**Кафедра інженерії програмного забезпечення та кібербезпеки**

ЛАБОРАТОРНА РОБОТА №4

**МЕТОДИ**

|  |  |
| --- | --- |
| **Виконав:** | *Лупина Ігор Ігорович* |
| **Група:** | *1-6* |
| **Варіант:** | *12* |
|  |  |
| **Викладач:** | *Макоєдова Валентина Олександрівна* |
| **Бали:** |  |

|  |  |
| --- | --- |
| **освітній ступінь** | **бакалавр /** bachelor |
| **галузь знань** | **12 Інформаційні технології /** Information Technology |
| **спеціальність** | **121Інженерія програмного забезпечення /** Software engineering |
|  |  |

**Київ 2023**

**Тема:** Методи

**Мета:** Закріпити вміння використовувати методи при написанні програм.

**Завдання:** Виконати задачі згідно варіанту

**Хід роботи**

**Задача 1:** Дано натуральне сило k. Визначити суму цифр цього числа.

Для виконання задачі 1 було написано наступну програму:

namespace Task1;

internal class Program

{

static void Main()

{

Console.Write("Input value: ");

if (!long.TryParse(Console.ReadLine(), out var value))

{

Console.WriteLine("Input is not an integer");

return;

}

Console.WriteLine("Sum of number's digits: {0}", DigitSum(value));

}

public static int DigitSum(long value)

{

var res = 0;

while (value > 0)

{

res += (int)(value % 10);

value /= 10;

}

return res;

}

}

В програмі присутні наступні методи:

* DigitSum – повертає суму цифр числа

Результат виконання програми зображено на рисунку 1.

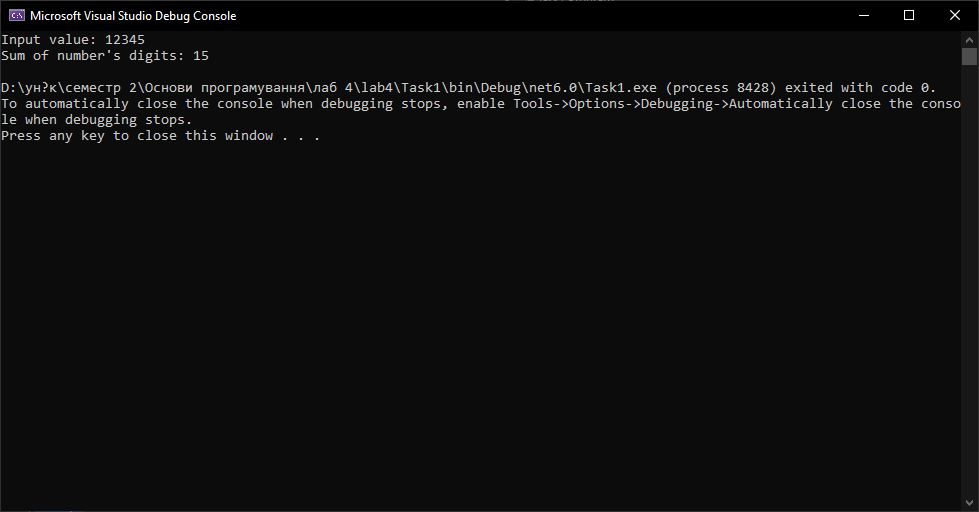


Рисунок 1 – Результат виконання програми задачі 1

**Задача 2:** Створити додаток з допомогою якого можна для ряду, члени якого обчислюються по формулі підрахувати суму членів ряду з точністю до 0,000001 і суму перших 10 членів ряду.

Функція ряду:

Для виконання задачі 2 було написано наступну програму:

using static System.Math;

namespace Task2;

internal class Program

{

const double Precision = 1e-6;

static void Main(string[] args)

{

Console.WriteLine("Sum of first 10 elements: {0:0.000000}", CalculateFirst10());

var value = Calculate();

Console.WriteLine("Sum with 1e-6 precision: {0:0.000000} ({1} elements calculated)", value.Item1, value.Item2);

}

static double Function(int n)

=> Pow(-1, n) / (Pow(n, 2) + Pow(2, n));

static double CalculateFirst10()

=> Enumerable.Range(1, 10).Aggregate(.0, (x, y) => x += Function(y));

static (double, int) Calculate()

{

var i = 1;

var res = .0;

double value;

do

{

value = Function(i);

res += value;

i++;

} while (Abs(value) > Precision);

return (res, i);

}

}

В програмі присутні наступні методи:

* Function – повертає значення n-ого елементу ряду
* Calculate – обчислює суму ряду з точністю 0,000001 та кількість обчислених елементів
* CalculateFirst10 – обчислює суму перших десяти елементів ряду

Результат виконання програми зображено на рисунку 2.

