Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра Информатики

Дисциплина «Программирование»

**ОТЧЕТ**

к лабораторной работе №3

на тему:

**«Классы математических функций, классы для работы с числами и датами»**

БГУИР 6-05-0612-02 67

|  |
| --- |
| Выполнил студент группы 353503  КОХАН Артём Игоревич |
|  |
| (дата, подпись студента) |
| Проверил ассистент каф. Информатики  РОМАНЮК Максим Валерьевич |
|  |
| (дата, подпись преподавателя) |

Минск 2024

# 1 Индивидуальное задание

**Задание 1. Вариант 7.** Описать класс (с произвольным именем) в отдельном файле проекта. В созданном классе разработать метод, который нечётное число заменяет на 0, а чётное число уменьшает в два раза.

**Задание 2. Вариант 3.** Добавить в проект папку Services. В созданной папке описать класс (с произвольным именем). В созданном классе описать метод, вычисляющий значение функции в соответствии с индивидуальным вариантом. Метод должен вывести в консоль результат выполнения и номер ветки, по которой производилось вычисление (см. рисунок 1).



Рисунок 1 – Заданная функция

**Задание 3.** В отдельном файле описать класс DateService. Класс должен предоставлять следующие методы: Определять день недели произвольной даты, передаваемой методу в качестве строкового параметра (например GetDay(string date)). Определять, сколько дней пройдет между текущей датой и датой, передаваемой методу в качестве числовых параметров: день, месяц, год (например GetDaysSpan(int day, int month, int year)).

# 2 Выполнение работы

Для первого задания был добавлен новый файл в проект с расширением .cs. Внутри этого файла был создан namespace с таким же именованием, как и у основного файла программы. Создан static класс MyMath и static метод с модификатором доступа public. Метод принимает на вход целое число, если оно чётное, то метод, c помощью тернарного оператора, вернёт число в два раза меньше, а если нечётное, то 0.

namespace Task1

{

static public class MyMath

{

static public long Operation(long numberValue)

{

return (numberValue % 2 == 0) ? numberValue / 2 : 0;

}

}

}

Рассмотрим работу данного метода в файле Program.cs, где в метод MyMath.Operation мы передаём число типа данных long.

namespace Task1

{

internal class Program

{

static void Main(string[] args)

{

bool replay = true;

while (replay)

{

string? str = null;

long number = 0;

while (true)

{

Console.WriteLine("Input integer number:");

str = Console.ReadLine();

if (long.TryParse(str, out number))

{

number = long.Parse(str);

break;

}

}

Console.WriteLine("Number later: " + MyMath.Operation(number));

Console.WriteLine("Input '+' to end or another symbol to continue");

str = Console.ReadLine();

if (str == "+") replay = false;

}

}

}

}

Результат работы программы продемонстрирован ниже (см. рисунок 2).

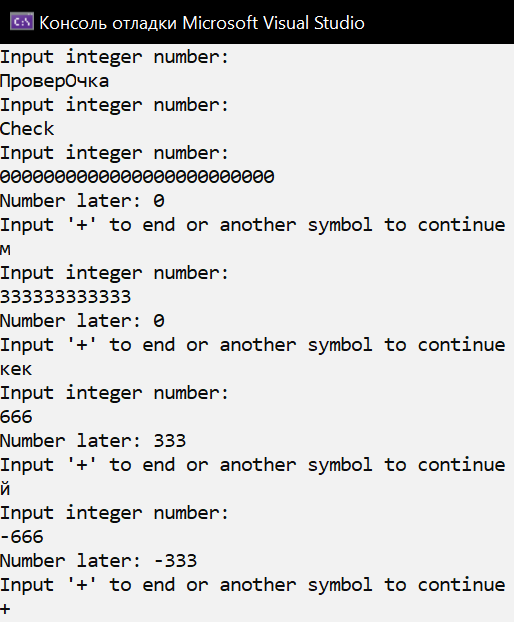


Рисунок 2 – Результат работы программы первого задания

Для второго задания была создана папка Services в проекте и в этой папке создан файл MyMath.cs. В файле создаётся уже namespace под названием Task2.Services и статический класс MyMath в котором создаём статический метод Foo для вычисления значения функции y(x). Следует отметить, что x определяется в зависимости от значений z и b.

