**Кафедра інженерії програмного забезпечення та кібербезпеки**

ЛАБОРАТОРНА РОБОТА №2

**Основні оператори мови програмування C#**

|  |  |
| --- | --- |
| **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:** Основні оператори мови програмування C#

**Завдання:**

1. Створити блок-схему розв’язання задачі у відповідності з індивідуальним варіантом.
2. Розробити програму мовою C# у відповідності з індивідуальним варіантом.
3. Звіт оформити у середовищі текстового редактора MS Word: титульний лист, блок-схема, код програми, скріншот виконання програми).

**Задача 1:**

**Задача 2:**

Розробити програму, під час виконання якої можна ввести число Х. Якщо введено число, менше 1, слід вивести повідомлення про помилку. В іншому випадку потрібно вивести послідовність Х чисел Фібоначчі. Задачу слід вирішувати за допомогою циклу.

Для виконання задачі 1 було розроблено блок-схему, зображену на рисунку 1. Блок-схеми функцій func, sum та product зображено на рисунках 2, 3 і 4 відповідно.

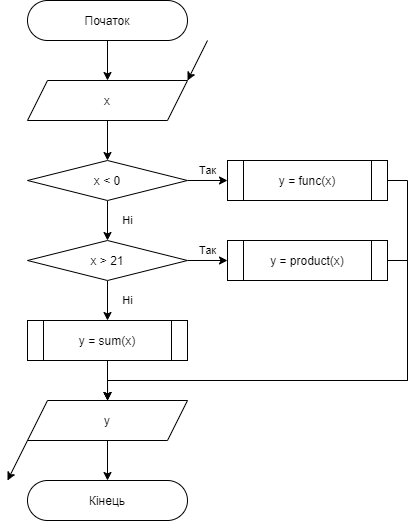


Рисунок 1. Блок-схема алгоритму розв’язання задачі 1

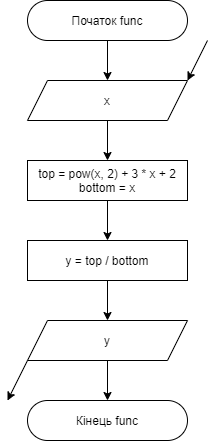


Рисунок 2. Блок-схема алгоритму функції func

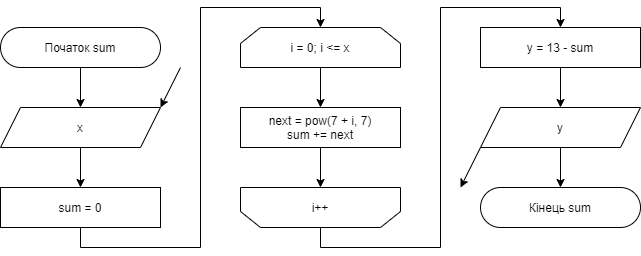


Рисунок 3. Блок-схема алгоритму функції sum

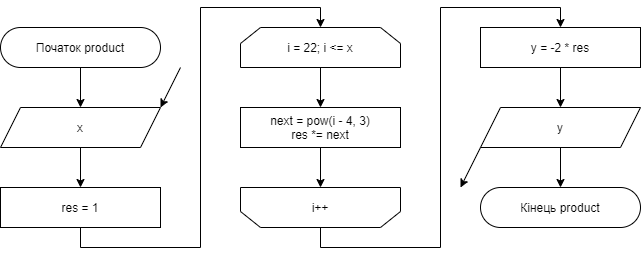


Рисунок 4. Блок-схема алгоритму функції product

Згідно з блок-схемами алгоритмів було розроблено програму мовою C#:

namespace Lab2.Task1;

class Program

{

static void Main()

{

Console.Write("x: ");

var xStr = Console.ReadLine();

if (!int.TryParse(xStr, out int x))

{

Console.WriteLine($"{xStr} is not an integer");

return;

}

var y = x switch

{

< 0 => Func(x),

> 21 => Product(x),

\_ => Sum(x)

};

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

}

private static double Func(int x)

{

var top = Math.Pow(x, 2) + 3 \* x + 2;

var bottom = x;

var y = top / bottom;

return y;

}

private static double Sum(int x)

{

var sum = .0;

for (var i = 0; i <= x; i++)

{

var next = Math.Pow(7 + i, 7);

sum += next;

}

var y = 13 - sum;

return y;

}

private static double Product(int x)

{

var res = 1.0;

for (var i = 22; i <= x; i++)

{

var next = Math.Pow(i - 4, 3);

res \*= next;

}

var y = -2 \* res;

return y;

}

}

Результат виконання програми з різними параметрами зображено на рисунках 5-7.

