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;

// Глобальные переменные для второй версии

unsigned char globalA;

float globalB;

double globalArray[5]; // Глобальный массив для хранения элементов

double globalPositiveSum = 0; // Глобальная переменная для суммы положительных элементов

int globalCount = 0; // Счетчик для глобального массива

// Функция 1: Вычисление суммы и размер памяти

void task1(bool useGlobal) {

    if (useGlobal) {

        // Используем глобальные переменные

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

        cin >> globalA;

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

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

            cin.clear();

            cin.ignore(numeric\_limits<streamsize>::max(), '\n');

            return;

        }

        cout << "Введите значение для float: ";

        cin >> globalB;

        // Вычисляем сумму

        float sum = globalA + globalB;

        int sumInt = static\_cast<int>(sum);

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

        cout << "Размер суммы (в байтах): " << sizeof(sum) << endl;

    } else {

        // Используем локальные переменные

        unsigned char a;

        float b;

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

        cin >> a;

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

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

            return;

        }

        cout << "Введите значение для float: ";

        cin >> b;

        // Вычисляем сумму

        float sum = a + b;

        int sumInt = static\_cast<int>(sum);

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

        cout << "Размер суммы (в байтах): " << sizeof(sum) << endl;

    }

}

// Функция 2: Заполнение массива и вычисление суммы положительных элементов

void task2(bool useGlobal) {

    if (useGlobal) {

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

        globalPositiveSum = 0; // Обнуляем сумму перед вводом

        for (globalCount = 0; globalCount < 5; globalCount++) {

            cin >> globalArray[globalCount];

            // Если число положительное, добавляем его к сумме

            if (globalArray[globalCount] > 0) {

                globalPositiveSum += globalArray[globalCount];

            }

        }

        // Выводим сумму положительных элементов

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

    } else {

        // Локальные переменные

        double localArray[5];

        double positiveSum = 0; // Сумма положительных чисел

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

        for (int i = 0; i < 5; i++) {

            cin >> localArray[i];

            // Если число положительное, добавляем его к сумме

            if (localArray[i] > 0) {

                positiveSum += localArray[i];

            }

        }

        // Выводим сумму положительных элементов

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

    }

}

// Основная функция

int main() {

    char choice;

    int task, version;

    do {

        // Меню для выбора задания и версии

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

        cin >> task;

        if (task == '0') break; // Выход из программы

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

        cin >> version;

        switch (task) {

            case '1':

                task1(version == 2);

                break;

            case '2':

                task2(version == 2);

                break;

            default:

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

                break;

        }

    } while (true);

    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 успешно выполняет поставленные задачи. В первом задании она корректно вычисляет сумму значений типа `unsigned char` и `float`, преобразует результат в `int` и выводит его, а также определяет размер памяти, занимаемой суммой. Во втором задании программа заполняет массив из пяти элементов типа `double` с клавиатуры и вычисляет сумму положительных элементов массива, выводя результат на экран.

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.