ПРИЛОЖЕНИЕ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»

(СПбГУТ)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Кафедра Систем обработки данных

**Дисциплина «Технологии программирования»**

**ОТЧЕТ**

**за практическое занятие №2**

**Тема: Взаимодействие функций в программах**

Выполнил

Студент 2 курса, гр. ИБ-32вп

\_\_\_\_\_\_\_\_\_\_\_\_\_ Братишкин Д.Е.

Принял

Доцент кафедры БИС

\_\_\_\_\_\_\_\_\_\_\_\_\_ Медведев В.А.

Санкт-Петербург

2024

Тема: Взаимодействие функций в программах

Цель: Получение практических навыков проектирования взаимодействия функций в программах с использованием локальных и глобальных переменных.

Задание на практическое занятие

Вариант A.

Задание 1:

Задать данные указанного типа (`unsigned char`, `float`), вычислить их сумму (в типе `int`) и определить размер отведённой для неё памяти. Вывести на экран значение суммы.

Задание 2:

Заполнить пятиэлементный массив (`double`) с клавиатуры и определить вычисляемый показатель (сумма положительных элементов массива). Каждое задание оформить в виде отдельной функции.

Задание 3:

Объединить обе версии в одну программу. Разработать меню для управления программой, в котором по желанию пользователя обеспечить:

- Выполнение любого из заданий для любой из версий и в любом порядке.

- Выход из программы.

Код программы

#include <iostream>

#include <limits>

using namespace std;

unsigned char globalChar;

float globalFloat;

double globalArray[5];

double globalPositiveSum = 0;

int globalSum = 0;

void sumGlobalValues() {

cout << "Введите значение для unsigned char (0-255): ";

int input;

cin >> input;

if (cin.fail() || input < 0 || input > 255) {

cout << "Ошибка: введите значение в диапазоне от 0 до 255." << endl;

return;

}

globalChar = static\_cast<unsigned char>(input);

cout << "Введите значение для float: ";

cin >> globalFloat;

globalSum = static\_cast<int>(globalChar + globalFloat);

}

void calculateGlobalPositiveSum() {

globalPositiveSum = 0;

cout << "Введите 5 элементов массива типа double:" << endl;

for (int i = 0; i < 5; i++) {

cin >> globalArray[i];

if (globalArray[i] > 0) {

globalPositiveSum += globalArray[i];

}

}

}

int calculateLocalSum(unsigned char a, float b) {

return static\_cast<int>(a + b);

}

double calculateLocalPositiveSum(double arr[]) {

double positiveSum = 0;

for (int i = 0; i < 5; i++) {

if (arr[i] > 0) {

positiveSum += arr[i];

}

}

return positiveSum;

}

int main() {

int task, version;

bool end = false;

do {

cout << "Выберите задание:\n1 - Задание 1\n2 - Задание 2\n0 - Выход\n";

cin >> task;

if (task == 0) {

end = true;

break;

}

if (task != 1 && task != 2) {

cout << "Ошибка: неверный выбор задания!" << endl;

continue;

}

cout << "Выберите версию:\n1 - Локальные переменные\n2 - Глобальные переменные\n";

cin >> version;

switch (task) {

case 1:

if (version == 1) {

unsigned char a;

float b;

cout << "Введите значение для unsigned char (0-255): ";

int input;

cin >> input;

if (cin.fail() || input < 0 || input > 255) {

cout << "Ошибка: введите значение в диапазоне от 0 до 255." << endl;

break;

}

a = static\_cast<unsigned char>(input);

cout << "Введите значение для float: ";

cin >> b;

int localSum = calculateLocalSum(a, b);

cout << "Сумма (int): " << localSum << endl;

} else if (version == 2) {

sumGlobalValues();

cout << "Сумма (int): " << globalSum << endl;

} else {

cout << "Ошибка: неверный выбор версии!" << endl;

}

break;

case 2:

if (version == 1) {

double localArray[5];

cout << "Введите 5 элементов массива типа double:" << endl;

for (int i = 0; i < 5; i++) {

cin >> localArray[i];

}

double localPositiveSum = calculateLocalPositiveSum(localArray);

cout << "Сумма положительных элементов массива: " << localPositiveSum << endl;

} else if (version == 2) {

calculateGlobalPositiveSum();

cout << "Сумма положительных элементов массива: " << globalPositiveSum << endl;

} else {

cout << "Ошибка: неверный выбор версии!" << endl;

}

break;

}

} while (!end);

return 0;

}

Результаты работы программы

Задание 1:

1. Ввод: 200, 10.5

- Вывод: Сумма (int): 200

- Вывод: Размер суммы (в байтах): 4

2. Ввод: 50, 3.14

- Вывод: Сумма (int): 50

- Вывод: Размер суммы (в байтах): 4

Задание 2:

1. Ввод: 1.5, 2.3, -3.7, 4.1, 5.9

- Вывод: Сумма положительных элементов массива: 13.8

2. Ввод: -1.2, -3.4, 0, 5.6, -7.8

- Вывод: Сумма положительных элементов массива: 5.6

Выводы по разработанным элементам программы

Программа на C++ предоставляет выбор между двумя задачами: вычисление суммы и суммы положительных элементов массива, с вариантами использования локальных или глобальных переменных.

- Глобальные переменные:

- globalChar (unsigned char) - хранит значение 0–255.

- globalFloat (float) - хранит число с плавающей запятой.

- globalArray (double[5]) - массив из 5 элементов типа double.

- globalSum (int) - хранит сумму globalChar и globalFloat.

- globalPositiveSum (double) - хранит сумму положительных значений массива globalArray.

- Функции:

- sumGlobalValues() - обновляет globalChar и globalFloat, вычисляет их сумму и сохраняет в globalSum.

- calculateGlobalPositiveSum() - заполняет globalArray и вычисляет сумму положительных элементов, сохраняя ее в globalPositiveSum.

- calculateLocalSum(unsigned char, float) - принимает локальные значения и возвращает их сумму.

- calculateLocalPositiveSum(double[]) - принимает локальный массив и возвращает сумму положительных элементов.

- В main() пользователь выбирает задачу и версию (локальные или глобальные переменные).

- В версии с локальными переменными значения вводятся и передаются в функции, которые возвращают результат.

- В версии с глобальными переменными значения обновляются в глобальных переменных, результат сохраняется в соответствующей глобальной переменной и выводится.

Программа оформлена в виде отдельных функций, что делает код более структурированным и легким для понимания и отладки. Обработка ошибок и ввод данных пользователем выполнены корректно, что повышает надежность и удобство использования программы.