

Лабораторна робота 2-3

Базові алгоритмічні конструкції (алгоритми обчислювальних процесів, що розгалужуються)

(тема 14, 4 години)

Мета роботи:

Навчитись складати базові алгоритми, які забезпечують розгалуження обчислювальних процесів.

Завдання:

скласти словесний опис та відповідну графічну схему алгоритму обчислювального процесу що розгалужується згідно індивідуального завдання;

надати псевдокод виконання поточного алгоритму.

Компетентності та результати навчання за лабораторною роботою

Компетентності	Результати навчання
1	2
Здійснювати аналіз обчислювальних процесів і виділяти процедури їх альтернативного продовження в залежності від поточних умов вибору.	Здатність складати базові алгоритми у вигляді відповідних графічних схем, які забезпечують розгалуження обчислювальних процесів
	Здатність складати псевдокод, що співвідноситься до алгоритму з розгалуженням.

Загальні відомості

У лінійних алгоритмах всі дії (операції, команди, оператори тощо) виконувалися послідовно і, як наслідок, вони не здатні реагувати на поточні умови.

Однак часто в процесі реалізації поточного сценарію потрібно змінювати потік керування, реагуючи на якісь зовнішні події.

Потік керування становить порядок, у якому виконуються оператори алгоритму. Крім того, часто використовуються терміни “порядок виконання” і “керуючий потік”.

Гілкою називають сегмент алгоритму, що містить один оператор або їх групу. Оператор розгалуження дозволяє запускати потрібний блок

операторів. Вибір здійснюється за умовою. Оператори розгалуження часто називають операторами вибору.

Основні символи схем алгоритмів наведено на рис. 1.



Рис. 1. Основні символи схем алгоритмів

Практично усі мови програмування забезпечують три типи структур вибору альтернатив (рис. 2):

єдиний вибір – структура if (ЯКЩО);

подвійний вибір – структура if / else (ЯКЩО / ІНАКШЕ);

множинний вибір – структура switch.

