

Лабораторна робота № 1
МОВА ПРОГРАМУВАННЯ C#. ТИПИ ДАНИХ. РЕАЛІЗАЦІЯ
ОСНОВНИХ ТИПІВ АЛГОРИТМІВ

Мета роботи: Набути умінь і навичок створення простих консольних додатків мовою C# у середовищі Microsoft Visual Studio 2022.

Варіант 15

Завдання для лабораторної роботи

Виконати завдання 1, 2 до даної лабораторної роботи. Номер варіанту обрати згідно номеру у журналі групи.

Завдання 1

Варіант 15. Дано катети прямокутного трикутника. Знайти його гіпотенузу та площу.

```
static void Task1()
{
    Console.WriteLine("Задача 1: Знайти гіпотенузу та площу прямокутного
трикутника");

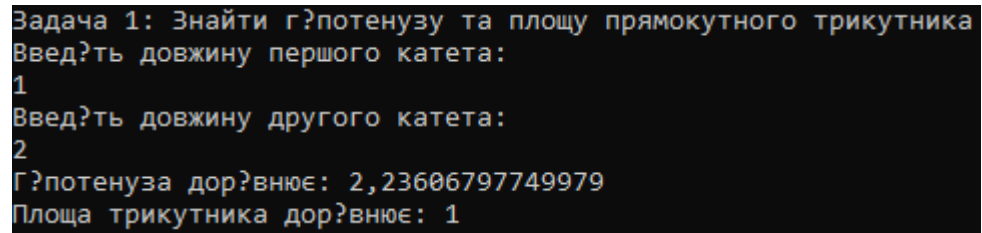
    Console.WriteLine("Введіть довжину першого катета:");
    double a = Convert.ToDouble(Console.ReadLine());

    Console.WriteLine("Введіть довжину другого катета:");
    double b = Convert.ToDouble(Console.ReadLine());

    double c = Math.Sqrt(a * a + b * b);
    Console.WriteLine($"Гіпотенуза дорівнює: {c}");

    double area = (a * b) / 2;
```

```
Console.WriteLine($"Площа трикутника дорівнює: {area}");  
}
```



Задача 1: Знайти гіпотенузу та площу прямокутного трикутника
Введіть довжину першого катета:
1
Введіть довжину другого катета:
2
Гіпотенуза дорівнює: 2,23606797749979
Площа трикутника дорівнює: 1

Рис. 1.1 – Результат виконання пошуку гіпотенузи та площі прямого трикутника

Завдання 2

Варіант 15. Дано дійсні числа a , b , c . Знайти суму тих з них, які належать інтервалу $[x, y]$.

```
static void Task2()  
{
```

```
    Console.WriteLine("\nЗадача 2: Знайти суму чисел, які належать інтервалу  
    [x, y]");
```

```
    Console.WriteLine("Введіть число a:");  
    double numA = Convert.ToDouble(Console.ReadLine());
```

```
    Console.WriteLine("Введіть число b:");  
    double numB = Convert.ToDouble(Console.ReadLine());
```

```
    Console.WriteLine("Введіть число c:");  
    double numC = Convert.ToDouble(Console.ReadLine());
```

```
    Console.WriteLine("Введіть початок інтервалу x:");
```

```
double x = Convert.ToDouble(Console.ReadLine());
```

```
Console.WriteLine("Введіть кінець інтервалу y:");
```

```
double y = Convert.ToDouble(Console.ReadLine());
```

```
double sum = 0;
```

```
if (x <= numA && numA <= y)
```

```
{  
    sum += numA;  
}
```

```
if (x <= numB && numB <= y)
```

```
{  
    sum += numB;  
}
```

```
if (x <= numC && numC <= y)
```

```
{  
    sum += numC;  
}
```

```
    Console.WriteLine($"Сума чисел, які належать інтервалу [{x}, {y}],  
дорівнює: {sum}");  
}
```

```
Задача 2: Знайти суму чисел, які належать інтервалу [x, y]
Введіть число a:
1
Введіть число b:
5
Введіть число c:
7
Введіть початок інтервалу x:
1
Введіть кінець інтервалу y:
6
Сума чисел, які належать інтервалу [1, 6], дорівнює: 6
```

Рис. 1.2 – Результат пошуку суми чисел, які належать інтервалу [x, y]

Питання для самоконтролю

1. Які основні типи даних у мові C#?

Основні типи даних у мові C#: цілі числа (int, long), числа з рухомою комою (float, double), символи (char), логічні значення (bool), рядки (string).

2. Які операції використовують у мові C#

Операції, що використовуються у мові C#: арифметичні (+, -, *, /), порівняння (==, !=, <, >), логічні (&&, ||, !), операції присвоєння (=), операції збільшення/зменшення (++, --), бітові операції (&, |, ^, <<, >>).

3. Який загальний вигляд оператора розгалуження if?

Загальний вигляд оператора розгалуження if:

if (умова)

{

}

else

{

}

4. Який загальний вигляд оператора вибору switch

Загальний вигляд оператора вибору switch:

switch (вираз)

```
{  
case значення1:  
break;  
case значення2:  
break;  
default:  
break;
```

5. Який загальний вигляд оператора циклу while?

Загальний вигляд оператора циклу while:

while (умова)

```
{  
}
```

6. Який загальний вигляд оператора циклу do-while?

Загальний вигляд оператора циклу do-while:

do

```
{
```

```
} while (умова);
```

7. Які відмінності у роботі операторів циклу while та do-while?

Відмінності у роботі операторів циклу while та do-while: оператор while перевіряє умову перед входом у цикл, тоді як оператор do-while перевіряє умову після кожного виконання циклу, тобто код у do-while завжди виконається принаймні один раз.

8. Який загальний вигляд оператора for?

Загальний вигляд оператора for:

for (ініціалізація; умова; інкремент)

```
{  
}
```

9. За яким алгоритмом виконується оператор for?

Оператор `for` виконується за алгоритмом: спочатку виконується ініціалізація, потім перевіряється умова, якщо умова істинна, виконується тіло циклу, після чого виконується інкремент або декремент, і цей процес повторюється до тих пір, поки умова стає хибною.

10. Яке призначення оператора `break`?

Призначення оператора `break` - зупинка виконання циклу або виходу із комутатора (`switch`).

11. Яке призначення оператора `continue`?

Призначення оператора `continue` - перехід до наступної ітерації циклу, обходячи решту коду у цьому циклі.