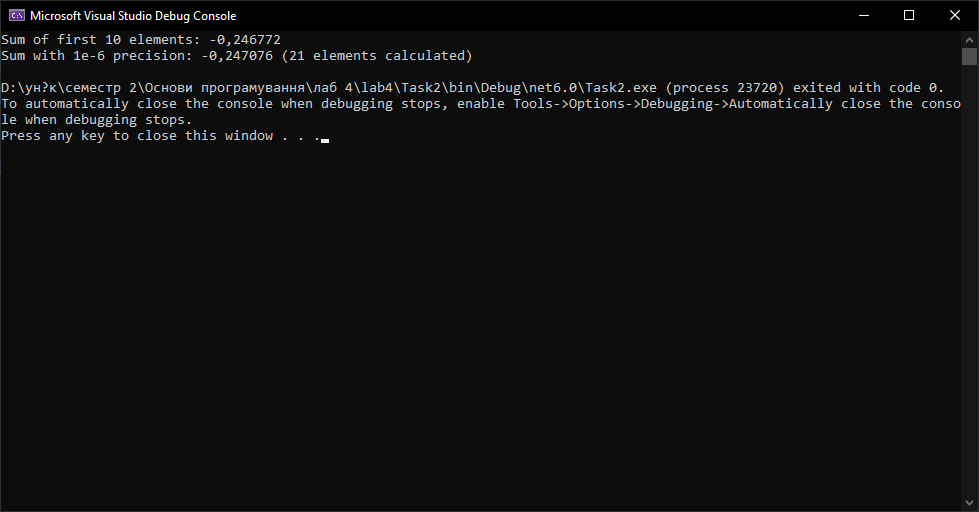


Рисунок 2 – Результат виконання програми задачі 2

**Задача 3:** Протабулювати функцію на проміжку [0; i] з кроком h = 0.1i, де i – номер варіанту. Результати обчислень вивести у вигляді таблиці пар чисел x, y та графіка. Виконати завдання пошуку даних відповідно до вашого варіанта. Якщо шуканих даних немає, то вивести про це повідомлення.

Функція:

Завдання пошуку: Обчислити добуток та кількість усіх значень функції для яких виконується нерівність y < -3, y > 0.4

Для вирішення задачі 3 було написано наступну програму:

using System.Text;

using System.Drawing;

using System.Drawing.Imaging;

using static System.Math;

namespace Task3;

internal class Program

{

const int Variant = 12;

const double XMin = 0;

const double XMax = Variant;

const double Step = 0.1 \* Variant;

const char Horizontal = (char)0x2500;

const char Vertical = (char)0x2502;

const char Cross = (char)0x253C;

const int Height = 1080;

const int Width = 1920;

const int Margin = 100;

const int Radius = 2;

static void Main()

{

var values = new List<(double, double)>();

for (double i = XMin; i <= XMax; i += Step)

{

values.Add((i, Function(i)));

}

PrintTable(values);

AnalyzeData(values);

GenerateGraph(values, Color.Black, Color.Red);

}

static double Function(double x)

=> 4 \* Sin(Abs(x / 2)) + 9.1 \* Cos(x - 1);

static void PrintTable(List<(double, double)> values)

{

Console.WriteLine($"{"",3}{'x',-4}|{'y',5}");

var delimiter = new StringBuilder().Append(Horizontal, 7).Append(Cross)

.Append(Horizontal, 9).ToString();

Console.WriteLine(delimiter);

foreach (var item in values)

{

Console.WriteLine("{0,6:0.00} {1} {2,8:0.0000}", item.Item1, Vertical, item.Item2);