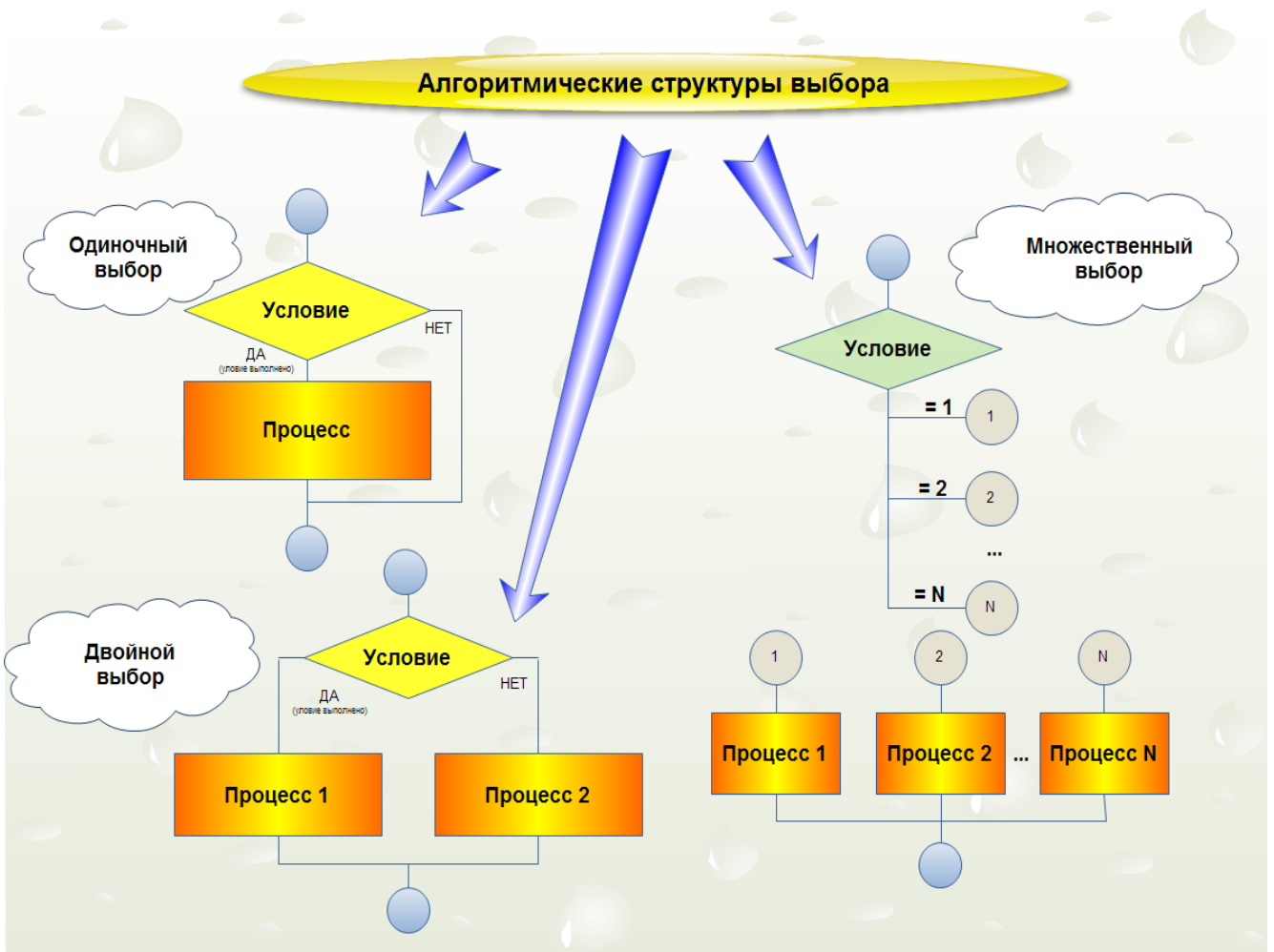


Рис. 2. Три типа структур выбору альтернатив

Структура выбору if.

Графічна схема оператора наведена на рис. 3.

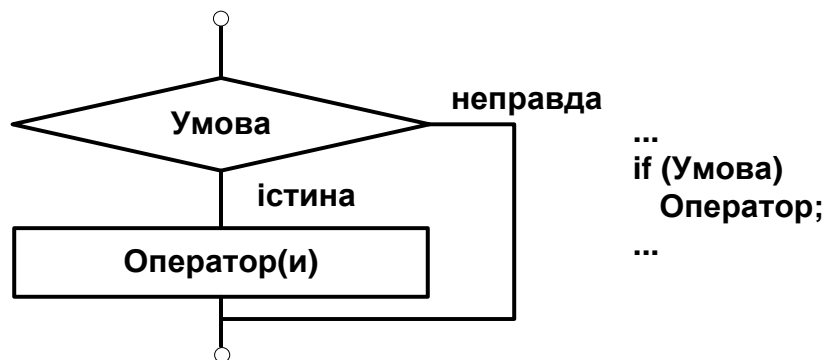


Рис. 3. Графічна схема оператора if

Структура выбору if / else.

Графічна схема оператора наведена на рис. 4.

