

Міністерство освіти і науки України  
Прикарпатський національний університет імені Василя Стефаника  
Фізико-технічний факультет  
Кафедра комп'ютерної інженерії та електроніки

Звіт  
з дисципліни «Програмування. Частина 1» по  
виконанню лабораторної роботи №8  
на тему:  
“ФУНКЦІЇ, ЩО НЕ ПОВЕРТАЮТЬ ЗНАЧЕННЯ ”

Виконав:  
студент групи КІ-12  
Лісевич Руслан

**Мета роботи:** навчитися використовувати власні функції користувача, що не повертають значення для розв'язування задач.

### Завдання 1:

1. Створіть еквівалент калькулятора, що виконує чотири основних арифметичних операції. Програма повинна запрошувати ввід користувачем першого операнда, знаку операції і другого операнда, які передаються у якості параметрів функції, що не повертає значення. Для зберігання операндів слід використовувати змінні дійсного типу. Перевірити вибір операції можна за допомогою оператора if. У кінці програма повинна відображати результат на екрані. Результат роботи програми

може виглядати таким чином:

Введіть перший операнд, операцію і другий операнд: 10/3 Результат дорівнює 3.333333

Виконати ще одну операцію (y/n)? y

Введіть перший операнд, операцію і другий операнд: 12 + 100

Результат дорівнює 112

Виконати ще одну операцію (y/n)? N

### Лістинг виконавчого коду програми з коментарями

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

void Calc() {
    double operand1, operand2;
    char operation;

    cout << "Введіть перший операнд, операцію і другий операнд: ";
    cin >> operand1 >> operation >> operand2;

    double result;

    if (operation == '+') {
        result = operand1 + operand2;
    }
}
```

```

    else if (operation == '-') {
        result = operand1 - operand2;
    }
    else if (operation == '*') {
        result = operand1 * operand2;
    }
    else if (operation == '/') {
        if (operand2 != 0) {
            result = operand1 / operand2;
        }
        else {
            cout << "Ділення на нуль неможливе." << endl;
            return;
        }
    }
    else {
        cout << "Невірна операція." << endl;
        return;
    }

    cout << "Результат дорівнює " << result << endl;
}

int main() {
    setlocale(LC_ALL, "UKR");
    char choice;

    do {
        Calc();

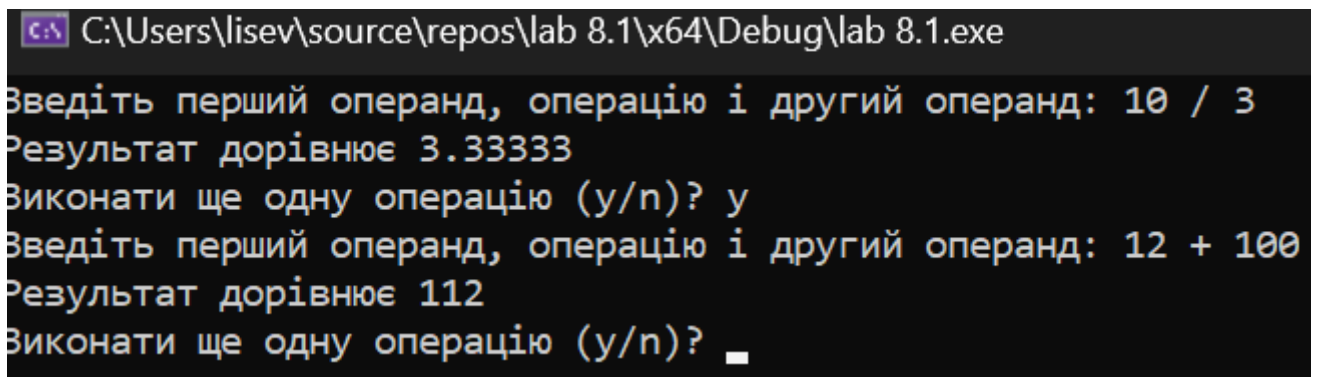
        cout << "Виконати ще одну операцію (y/n)? ";
        cin >> choice;

    } while (choice == 'y' || choice == 'Y');

    return 0;
}

```

## Результати виконання програми



```

C:\Users\lisev\source\repos\lab 8.1\x64\Debug\lab 8.1.exe
Введіть перший операнд, операцію і другий операнд: 10 / 3
Результат дорівнює 3.33333
Виконати ще одну операцію (y/n)? y
Введіть перший операнд, операцію і другий операнд: 12 + 100
Результат дорівнює 112
Виконати ще одну операцію (y/n)? _

```

## Завдання 2:

2. Створіть програму, що викликає функцію, яка приймає два аргументи. Перший аргумент символ, а другий аргумент типу int не рівний нулю. Функція

виводить символ стільки разів, скільки рівне значення другого введенного параметра.

### Лістинг виконавчого коду програми з коментарями

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

void PrintSymbol(char symbol, int count) {
    for (int i = 0; i < count; i++) {
        cout << symbol;
    }
    cout << endl;
}

int main() {
    setlocale(LC_ALL, "UKR");
    char symbol;
    int count;

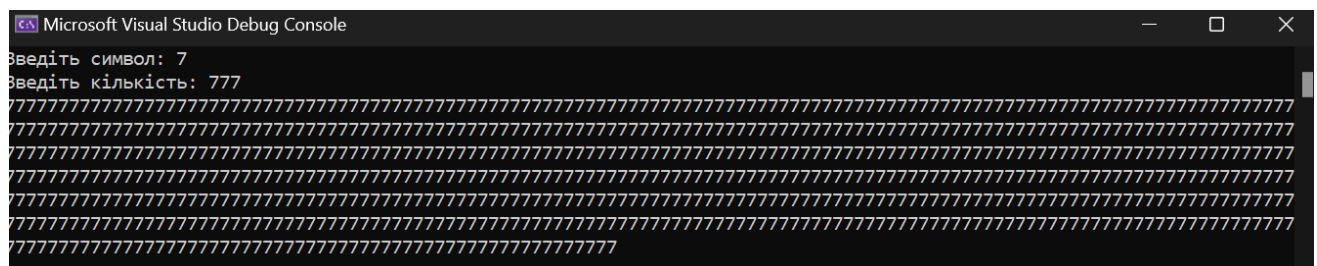
    cout << "Введіть символ: ";
    cin >> symbol;

    cout << "Введіть кількість: ";
    cin >> count;

    if (count != 0) {
        PrintSymbol(symbol, count);
    }
    else {
        cout << "Кількість повинна бути не нуль." << endl;
    }

    return 0;
}
```

### Результати виконання програми



**Висновки:** цій роботі ми навчилися використовувати власні функції користувача, які не повертають значення, для розв'язування задач. Використання таких функцій дозволяє структурувати програму, роблячи її більш зрозумілою та керованою.

Ми приділили увагу основним принципам роботи з власними функціями без повернення значення, навчилися їх визначати та викликати. Також ми зрозуміли важливість розділення функцій на більш маленькі та зрозумілі блоки коду, що полегшує розробку та підтримку програми.