ПРАКТИЧНА РОБОТА №10

Перевантаження функцій у C++

Виконав: студент групи ALK-43
Вівчар Вадим Вікторович

Мета роботи

Ознайомитись із механізмом перевантаження функцій у мові C++, навчитись створювати декілька функцій з однаковим ім’ям, але різними параметрами, та продемонструвати роботу цього механізму на практиці.

Теоретичні відомості

Перевантаження функцій (function overloading) — це механізм, що дозволяє створювати кілька функцій із однаковим ім’ям, які відрізняються кількістю або типом аргументів. Компілятор автоматично визначає, яку з функцій потрібно викликати, залежно від кількості чи типів переданих аргументів. Це один із способів реалізації статичного

поліморфізму в C++.

Варіант 1. Єдиний файл (обидва типи перевантаження)

```
#include <iostream>
#include <iomanip>
#include <limits>
using namespace std;

int add(int a, int b) { return a + b; }
double add(double a, double b) { return a + b; }
double area(double a) { return a * a; }
double area(double a, double b) { return a * b; }

int main() {
    cout << fixed << setprecision(2);
    int ix = 5, iy = 12;
    double dx = 2.5, dy = 4.3;
    cout << "=== Overloading by TYPE ===\n";
    cout << "add(int,int): add(" << ix << ", " << iy << ") = " << add(ix, iy) << "\n";
    cout << "add(double,double): add(" << dx << ", " << dy << ") = " << add(dx, dy) << "\n";
    cout << "add((double)ix, dy) = " << add(static_cast<double>(ix), dy) << "\n\n";
    double a = 5.0, b = 8.0;
    cout << "=== Overloading by NUMBER OF PARAMETERS ===\n";
    cout << "area(a) [square], a = " << a << " -> " << area(a) << "\n";
    cout << "area(a,b) [rectangle], a = " << a << ", b = " << b << " -> " << area(a, b) << "\n";
    cout << "\nDone. Press Enter to exit...";
    cin.ignore(numeric_limits<streamsize>::max(), '\n');
    cin.get();
    return 0;
}
```

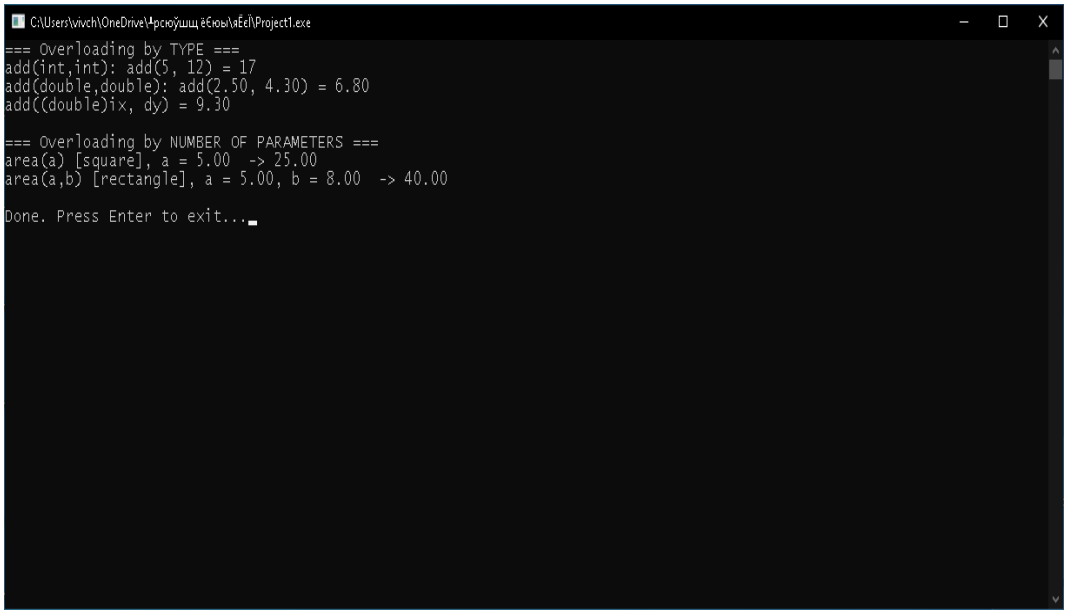

Рис. 1 — Результат виконання програми (обидва типи перевантаження)