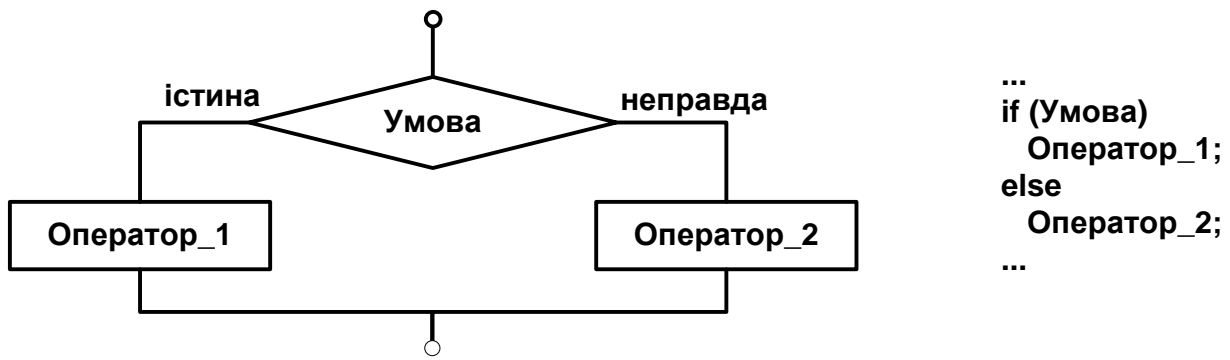


Рис. 4. Графічна схема оператора if / else

Множинний вибір – структура switch або case

Оператор switch дозволяє програмі обрати одну з кількох дій на основі значення заданого виразу. Логіка, яка реалізована switch, подібна до логіки оператора if / else. Графічна схема оператора switch наведена на рис. 5.

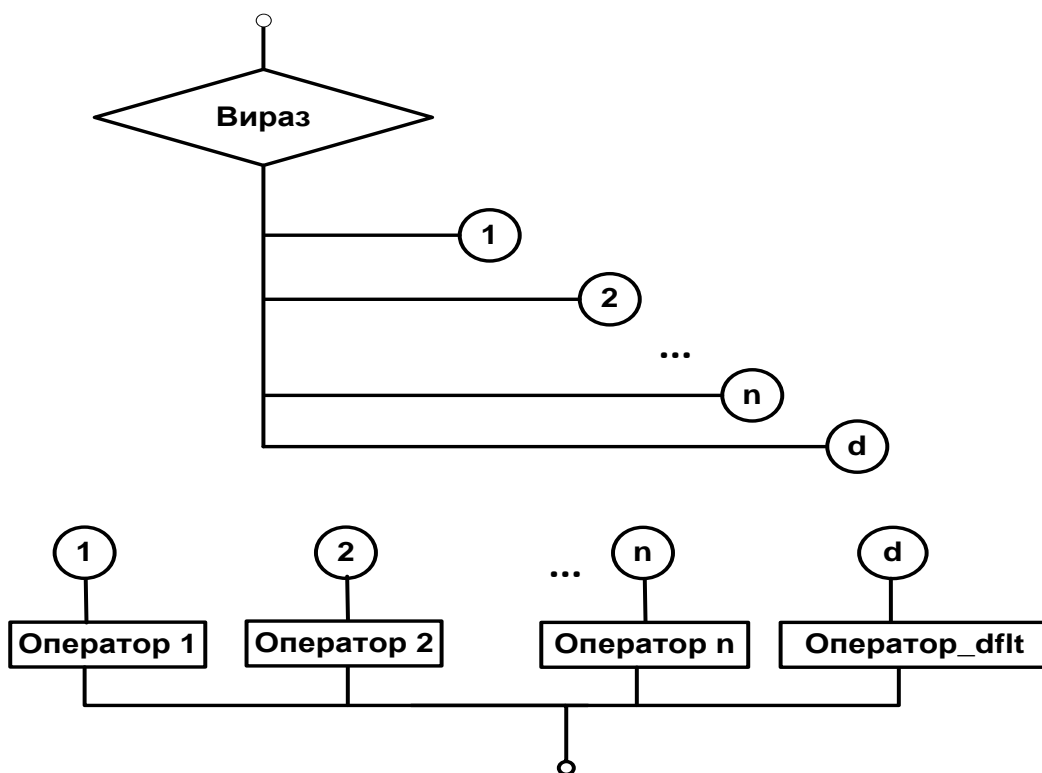


Рис. 5. Графічна схема оператора switch

Порядок виконання лабораторної роботи

1. Розробити алгоритм обчислення заданої функції та намалювати його графічну схему (блок-схему).

Формули для обчислення у вигляді $F = f(a, b, c, x)$ і опис змінних взяти з відповідного варіанта індивідуального завдання (див. файл «Додаток до лаб_2»).

Завдання відповідає наступному сценарію роботи:

Введіть чотири дійсні числа:

a =? <число_1> <Введення>

b =? < число _1> <Введення>

c =? < число _1> <Введення>

x =? < число _1> <Введення>

Обрана гілка № < тут виводиться номер гілки 1,2 або 3>

F = <тут виводиться результат обчислення>

2. В алгоритмі необхідно передбачити особливі випадки (наприклад, ділення на нуль), при цьому повинно виводитися відповідне попередження.

