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 С.

Задание 1. Задать данные указанного типа (unsigned char, float), вычислить их сумму (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.

Код программы

#include <iostream>

#include <limits>

using namespace std;

void task1() {

    unsigned char a;

    float b;

    // запрашиваем у пользователя значение для типа unsigned char

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

    cin >> a;

    // проверяем, если ввод некорректен или значение вне диапазона

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

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

        return;

    }

    // запрашиваем значение для переменной типа float

    cout << "Введите значение для float: ";

    cin >> b;

    // вычисляем сумму a и b

    float sum = a + b;

    int sumInt = static\_cast<int>(sum); // приводим сумму к типу int

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

    // выводим размер переменной sum в байтах

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

}

void task2() {

    const int MAX\_SIZE = 100;

    double arr[MAX\_SIZE]; // массив для хранения элементов

    int count = 0;

    double value;

    double positiveSum = 0; // переменная для суммы положительных чисел

    // запрашиваем ввод элементов массива

    cout << "Введите элементы массива типа double (введите нечисловое значение для завершения ввода):" << endl;

    while (count < MAX\_SIZE && cin >> value) {

        arr[count++] = value;

        // если число положительное, добавляем его к сумме

        if (value > 0) {

            positiveSum += value;

        }

    }

    // очищаем поток ввода

    cin.clear();

    // игнорируем остаток строки после завершения ввода

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

    // выводим сумму положительных элементов

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

}

int main() {

    task1();

    task2();

    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.