namespace Task2.Services

{

static public class MyMath

{

static public double Foo(double z, double b)

{

double x = (z < 1) ? z / b : Math.Sqrt(z \* z \* z \* b \* b \* b);

Console.WriteLine((z < 1) ? $"On the first branch, because z < 1, x = {x}" :

$"On the second branch, because z >= 1, x = {x}");

return -Math.PI + Math.Cos(x \* x \* x) \* Math.Cos(x \* x \* x) + Math.Sin(x \* x) \*

Math.Sin(x \* x) \* Math.Sin(x \* x);

}

}

}

Рассмотрим работу данного метода в файле Program.cs, где в метод MyMath.Foo мы передаём числа типа данных double. Проверка на ввод для z и b реализована в отдельной static функции CorrectInput.

namespace Task2

{

internal class Program

{

static void CorrectInput(out double value)

{

string? str = null;

while (true)

{

str = Console.ReadLine();

if (double.TryParse(str, out value))

{

value = double.Parse(str);

break;

}

Console.WriteLine("re-Input value: ");

}

}

static void Main(string[] args)

{

bool replay = true;

while (replay)

{

string? str = null;

double z = 0, b = 0;

Console.WriteLine("Input z:");

CorrectInput(out z);

Console.WriteLine("Input b:");

CorrectInput(out b);

Console.WriteLine((b == 0 && z < 1)? "Indefinite" :

"y = " + Services.MyMath.Foo(z, b));

Console.WriteLine("Input '+' to end or another symbol to continue");

str = Console.ReadLine();

if (str == "+") replay = false;

}

}

}

}

Результат работы программы продемонстрирован ниже (см. рисунок 3).

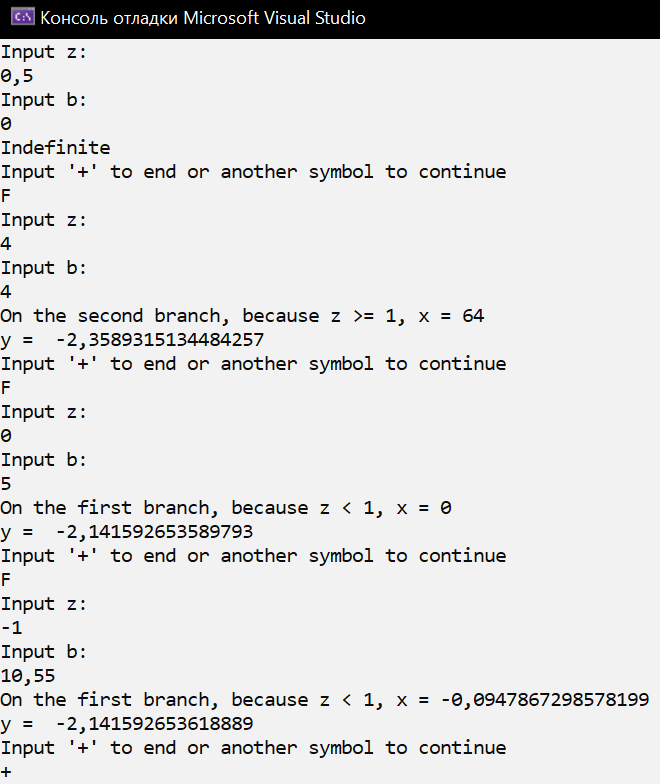


Рисунок 3 – Результат работы программы второго задания

В коде третьего задания при помощи класса DateTime и TimeSpan были написаны методы для выявления дня недели по введённой дате и нахождения разницы в между сегодняшней датой и введённой датой. Следует отметить, что в методе GetDay имеется проверка на ввод, которая проверяет, может ли строка быть преобразована к DateTime.

namespace Task3

{

static public class DateService

{

static public string GetDay(string? date)

{

DateTime dateTime = new DateTime();

while (true)

{

Console.WriteLine("Input date YYYY-MM-DD:");

date = Console.ReadLine();

if(DateTime.TryParse(date, out dateTime))

{

dateTime = DateTime.Parse(date);

break;

}

}

return dateTime.DayOfWeek.ToString();

}

static public int GetDaysSpan(int day, int month, int year)

{

DateTime currentDate = DateTime.Now;