Варіант 2. Перевантаження за типом аргументів

```
#include <iostream>
```

```
using namespace std;

int add(int a, int b) { return a + b; }
double add(double a, double b) { return a + b; }

int main() {
    int x = 7, y = 3;
    double p = 1.25, q = 4.75;
    cout << "add(int,int) = " << add(x, y) << "\n";
    cout << "add(double,double) = " << add(p, q) << "\n";
    return 0;
}
```

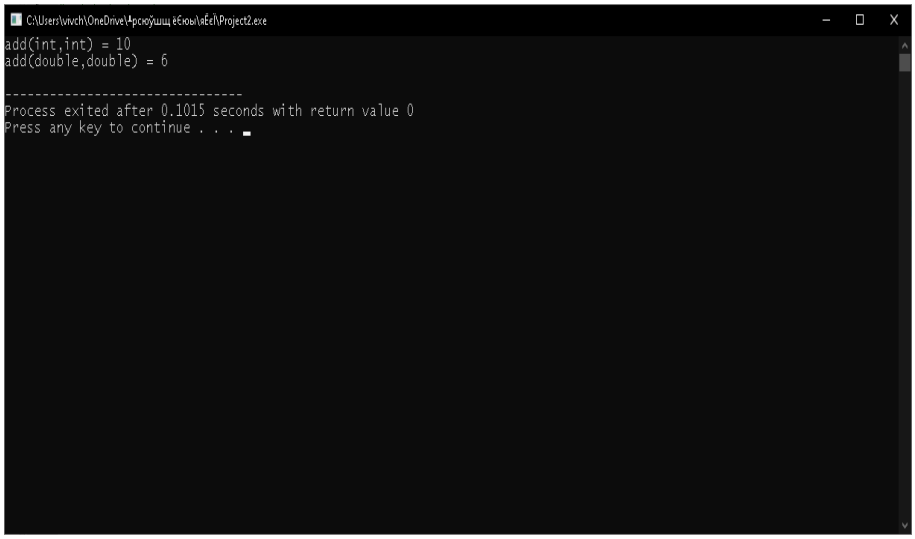

Рис. 2 — Результат виконання програми add()

Варіант 3. Перевантаження за кількістю аргументів

```
#include <iostream>
using namespace std;

double area(double a) { return a * a; }
double area(double a, double b) { return a * b; }

int main() {
    double side = 6.0;
    double len = 4.0, wid = 7.0;
    cout << "area(square, a) = " << area(side) << "\n";
    cout << "area(rectangle, a,b) = " << area(len, wid) << "\n";
    return 0;
}
```

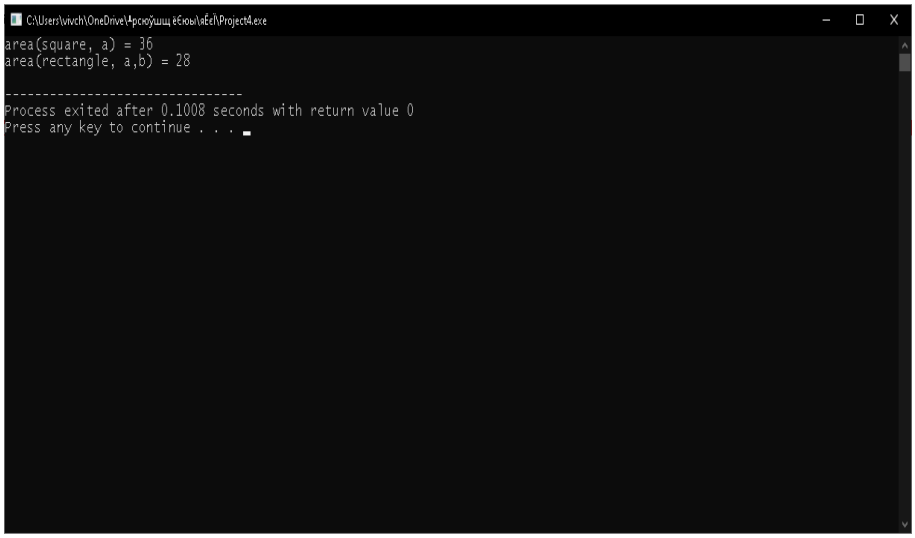

Рис. 3 — Результат виконання програми area()

Висновок

У ході практичної роботи створено три приклади програм, що демонструють механізм перевантаження функцій у C++: у спільному файлі (два типи перевантаження одночасно), окремо за типом аргументів і окремо за кількістю параметрів. Отримані результати підтверджують правильність роботи коду та ефективність використання перевантаження функцій для спрощення синтаксису і підвищення читабельності програм.