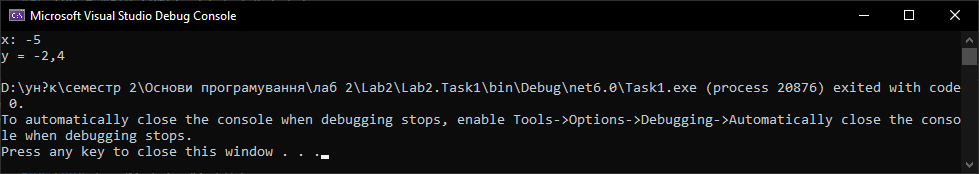


Рисунок 5. Результат виконання програми з параметром x = -5

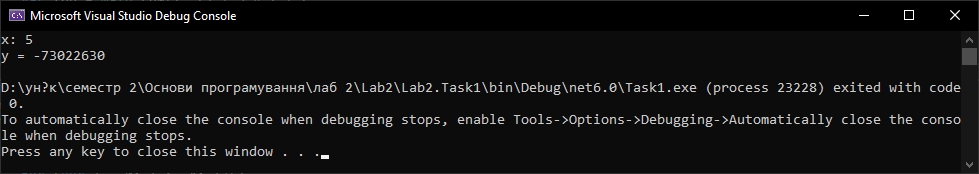


Рисунок 6. Результат виконання програми з параметром x = 5

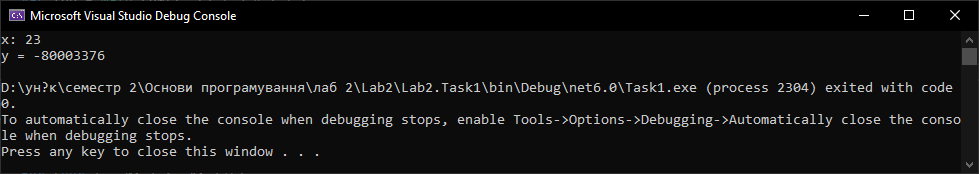


Рисунок 7. Результат виконання програми з параметром x = 23

Для виконання задачі 2 було розроблено блок-схему, зображену на рисунку 8.

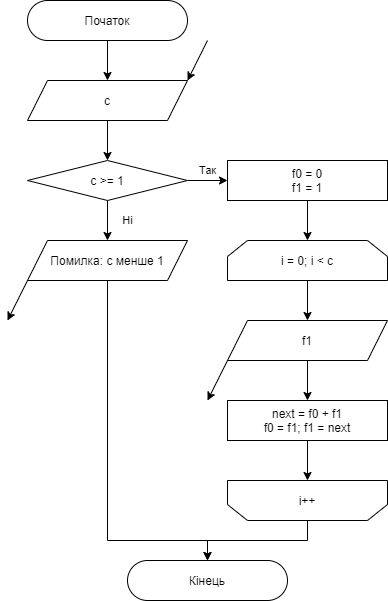


Рисунок 8. Блок-схема алгоритму вирішення задачі 2

Згідно з блок-схемою алгоритму було розроблено програму мовою C#:

namespace Lab2.Task2;

class Program

{

static void Main()

{

Console.Write("Count: ");

var cStr = Console.ReadLine();

if (!int.TryParse(cStr, out int c))

{

Console.WriteLine($"{cStr} is not an integer");

return;

}

if (c < 1)

{

Console.WriteLine("Error: x must be 1 or greater");

return;

}

var f0 = 0;

var f1 = 1;

for (var i = 0; i < c; i++)

{

Console.Write($"{f1} ");

var next = f0 + f1;

f0 = f1;

f1 = next;

}

Console.WriteLine();

}

}

Результат виконання програми з різними параметрами зображено на рисунках 9-12.

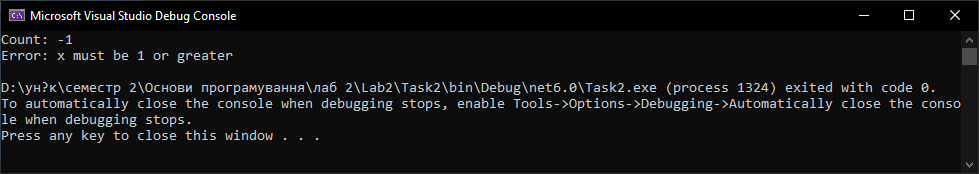


Рисунок 9. Результат виконання програми з параметром с = -1

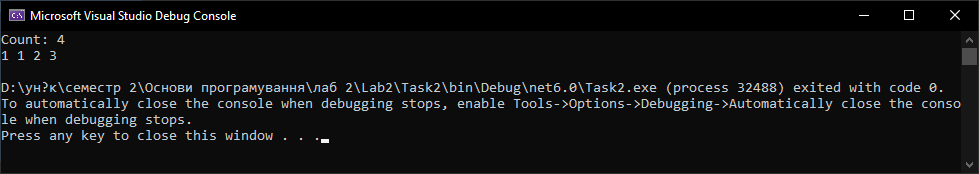


Рисунок 10. Результат виконання програми з параметром с = 4

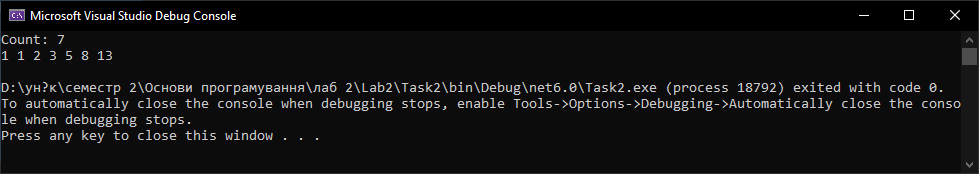


Рисунок 11. Результат виконання програми з параметром с = 7

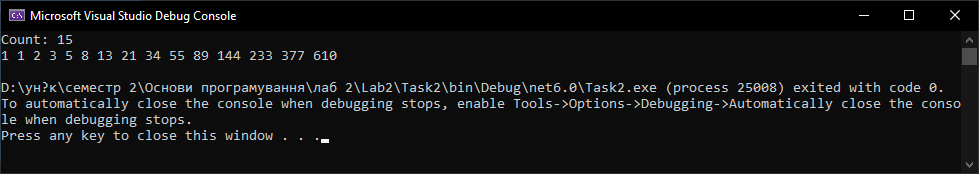


Рисунок 12. Результат виконання програми з параметром с = 15

**Висновок:** під час виконання лабораторної роботи я ознайомився з основними операторами мови C#, опанував їх використання та застосував їх у розв’язанні прикладних задач.