

Самостійна робота 2. Розробка консольних програм, які реалізують обчислення циклічних процесів з розгалуженням

Самостійна робота 2 продовжує поглиблене вивчення теми 3: «Програмування обчислювальних процесів». Основна увага при поглибленому вивченні цієї теми приділялась наступним питанням: розробленню консольної програми, яка реалізує обчислення математичного виразу з розгалуженням згідно індивідуального варіанту; розробленню консольної програми, яка реалізує створення таблиці обчислення значень математичного виразу згідно індивідуального варіанту.

Мета роботи: отримання знань та навичок зі створення в програмному середовищі Visual Studio консольних додатків, які реалізують циклічні обчислення математичних виразів з розгалуженням.

У результаті виконання самостійної роботи у студента формуються такі **компетентності**: застосовувати алгоритмічні структури управління програмою та варіанти їх реалізації мовою C#; здатність створювати багатострокові табличні документи.

Результатом виконання самостійної роботи є налаштований консольний додаток та демонстрація його працездатності.

Завдання для самостійної роботи

Виконувати дану самостійну роботу доцільно перед виконанням лабораторної роботи «Програмування циклічних обчислювальних процесів».

У ході виконання роботи необхідно:

1. Проаналізувати математичний вираз з можливістю вибору одної з трьох гілок обчислення, який надалі буде реалізовано в середовищі Visual Studio.NET. Намалювати алгоритм його обчислення у вигляді графічної схеми. Алгоритм повинен забезпечувати формування таблиці з десяти строками, в кожній з якої наведена інформація щодо обчислення заданої функції, її поточного аргументна та номера гілки.

2. За допомогою калькулятора підготувати контрольні приклади, які дозволяють перевірити дію алгоритму по кожній з гілок обчислення.

3. Відповідно до графічної схеми алгоритму скласти перелік команд, які забезпечують формування таблиці з результатами обчислення.

4. В середовище Visual Studio.NET створити шаблон консольного додатку та набрати у вбудований в нього редактор тексту розроблену програму.

5. Виконати компіляцію, налагодження і запуск програми; отримати роздруківку вихідного тексту програми і результату її роботи.

Контрольні запитання для самодіагностики

1. Дайте 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 і як цей вибір доцільно реалізувати за допомогою структури "switch"?
5. Умовний вираз. Наведіть його синтаксис і відповідний приклад застосування.
6. Дайте загальний огляд структур повторення.
7. Опишіть особливості застосування циклу з передумовою (while). Наведіть приклад.
8. Коли доцільно застосовувати цикл з постумовою (do / while)? Наведіть приклад.
9. Дайте синтаксис оператора циклу for.
10. Наведіть загальні рекомендації відносно вибору циклів.

Методичні рекомендації до теми

Самостійна робота повинна виконуватися в наступній послідовності.

1. Аналіз математичного виразу, обчислення якого надалі буде реалізовано в середовище Visual Studio.NET.

Наприклад, необхідно обчислити і вивести на екран у вигляді таблиці значення функції F на інтервалі від Xпочат. до Xкінц. з кроком dX

$$F = \begin{cases} a \cdot x^2 + b / c, & \text{при } x < 1 \text{ і } c \neq 0, \\ (x - a) / (x - c)^2, & \text{при } x > 1,5 \text{ і } c = 0, \\ a / x^2, & \text{в інших випадках,} \end{cases}$$

де a, b, c - дійсні числа. Причому, функція F повинна приймати дійсне значення, якщо вираз $(A_c \& B_c) \text{ MOD } 2 \ C_c$ не дорівнює нулю, і ціле значення в іншому випадку. Через A_c, B_c та C_c позначені цілі частини значень a, b, c , операції $\&$ і $\text{MOD } 2$ (додавання по модулю 2) - порозрядні.

Як результат виконання цього пункту, - необхідно отримати набір чисельних прикладів, який охоплює дію алгоритму обчислення по кожній з трьох гілок вихідного виразу. Крім цього, слід передбачити вихідні дані, які призводять до особливих випадків (наприклад, ділення на нуль).

2. Побудова графічної схеми алгоритму обчислення і написання коду програми.

На даному етапі необхідно вибрати дві графічні алгоритмічні схеми – одну для зображення процесу що розгалуження на відповідні гілки, а другу – для організації циклічного формування строк кінцевої таблиці.

Якщо для розгалуження вибрана графічна схема з двома гілками (їй відповідає оператора `if – else`), а для циклічного повтору – схема циклу з передумовою (йому відповідає оператора `for`) то фрагмент коду, який забезпечує обчислення функції F може бути таким:

```
...
double a, b, c, x0, xN, dx, F;
for (double i = x0; i <= xN; i = i + dx)
{
    if (i < 1 && c != 0)
    {
        F = a * Math.Pow(i, 2) + b / c;
    }
    else if (i > 1.5 && c == 0)
    {
        F = (i - a) / Math.Pow(i - c, 2);
    }
    else
    {
        F = a / Math.Pow(i, 2);
    }
    if (((A & B) ^ C) != 0)
    {
        double F1 = F;
        Console.WriteLine(F1);
    }
    else
```

```
{  
    int F2 = Convert.ToInt32(F);  
    Console.WriteLine(F2);  
}
```

...

3. В середовище Visual Studio.NET створити шаблон консольного додатку та набрати у вбудований в нього редактор тексту розроблену програму.

5. Виконати компіляцію, налагодження і запуск програми; отримати роздруківку вихідного тексту програми і результату її роботи.

Література: [1; 3].