}

Console.WriteLine();

}

static void AnalyzeData(List<(double, double)> values)

{

var query = values.Select(x => x.Item2).Where(x => x < -3 || x > 0.4);

var count = query.Count();

var product = query.Aggregate(1.0, (x, y) => x \*= y);

Console.WriteLine("y < -3 or y > 0.4");

Console.WriteLine($"Count: {count}");

Console.WriteLine($"Product: {product:F4}");

Console.WriteLine();

}

static void GenerateGraph(List<(double, double)> values, Color gridColor, Color graphColor)

{

using var bitmap = new Bitmap(Width, Height);

using var graphics = Graphics.FromImage(bitmap);

using var pen = new Pen(Color.White, 1);

using var font = new Font("Consolas", 12);

graphics.FillRectangle(pen.Brush, new(0, 0, Width, Height));

var xCount = values.Count;

var xBaseLine = 0;

var yMax = values.Max(x => x.Item2);

yMax = Ceiling(Abs(yMax) + 1) \* Sign(yMax);

var yMin = values.Min(x => x.Item2);

yMin = Ceiling(Abs(yMin)) \* Sign(yMin);

var yValStep = 1;

var yCount = Abs(yMin - yMax);

#region y Axis

pen.Color = gridColor;

var yStep = (Height - 2 \* Margin) / yCount;

var yBaseLine = Margin;

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

{

var y = Height - (Margin + (int)Truncate(yStep \* i));

if (i != yCount)

{

var yVal = yMin + yValStep \* i;

if (yVal == 0) xBaseLine = y;

var str = yVal.ToString("F2");

var strSize = graphics.MeasureString(str, font);

graphics.DrawLine(pen, new(yBaseLine - 5, y), new(yBaseLine + 5, y));

graphics.DrawString(str, font, pen.Brush, new PointF(yBaseLine - strSize.Width - 10, y - strSize.Height / 2));

continue;

}

graphics.DrawLine(pen, new(yBaseLine, y), new(yBaseLine - 5, y + 5));

graphics.DrawLine(pen, new(yBaseLine, y), new(yBaseLine + 5, y + 5));

}

graphics.DrawLine(pen, new(yBaseLine, Margin), new(yBaseLine, Margin + (int)Truncate(yStep \* yCount)));

#endregion

#region x Axis

var xStep = (double)(Width - 2 \* Margin) / xCount;

PointF? lastPoint = null;

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

{

var x = Margin + (int)Truncate(xStep \* i);

#region Arrow

if (i == xCount)

{

graphics.DrawLine(pen, new(x, xBaseLine), new(x - 5, xBaseLine - 5));

graphics.DrawLine(pen, new(x, xBaseLine), new(x - 5, xBaseLine + 5));

continue;

}

#endregion

#region Point

pen.Color = graphColor;

var yVal = values[i].Item2;

var y = xBaseLine - (int)(yStep \* yVal);

var point = new PointF(x, y);

var halfSize = new Size(Radius, Radius);

graphics.FillRectangle(pen.Brush, new RectangleF(point - halfSize, 2 \* halfSize));

if (lastPoint != null)

{

graphics.DrawLine(pen, lastPoint.Value, point);

}

lastPoint = point;

#endregion

#region Grid

pen.Color = gridColor;

var str = values[i].Item1.ToString("F2");

var strSize = graphics.MeasureString(str, font);

graphics.DrawLine(pen, new(x, xBaseLine - 5), new(x, xBaseLine + 5));

if (i != 0 && strSize.Width < xStep) graphics.DrawString(str, font, pen.Brush, new PointF(x - strSize.Width / 2.0f, xBaseLine + strSize.Height / 2));

#endregion

}

graphics.DrawLine(pen, new(Margin, xBaseLine), new(Margin + (int)Truncate(xStep \* xCount), xBaseLine));

#endregion

bitmap.Save("graph.png", ImageFormat.Png);

}

}

В програми присутні наступні методи:

* Function – повертає значення функції для вказаного значення x
* PrintTable – виводить список пар значень x, y в консоль у вигляді таблиці
* AnalyzeData – аналізує список пару значень x, y згідно завдання та виводить результати в консоль
* GenerateGraph – генерує графік функції за списком пар значень x, y та зберігає його у вигляді зображення в форматі PNG

Результати роботи програми зображено на рисунках 3-4.



