Рекурсивні функції. Перевантаження функцій

Лектор:
к.т.н., доц., доцент кафедри прикладної математики
Рижа Ірина Андріївна

Про що ця лекція???

- Розглянемо функції із аргументами за замовчуванням та рекурсивні функції.
- Опишемо особливості перевантаження функцій у C++.

Аргументи за замовчуванням

Параметри за замовчуванням

дозволяють оголосити функцію таким чином, що звертання до неї матиме меншу кількість фактичних параметрів.

- Параметрами за замовчуванням можуть бути крайні праві параметри у списку аргументів.
- Не може бути параметром за замовчуванням той аргумент, справа від якого формальний параметр не є параметром за замовчуванням.

Наприклад

```
int st(int, int = 2); //так можна!
void func1(int, double = 1.5, char = ' '); //так можна!
int st(int = 1, int); //так НЕ можна!
void func1(int = 2, double, char = ' '); //так НЕ можна!
```

Аргументи за замовчуванням

- Задання параметрів за замовчуванням є доречним тоді, коли деякі аргументи дуже часто використовуються в коді програми з однаковим значенням.

- Значення за замовчуванням також можна використовувати, якщо програміст хоче модифікувати вже написану функцію, додаючи до неї новий аргумент. В такому випадку не потрібно буде змінювати виклики функції, адже значення нового аргументу буде значенням за замовчуванням.

Приклад 1

Написати функцію обчислення цілого невід’ємного степеня цілого числа, де в якості параметра за замовчуванням є показник степеня.

Рекурсивні функції

Рекурсивна функція

— функція, серед виконавчих інструкцій якої є інструкція чи інструкції виклику функцією самої себе з відповідними параметрами.

Переваги рекурсії

- зменшує код програми;
- робить код прозорішим і зрозумілішим.

До чого зобов'язує рекурсія???

- Слід встановити *рекурентне співвідношення*, тобто закон звертання функції самої до себе
- Обов’язково потрібно визначити *умову завершення* виконання функції, тобто умову припинення звертання функції самої до себе.

Приклад 2

Написати рекурсивну функцію обчислення факторіала невід’ємного цілого числа.

Рекурсивні функції

Чи всі рекурсивні виклики функції зберігатимуться у пам'яті комп'ютера???

- У пам'яті міститимуться значення параметрів для кожного виклику, але тіло функції буде присутнім у пам'яті в одиничному екземплярі.
- Проте програма, що використовує рекурсії зі значною кількістю вкладених викликів, може вичерпати усі ресурси оперативної пам'яті, що спричинить проблеми у роботі програмного середовища чи навіть операційної системи.

Рекурсивні функції

Послідовність чисел Фібоначчі

$$f_n = f_{n-1} + f_{n-2}, \qquad n \geq 2,$$
$$f_0 = f_1 = 1.$$

```
unsigned fibonacci(unsigned n) //рекурсивний варіант
{
     if (n < 2)
     return 1;
     return (fibonacci(n - 1) + fibonacci(n - 2));
}

unsigned fibonacci_iter(unsigned n) //ітераційний варіант
{
     unsigned f1, f2, f0, i = 2; f2 = f1 = 1;
     for (; i <= n; i++)
     {
     f0 = f1; f1 = f2; f2 += f0;
     }
     return f2;
}
```

Перевантаження функцій

Функції у С

- У мові програмування С ім’я кожної функції є унікальним.

Наприклад, функції обчислення модуля числа є різними:

`abs(a)` — повертає ціле значення типу `int` модуля числа незалежно від типу аргументу;

`fabs(a)` — повертає дійсне значення (`float`) модуля числа;

`labs(a)` — повертає довге ціле (`long`) число.

Перевантаження функцій

Функції у C++

- Користувач може створити дві й більше функцій, які мають однакові імена, але компілятором сприйматимуться по різному.

Перевантаження функції (overload)

— переозначення функцій з однаковими іменами за умови, що список їхніх параметрів є різним (типи параметрів, їх кількість чи порядок).

- При цьому тип функції (тип результату), який повертає функція, не визначає можливості перевантаження.

Наприклад, функція `abs(a)` із заголовкового файлу `math.h` (`cmath`) є перевантаженою у середовищі C++ MSVS, тобто повертає абсолютне значення того типу, якого є аргумент.

Приклад 3

Розглянемо приклад перевантаження функції обчислення середнього арифметичного елементів масиву та виведення їх на екран.

Перевантаження функцій

Зауваження

При перевантаженні функцій, у яких є параметри за замовчуванням, компілятор іноді може не «знати», який саме варіант функції обрати, тому фіксуватиме синтаксичну помилку.

