Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерного проектирования

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 05

|  |
| --- |
| Выполнила студент группы 353504  АНТОНОВА Лидия Сергеевна |
|  |
| (дата, подпись студента) |
| Проверил ассистент каф. Информатики  РОМАНЮК Максим Валерьевич |
|  |
| (дата, подпись преподавателя) |

Минск 2024

# 1 задание

1 Описать класс (с произвольным именем) в отдельном файле проекта. В созданном классе разработать метод f(n), который для заданного натурального числа n находит значение n^2. Вычислить с помощью него значение выражения 2^2+3^2+4^2.

2 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. Условие изображено на рисунке 1.

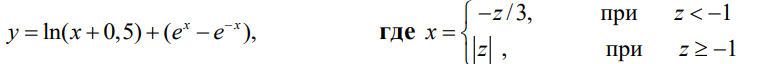


Рисунок 1

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).

# Выполнение работы

Описать класс (с произвольным именем) в отдельном файле проекта. В созданном классе разработать метод f(n), который для заданного натурального числа n находит значение n^2. Вычислить с помощью него значение выражения 2^2+3^2+4^2.

Ниже представлен листинг кода.

using System;

public class Square

{

public static int f(int n)

{

return n \* n;

}

}

class Program

{

static void Main()

{

Console.WriteLine($"2^2 + 3^2 + 4^2 = {Square.f(2) + Square.f(3) + Square.f(4)}");

Console.WriteLine("Enter n:");

int n;

while (!Int32.TryParse(Console.ReadLine(), out n))

{

Console.Write("Invalid input. Please enter a valid number for n: ");

}

Console.WriteLine($"{n}^2 = {Square.f(n)}");

}

}

В данном случае был разработан класс Square, который содержит статический метод f(n). Этот метод предназначен для вычисления квадрата заданного натурального числа n. Класс Square и его метод f(n) демонстрируют использование статических методов в языке C#, что позволяет вызывать метод без необходимости создания экземпляра класса.

Ключевым моментом в реализации класса Square является простота и четкость кода, что делает его легко читаемым и понятным. Метод f(n) использует арифметическую операцию умножения для возврата квадрата числа, что является базовой операцией в программировании и математике.

В основной программе Program.cs демонстрируется использование метода f(n) для вычисления суммы квадратов чисел 2, 3 и 4. Результат вычисления выводится в консоль, что позволяет пользователю наглядно увидеть результат работы программы. Кроме того, программа предлагает пользователю ввести собственное значение n и вычислить его квадрат, что добавляет интерактивности и практической пользы.

Программа также содержит обработку ввода пользователя с помощью цикла и метода Int32.TryParse, который гарантирует, что будет введено корректное числовое значение. Это важный аспект создания устойчивых к ошибкам пользовательских интерфейсов.

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. Условие изображено на рисунке 1.

Ниже представлен листинг кода.

// Function.cs

using System;

namespace Services

{

public class Function

{

public void CalculateAndPrint(double z)

{

double x;

double y;

int branchNumber;

if (z < -1)

{

x = -z / 3;

branchNumber = 1;

}

else

{

x = Math.Abs(z);

branchNumber = 2;

}

y = Math.Log(x + 0.5) + (Math.Exp(x) - Math.Exp(-x));

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

Console.WriteLine($"Branch number: {branchNumber}");

}

}

}

// Program.cs  
using System;

using Services;

class Program

{

static void Main()

{

var calculate = new Function();

Console.WriteLine("Enter z:");

double z;

while (!Double.TryParse(Console.ReadLine(), out z))

{

Console.Write("Invalid input. Please enter a valid number for z: ");

}

calculate.CalculateAndPrint(z);

}

}

В данном задании требовалось добавить в проект папку Services и в ней разработать класс с методом для вычисления значения функции по заданному условию.

Класс, названный Function, был создан для выполнения специфической задачи: вычисления значения функции в зависимости от входного параметра z. Метод CalculateAndPrint в этом классе не только выполняет вычисления, но и информирует пользователя о результате и логическом пути (ветке), по которому были проведены расчеты.

Важной особенностью реализации является использование условных операторов для определения логики вычисления. В зависимости от значения z, выбирается соответствующая ветка: если z меньше -1, используется первая ветка с определенной формулой для x, в противном случае — вторая ветка. Это позволяет динамически изменять ход вычислений в зависимости от входных данных.

Функция Math.Log используется для вычисления натурального логарифма, а функции Math.Exp и Math.Exp(-x) — для вычисления экспонент, что демонстрирует применение математических функций из стандартной библиотеки .NET. Результаты вычислений выводятся в консоль, что делает программу интерактивной и позволяет пользователю наблюдать за процессом работы программы.

В основной программе Program.cs представлен интерфейс для взаимодействия с пользователем. Пользователю предлагается ввести значение z, после чего программа использует метод CalculateAndPrint для вычисления и вывода результата. Обработка ввода с помощью Double.TryParse обеспечивает корректность введенных данных и устойчивость программы к ошибкам пользователя.

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).

Ниже представлен листинг кода.

// DateServices.cs

using System;

using System.Globalization;

namespace Services

{

public static class DateServices

{

public static string GetDay(string date)

{

DateTime parsedDate;

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

{

return parsedDate.ToString("dddd", new CultureInfo("ru-RU"));

}

else

{

return "Некорректная дата";

}

}

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

{

DateTime currentDate = DateTime.Now;

DateTime futureDate;

try

{

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

}

catch (ArgumentOutOfRangeException)

{

return -1;

}

if (futureDate < currentDate)

{

return -1;

}

else

{

return (futureDate - currentDate).Days;

}

}

}

}

// Program.cs

using System;

using Services;

class Program

{

static void Main()

{

Console.WriteLine("1 - Day of Week \"dd.MM.yyyy\"\n2 - Number of Days");

var x = Convert.ToInt16(Console.ReadLine());

switch (x)

{

case 1:

var s = Console.ReadLine();

Console.WriteLine(DateServices.GetDay(s));

break;

case 2:

Console.WriteLine("Input day: ");

var day = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Input month: ");

var month = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Input year: ");

var year = Convert.ToInt32(Console.ReadLine());

Console.WriteLine(DateServices.GetDaysSpan(day, month, year));

break;

}

}

}

Был создан класс DateServices, который предоставляет два метода: GetDay и GetDaysSpan. Эти методы позволяют работать с датами, выполняя расчеты, связанные с днями недели и разницей в днях между датами.

Класс DateServices является статическим, что позволяет вызывать его методы без необходимости создания экземпляра класса. Это делает класс удобным для выполнения операций с датами, которые могут быть необходимы в различных частях приложения.

Метод GetDay принимает строковое представление даты и пытается преобразовать его в объект DateTime. Если преобразование успешно, метод возвращает день недели для этой даты на русском языке, используя культурные настройки ru-RU. В случае некорректного ввода метод возвращает сообщение об ошибке.

Метод GetDaysSpan рассчитывает количество дней между текущей датой и заданной датой в будущем. Для этого он принимает числовые параметры дня, месяца и года, создает из них объект DateTime и вычисляет разницу в днях. Если заданная дата меньше текущей или введены некорректные данные, метод возвращает -1, что может служить индикатором ошибки для вызывающего кода.

В основной программе Program.cs представлен пользовательский интерфейс, который позволяет выбрать одну из двух операций: определение дня недели или расчет количества дней до заданной даты. Пользовательский ввод обрабатывается с помощью конструкции switch, которая направляет выполнение программы в зависимости от выбора пользователя.

# Вывод

В ходе выполнения Лабораторной работы №3 были приобретены навыки работы с простыми классами.