Рисунок 3. Результати виконання програми задачі 3 в консолі

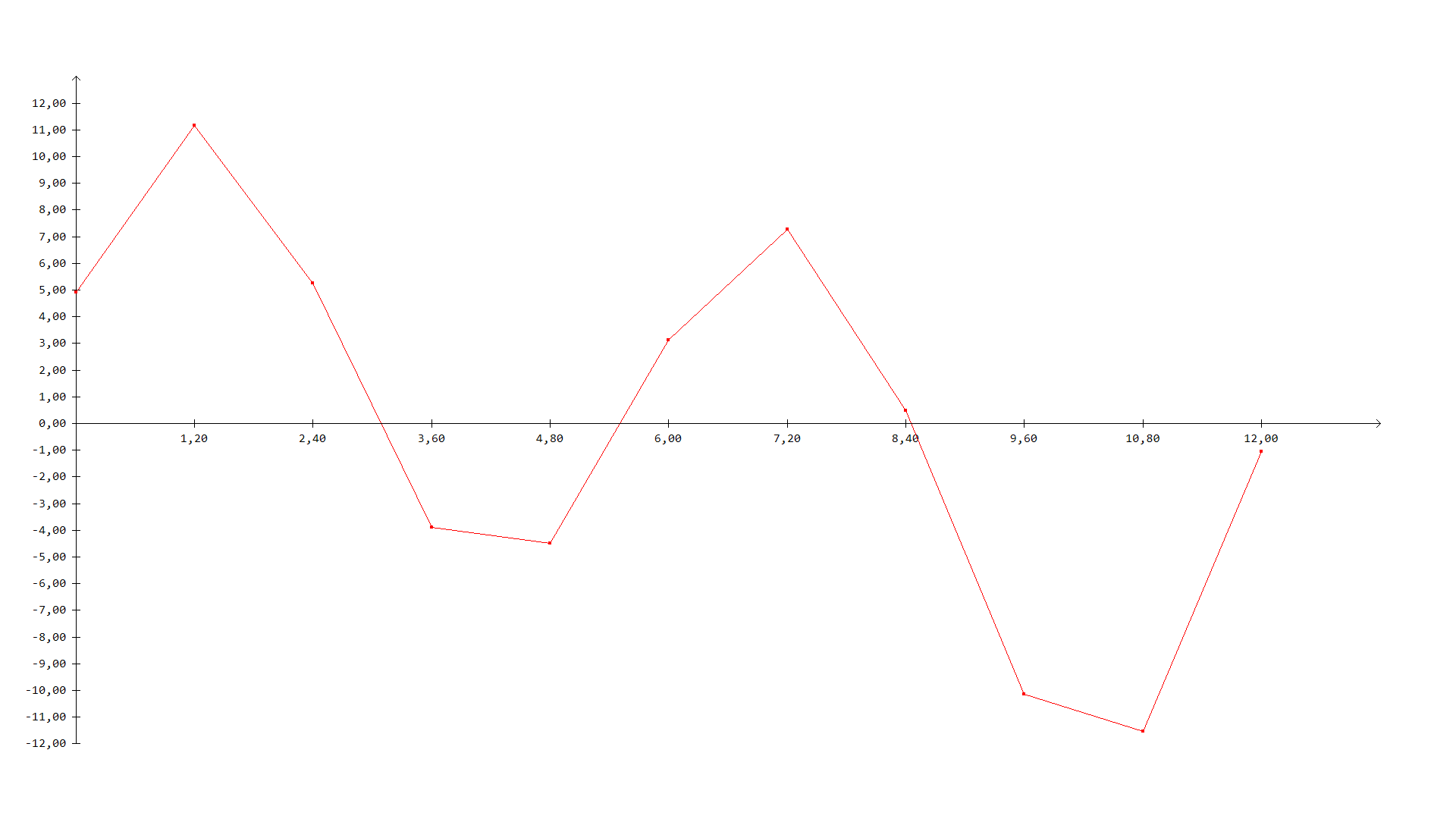


Рисунок 4. Згенерований графік функції

**Задача 4:** Визначити число способів виплати Отримати всі способи виплати суми грн. за допомогою монет номіналом 1, 2, 5 і 10 грн.