Приклад 4

Записати функцію обчислення визначеного інтегралу $\int_a^b f(x)dx$ методом прямокутників із заданим числом $n + 1$ точок розбиття інтервалу $[a; b]$.

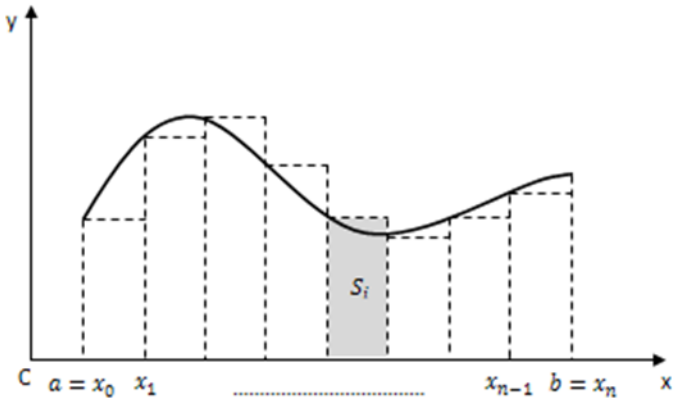
Використовуючи цю підпрограму обчислити інтеграл $\int_a^b e^{-x^2}dx$ із заданою точністю ε.

Ідея алгоритму

Нехай на відрізку $[a; b]$ задана неперервна функція $f(x)$. Потрібно обчислити методом прямокутників інтеграл:

$$\int_a^b f(x)dx.$$

Задане число вузлів $n + 1$ розбиває проміжок інтегрування $[a; b]$ на n рівних відрізків $[x_i; x_{i+1}]$, $i = \overline{0, n-1}$ довжиною $h = \frac{b-a}{n}$. На кожному відрізку $[x_i; x_{i+1}]$ функція $f(x)$ замінюється сталою $f(x_i)$. Це означає, що площу криволінійної трапеції можна замінити площею n прямокутників.



Ідея алгоритму

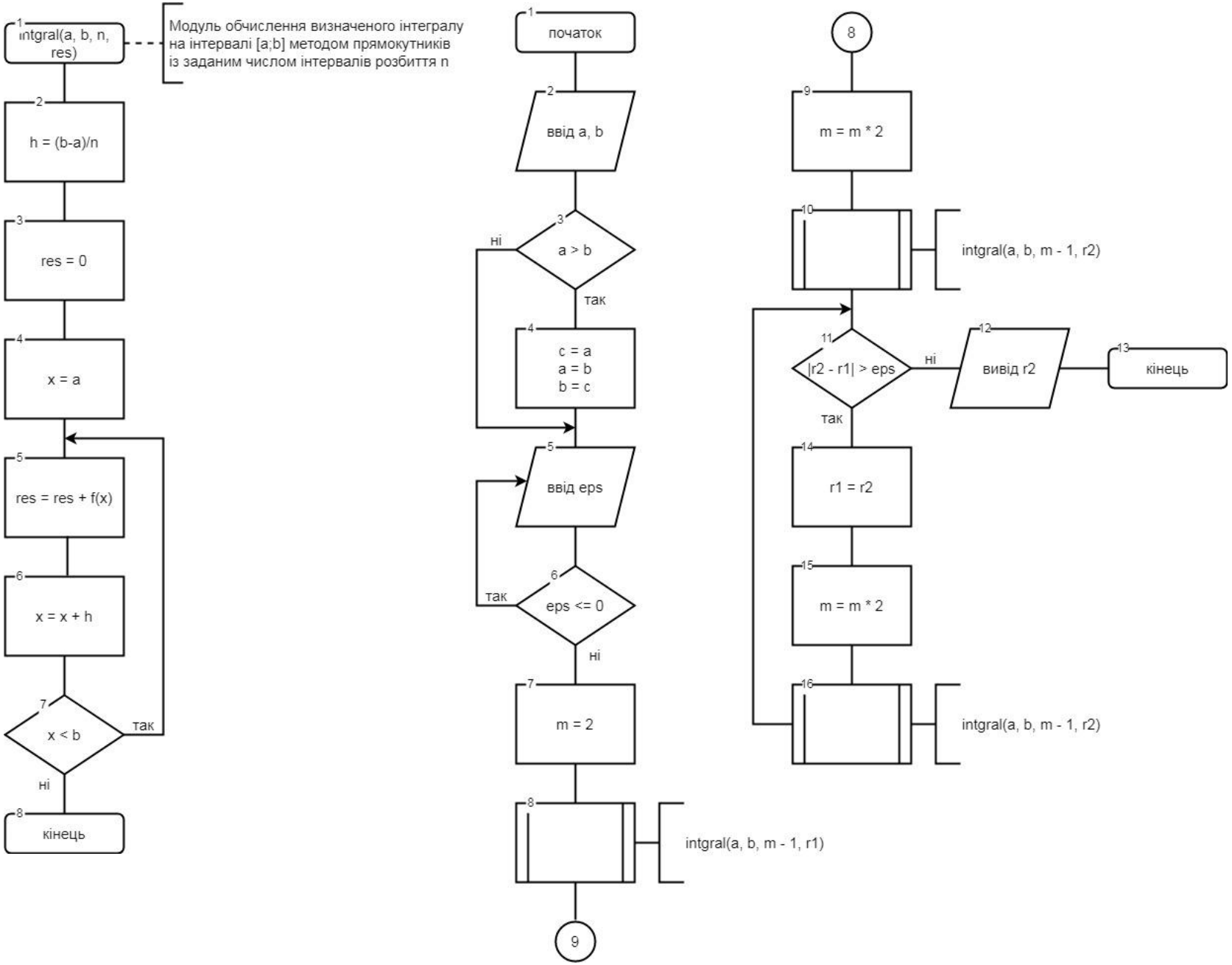
Площу одного такого прямокутника можна обчислити за формулою:

