Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Кафедра систем управления

Отчет по лабораторной работе № 6

Тема: «Программирование с использованием функций»

Вариант 7

Выполнил:

студент группы 322441 Калинин В.А.

Проверил:

зам.каф ИТАС Навроцкий А.А.

Минск

2023

Цель работы

Цель работы: изучить правила и принцип составления функций; написать и отладить программу с использованием функций.

**Краткие теоретические сведения**

Функция – последовательность операторов, оформленная таким образом, что ее можно вызвать по имени из любого места программы.

Функция описывается следующим образом:

тип\_возвращаемого\_значения имя\_функции (список\_параметров)

{

Тело\_функции

}

Первая строка данного описания называется заголовком функции. Тип возвращаемого значения может быть любым, кроме массива или функции. Если функция не возвращает значение, то указывается тип void. В С++ по умолчанию тип возвращаемого результата − int.

Список параметров представляет собой набор конструкций следующей формы:

тип\_параметра имя\_параметра

Например:

int sum(int a, double b, char c)

Если функция не получает никаких данных, то скобки остаются пустыми:

int fun()

Широко используются прототипы функций (их предварительное объявление). Прототип аналогичен заголовку функции, за исключением того, что имена формальных параметров не указываются (остаются только типы), и в конце ставится точка с запятой:

int sum(int, double, char);

Широкое использование прототипов вызвано следующим:

* имеющие прототипы функции могут быть вызваны из других модулей;
* использование прототипов позволяет размещать функции в произ- вольном порядке (а не до первого их использования);
* размещение прототипов в одном месте делает программу более читабельной.

Правила оформления тела функции такие же, как и для любого другого участка программы. Все объявления носят локальный характер, т. е. объявленные переменные доступны только внутри функции.

Не допускается вложение функций друг в друга.

Выход из функции происходит при достижении закрывающей функцию скобки или после выполнения оператора return.

**Передача параметров**

При работе должно соблюдаться следующее правило: при объявлении и вызове функции параметры должны соответствовать по количеству, порядку следования и типам. Существует три основных способа передачи параметров: передача по значению, по ссылке и по указателю.

8.2.1. Передача параметров по значению

В функции создаются временные переменные, в которые передаются значения из вызывающей функции. Например:

int fun1(double, int, char); // Прототип функции

…

int fun1(double a, int b, char c) // Заголовок функции

{

Тело\_функции

}

…

int s = fun1(d, 8, chr); // Вызов функции

В момент обращения к функции в памяти создаются временные переменные с именами a, b, c. В созданные переменные копируются значения: d, 5, chr. После этого связь между передаваемыми и временными переменными разрывается.

8.2.2. Передача параметров по ссылке

В функцию передаются адреса переменных из основной программы. Ссылочный параметр – псевдоним соответствующего аргумента. Для получения адреса используется операция «взять адрес». Например:

void fun2(double &, int &); // Прототип функции

…

void fun2(double &a, int& b) // Заголовок функции

{

Тело\_функции

}

…

fun2(d, r); // Вызов функции

При таком вызове передается не переменная, а ее адрес, полученный с использованием операции «взять адрес». Следовательно, при обращении к временной переменной в действительности происходит обращение к исходной переменной в вызывающей функции.

**Передача параметров по указателю**

При вызове функции в качестве аргумента передается не сама переменная, а ее адрес. Например:

void fun3(double \*, int \*); // Прототип функции

…

void fun3(double \*a, int\* b) // Заголовок функции

{

Тело\_функции

}

…

fun3(&f,&k); // Вызов функции

Задание

Задание: вывести на экран таблицу значений функции и ее разложения в ряд для х, изменяющегося от a до b с шагом h = (b - a)/10. Расчет у(х) и s(x) поместить в функцию. Использовать прототипы функций. Параметры передавать указанным способом. Расчет функции s(x) выполнить с заданной точностью ε.





Листинг программы

#include <iostream>

#include <cmath>

#include <iomanip>

using namespace std;

double calculateY(double &x) {

return cos(sin(x));

}

double calculateSx(double &x, double &e) {

double result = 1, result\_sum=1;

for (int j = 1; abs(result) > e ; j++) {

result \*= -1 \* sin(x) \* sin(x) / (2 \* j \* (2 \* j - 1)) ;

result\_sum += result;

}

return result\_sum;

}

int main() {

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Lab 6 var 7 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

double a = -0.5;

double b = 0.5;

double e = 1e-4;

double h = (b-a) / 10;

cout << "x" << setw(9) << "\t\t" << setw(9) << "y(x)" << "\t\t" << setw(9) << "s(x, e)" <<"\n";

for (double x = a; x <= b; x += h) {

if (x < e && x > -e) { x = 0;}

cout << x << setw(9) << "\t\t" << setw(9) << calculateY(x) << "\t\t" << setw(9) << calculateSx(x,e) <<"\n";

}

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

return 0;

}

Результат работы программы