Для виконання задачі 4 було написано наступну програму:

using System.Text;

namespace Task4;

internal class Program

{

static void Main()

{

Console.Write("Target: ");

if (!int.TryParse(Console.ReadLine(), out var value))

{

Console.WriteLine("Target is not an integer");

return;

}

Analyze(value);

}

static void Analyze(int value)

{

for (int tens = 0; tens <= value / 10; tens++)

for (int fives = 0; fives <= (value - 10 \* tens) / 5; fives++)

for (int twos = 0; twos <= (value - 10 \* tens - 5 \* fives) / 2; twos++)

{

int ones = value - 10 \* tens - 5 \* fives - 2 \* twos;

var values = new[] { (1, ones), (2, twos), (5, fives), (10, tens) }

.Where(x => x.Item2 != 0)

.Select((x) => $"{x.Item1} \* {x.Item2}");

var str = new StringBuilder()

.Append(value)

.Append(" = ")

.AppendJoin(" + ", values);

Console.WriteLine(str);

}

}

}

В програмі присутні наступні методи:

* Analyze – аналізує число згідно завдання та виводить результати в консоль

Результат виконання програми зображено на рисунку 5.

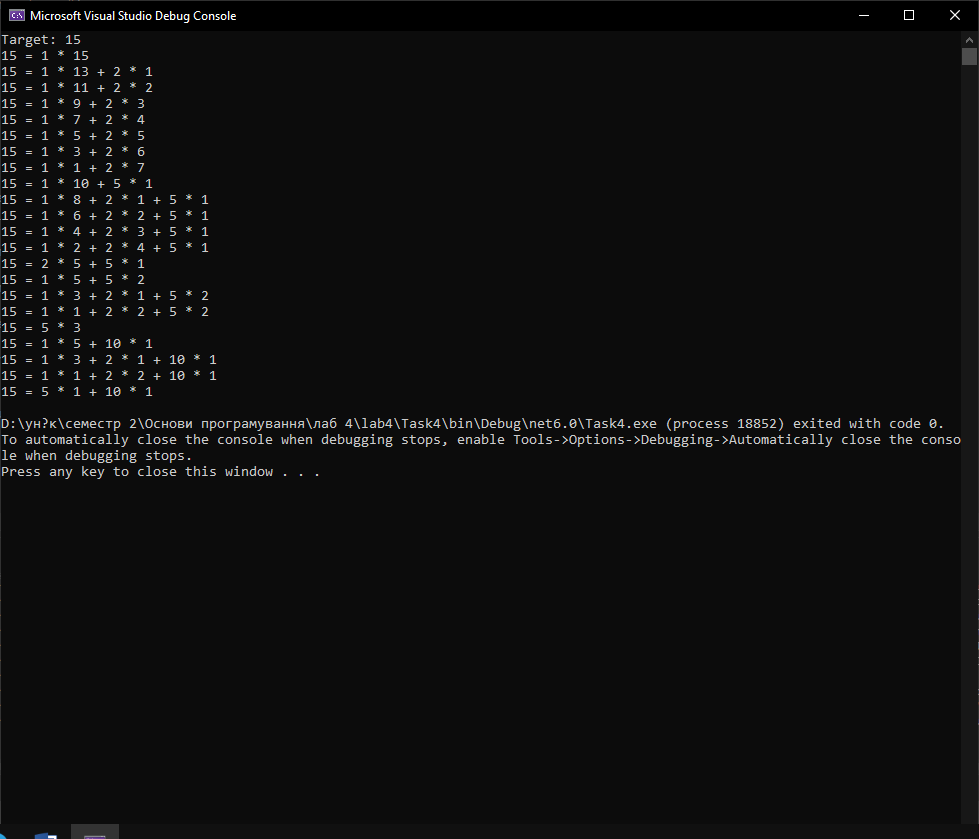


Рисунок 5. Результат виконання програми задачі 4

**Висновок:** Під час виконання практичної роботи я закріпив вміння використання методів при написанні програм.