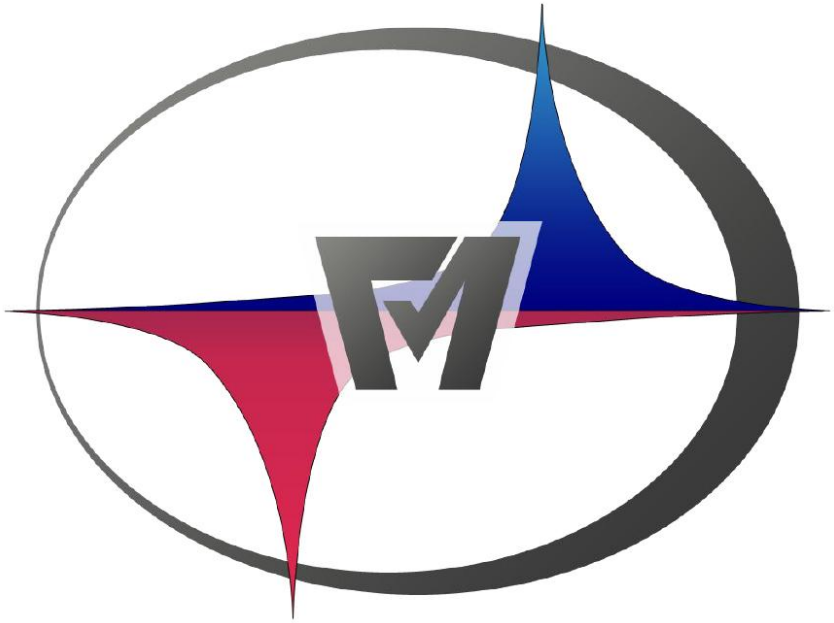
$$S_i = (x_{i+1} - x_i) f(x_i) = h f(x_i), \qquad i = \overline{0, n-1}.$$

Тоді загальна площа прямокутників і, відповідно, значення інтеграла дорівнює:

$$I_n = \sum_{i=0}^{n-1} S_i = \sum_{i=0}^{n-1} h f(x_i) = h \sum_{i=0}^{n-1} f(x_i) = h \sum_{i=0}^{n-1} f(a + ih)$$

- Щоби досягнути заданої точності ε, процес потрібно продовжувати до тих пір, поки $|I_n - I_{2n}|$ не стане меншим за ε.


1
intgral(a, b, n, res)
Модуль обчислення визначеного інтегралу на інтервалі [a;b] методом прямокутників із заданим числом інтервалів розбиття n
2
h = (b-a)/n
3
res = 0
4
x = a
5
res = res + f(x)
6
x = x + h
7
x < b
так
ні
8
кінець
1
початок
2
ввід a, b
3
a > b
ні
так
4
c = a
a = b
b = c
5
ввід eps
6
eps <= 0
так
ні
7
m = 2
8
intgral(a, b, m - 1, r1)
9
8
9
m = m * 2
10
intgral(a, b, m - 1, r2)
11
|r2 - r1| > eps
ні
так
12
вивід r2
13
кінець
14
r1 = r2
15
m = m * 2
16
intgral(a, b, m - 1, r2)



Кафедра прикладної математики

http://amath.lp.edu.ua