WriteLine("Ошибка: введено не численное значение");

continue;

}

break;

}

fun.HowMuchSteps(xStart, xFinish, xDelta);

}

}

}

**Файл с табличкой:**

using System;

using static System.Console;

namespace Lab\_2\_1

{

class Table

{

private int howMuchStep;

private string msg;

public void ShowTable(in Function fun, short leftPartWidth, short rigthPartWidth)

{

howMuchStep = fun.GetHowMuchStep();

for (int i = 0; i < leftPartWidth + rigthPartWidth + 1; i++)

{

if (i == 0)

Write("╔");

if (i < leftPartWidth || i > leftPartWidth)

Write("═");

if (i == leftPartWidth)

Write("╦");

if (i == leftPartWidth + rigthPartWidth)

Write("╗");

}

WriteLine();

msg = "Y = " + fun.ToString();

int length = msg.Length;

for (int i = 0; i < leftPartWidth + rigthPartWidth + 4; i++)

{

if (i <= leftPartWidth + 1)

{

if (i == leftPartWidth + 1)

Write("║");

if (i == 0)

Write("║");

if (i == leftPartWidth / 2 - 1)

Write("X");

else if (i < leftPartWidth)

Write(" ");

}

else

{

if (i == leftPartWidth + rigthPartWidth + 3)

Write("║");

if (i == leftPartWidth + (rigthPartWidth - length) / 2 + 2)

{

Write(msg);

i += length;

}

else if (i <= leftPartWidth + rigthPartWidth + 2)

Write(" ");

}

}

WriteLine();

for (int i = 0; i < leftPartWidth + rigthPartWidth + 1; i++)

{

if (i == 0)

Write("╠");

if (i < leftPartWidth || i > leftPartWidth)

Write("═");

if (i == leftPartWidth)

Write("╬");

if (i == leftPartWidth + rigthPartWidth)

Write("╣");

}

WriteLine();

string str = "";

string str2 = "";

int length2 = 0;

for (int i = 0; i <= howMuchStep; i++)

{

str = "";

str2 = "";

str += fun.GetX();

str2 += fun.GetFunction(fun.GetX());

length = str.Length;

length2 = str2.Length;

for (int j = 0; j < leftPartWidth + rigthPartWidth + 4; j++)

{

if (j < leftPartWidth + 2)

{

if (j == (leftPartWidth - length) / 2 + 1)

{

Write(str);

j += length;

}

if (j == 0)

Write("║");

else if (j < leftPartWidth + 1)

Write(" ");

else if (j == leftPartWidth + 1)

Write("║");

}

else

{

if (j == leftPartWidth + (rigthPartWidth - length2) / 2 + 1)

{

Write(str2);

j += length2;

}

if (j == rigthPartWidth + leftPartWidth + 2)

Write("║");

else if (j < leftPartWidth + rigthPartWidth + 2)

Write(" ");

}

}

fun.CountStep();

WriteLine();

}

for (int i = 0; i < leftPartWidth + rigthPartWidth + 1; i++)

{

if (i == 0)

Write("╚");

if (i < leftPartWidth || i > leftPartWidth)

Write("═");

if (i == leftPartWidth)

Write("╩");

if (i == leftPartWidth + rigthPartWidth)

Write("╝");

}

}

}

}

**Файл с собственным классом исключений:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_2\_1

{

class MyException : Exception

{

public MyException(string msg) : base(msg) { }

}

}

# **Тесты**

Изображение выглядит как текст

Автоматически созданное описание

# **Вывод**

Я получил навыки использования методов класса Math, научился создавать классы, содержащие поля и методы, конструкторы, свойства, научился создавать и использовать объекты класса.