DateTime inputDate = new DateTime(year, month, day);

TimeSpan timeSpan = inputDate - currentDate;

return Math.Abs(timeSpan.Days);

}

}

}

Рассмотрим работу методов в файле Program.cs, где в метод DateService.GetDay мы передаём тип данных string, а в метод DateService.GetDaysSpan передаём целочисленные значения дня, месяца, года. Чтобы не позволить ввести пользователю некорректную дату, будем проверять ввод с помощью static функций CheckDate и CorrectInput. Считаю удобным будет воспользоваться оператором выбора switch и запрашивать у пользователя каким методом он хочет воспользоваться.

namespace Task3

{

internal class Program

{

static bool CheckDate(int dayValue, int monthValue, int yearValue)

{

if (dayValue < 1 || dayValue > 31 || monthValue < 1 || monthValue > 12

|| yearValue < 1) return false;

else if (yearValue % 4 != 0 && monthValue == 2 && dayValue > 28) return false;

else if (yearValue % 4 == 0 && monthValue == 2 && dayValue < 30) return true;

else if ((monthValue == 4 || monthValue == 6 || monthValue == 9 || monthValue == 11) &&

dayValue > 30) return false;

return true;

}

static void CorrectInput(out int value)

{

string? str = null;

while (true)

{

str = Console.ReadLine();

if (int.TryParse(str, out value))

{

value = int.Parse(str);

break;

}

Console.WriteLine("re-Input value: ");

}

}

static void Main(string[] args)

{

bool replay = true;

while (replay)

{

string? str = null;

int day, month, year;

Console.WriteLine("Choise function:\n1 - DateService.GetDay\n" +

"2 - DateService.GetDaysSpan");

while(true)

{

str = Console.ReadLine();

if (str == "1" || str == "2") break;

else Environment.Exit(0);

}

switch(int.Parse(str)){

case 1:

{

Console.WriteLine("Day of week: " + DateService.GetDay(str));

break;

}

case 2:

{

while (true)

{

Console.WriteLine("Input day of year: ");

CorrectInput(out day);

Console.WriteLine("Input month of year: ");

CorrectInput(out month);

Console.WriteLine("Input year: ");

CorrectInput(out year);

if (CheckDate(day, month, year)) break;

}

Console.WriteLine("Between dates: " +

DateService.GetDaysSpan(day, month, year));

break;

}

}

Console.WriteLine("Input '+' to end or another symbol to continue");

str = Console.ReadLine();

if (str == "+") replay = false;

}

}

}

}

Результат работы программы продемонстрирован ниже (см. рисунок 4).

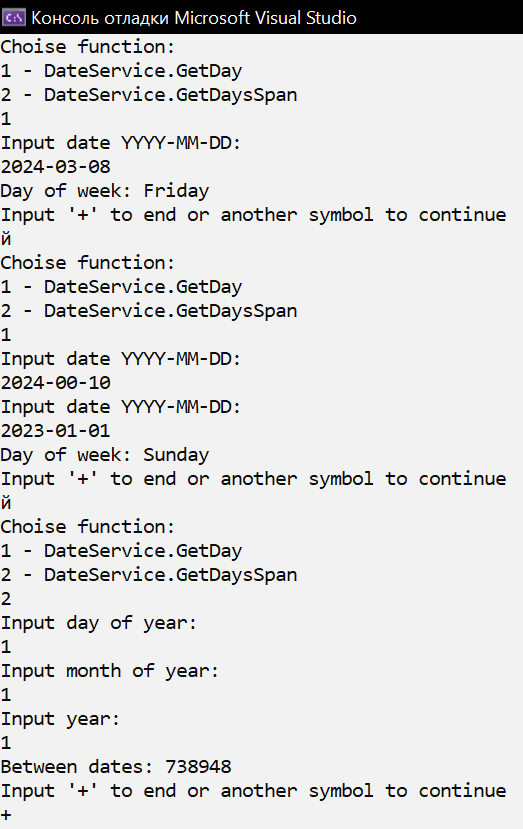


Рисунок 4 – Результат работы программы третьего задания

# Вывод

В ходе лабораторной работы были изучены принципы описания простейших классов, создание методов для решения задач. Был изучены методы класса DateTime и TimeSpan.