3. Підготувати арифметичні контрольні приклади (використовуючи, наприклад калькулятор), які повною мірою характеризують аналізовані вирази.

4. Особливу увагу потрібно звернути на обчисленню логічного виразу в умовних операторах, які відповідають різним варіантам поєднання вхідних змінних. Надати розгорнутий варіант відповіді у вигляді відповідної таблиці істинності.

Зміст звіту

1. Титульний лист.

2. Цілі лабораторного заняття і вказівка, які навички та вміння передбачається отримати в результаті його виконання.

3. Аналіз вихідного виразу індивідуального завдання з обґрунтуванням вибору найбільш доцільною послідовністю операторів обчислень у вигляді відповідної графічної схеми.

4. Чисельні приклади виконання завдання для усіх гілок вихідного виразу.

5. Висновки.

Контрольні запитання

1. Дайте визначення поняття потоку управління програмою.

2. Що становить структура вибору if, коли її треба використовувати?

3. Опишіть структуру вибору if / else.
4. У чому полягає специфіка множинного вибору та як цей вибір доцільно реалізувати за допомогою структури switch?

Допоміжна інформація (для тих, хто вивчав мову C# або C++)

Оператор умовного переходу if...else

Синтаксис умовного оператора if

```
if(выражение1) оператор1;  
[else оператор2;]
```

Работает оператор if следующим образом. Если выражение1 = true, то выполняется оператор1, а если иначе, то оператор2.

Воспользуемся оператором if для решения задачи попадания в заданную область S (рис.2).

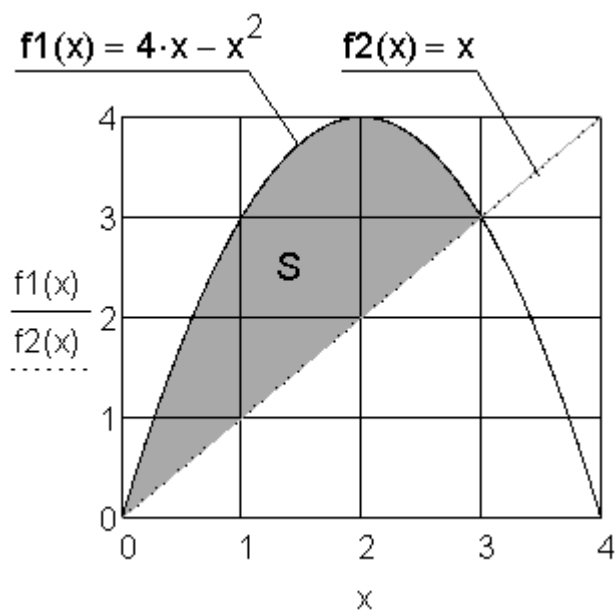


Рис.1. Область S обмежена функціями f1 та f2

Текст програми (листинг 2) та результат (рис. 2) представлені нижче.

Листинг 2. Розгалужене консольне застосування

```
using System;
```

```
namespace C2
{
    class Program
    {
        static void Main()
        {
            double x, y;
            bool u;
            Console.Write("x = ");
            x = Double.Parse(Console.ReadLine());
            Console.Write("y = ");
            y = Convert.ToDouble(Console.ReadLine());
            u = ((4 * x - Math.Pow(x,2)) > y) && (x < y);
            if (u) Console.WriteLine("x = " + x + " y = " + y + " попал");
            else Console.WriteLine("x = " + x + " y = " + y + " не попал");
        }
    }
}
```

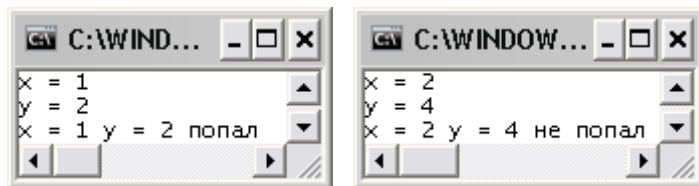


Рис. 2. Результат работы программы