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 №3**

**Тема: Ввод-вывод и форматирование данных**

Выполнил

Студент 2 курса, гр. ИБ-32вп

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

Принял

Доцент кафедры БИС

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

Санкт-Петербург

2024

Тема: Ввод-вывод и форматирование данных

Цель: Получение практических навыков работы с файлами и использования средств форматирования данных.

Задание на практическое занятие

Вариант A.

1. Задание 1: Используя манипуляторы ввода-вывода, в цикле сформировать на экране матрицу согласно индивидуальному номеру.

2. Задание 2: Разработать функцию преобразования исходной строки, используя только библиотечные функции, без обращения к элементам символьного массива по индексу.

3. Задание 3: Спроектировать структуру, описывающую характеристики «Предмета». Создать массив из 5-ти структурных переменных и записать их в файл, определив вычисляемый показатель.

4. Задание 4: Разработать меню, в котором по желанию пользователя можно выполнить любой пункт задания, прочитать из файла информацию о любом из «предметов» или выйти из программы.

Код программы

#include <iostream>

#include <iomanip>

#include <fstream>

#include <cstring>

#include <string>

#include <sstream>

#include <vector>

using namespace std;

// Функция для вывода матрицы с использованием манипуляторов

void printMatrix(int n) {

for (int i = n; i >= 1; i--) {

cout << setw(3) << i; // Применяем setw для выравнивания чисел

for (int j = 1; j <= n; j++) {

if (j <= n - i + 1) {

cout << setw(3) << "\*"; // Используем setw для выравнивания звездочек

}

}

cout << endl;

}

}

// Функция для обмена всех слов "Two" и "Null"

string swapWords(string input) {

// Разбиваем строку на слова и сохраняем в вектор

istringstream iss(input);

vector<string> words;

string word;

// Считываем все слова в вектор

while (iss >> word) {

words.push\_back(word);

}

// Переменные для хранения индексов слов "Two" и "Null"

int indexTwo = -1;

int indexNull = -1;

// Ищем индексы слов "Two" и "Null"

for (size\_t i = 0; i < words.size(); ++i) {

if (words[i] == "Two") {

indexTwo = i;

}

if (words[i] == "Null") {

indexNull = i;

}

}

// Если оба слова найдены, меняем их местами

if (indexTwo != -1 && indexNull != -1) {

swap(words[indexTwo], words[indexNull]);

}

// Собираем строку обратно

stringstream result;

for (size\_t i = 0; i < words.size(); ++i) {

result << words[i];

if (i != words.size() - 1) {

result << " "; // Добавляем пробел между словами

}

}

return result.str();

}

// Структура для описания предмета

struct Item {

string name; // Имя предмета

double price; // Цена предмета

double weight; // Вес предмета

double value; // Вычисляемый показатель (например, ценность на единицу веса)

};

// Функция для работы с массивом структур и записи в файл

void saveItemsToFile() {

// Массив с данными о предметах

Item items[5] = {

{"Item 1", 100.0, 1.5, 0.0},

{"Item 2", 200.5, 2.0, 0.0},

{"Item 3", 300.75, 1.8, 0.0},

{"Item 4", 150.25, 1.2, 0.0},

{"Item 5", 400.0, 2.5, 0.0}

};

// Вычисление показателя ценности для каждого предмета

for (auto& item : items) {

item.value = item.price / item.weight;

}

// Открываем файл для записи

ofstream outFile("items.txt");

if (outFile.is\_open()) {

// Записываем имена предметов

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

outFile << setw(15) << left << items[i].name; // Имя предмета

}

outFile << endl;

// Записываем цены

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

outFile << setw(15) << fixed << setprecision(1) << items[i].price; // Цена

}

outFile << endl;

// Записываем веса

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

outFile << setw(15) << fixed << setprecision(1) << items[i].weight; // Вес

}

outFile << endl;

// Записываем ценности

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

outFile << setw(15) << fixed << setprecision(4) << items[i].value; // Ценность

}

outFile << endl;

outFile.close(); // Закрываем файл

cout << "Данные о предметах записаны в файл items.txt." << endl;

} else {

cout << "Ошибка: не удалось открыть файл." << endl;

}

}

// Функция для чтения данных из файла

void readItemsFromFile() {

ifstream inFile("items.txt");

if (!inFile) {

cout << "Ошибка: не удалось открыть файл." << endl;

return;

}

string line;

while (getline(inFile, line)) {

cout << line << endl;

}

inFile.close();

}

// Основная функция

int main() {

char choice;

int task, size;

do {

cout << "Выберите задание:\n1 - Задание 1 (матрица)\n2 - Задание 2 (строка)\n3 - Задание 3 (предметы)\n4 - Чтение из файла\n0 - Выход\n";

cin >> task;

switch (task) {

case 1:

cout << "Введите размер матрицы: ";

cin >> size;

printMatrix(size); // выводим матрицу заданного размера

break;

case 2: {

cout << "Введите исходную строку: ";

cin.ignore(); // Игнорируем символ новой строки после числа

string inputStr;

getline(cin, inputStr);

// Динамически выделяем память для строки

string \*inputPtr = new string(inputStr);

string transformed = swapWords(\*inputPtr);

cout << "Исходная строка: " << \*inputPtr << endl;

cout << "Преобразованная строка: " << transformed << endl;

delete inputPtr; // Освобождаем память

break;

}

case 3:

saveItemsToFile();

break;

case 4:

readItemsFromFile();

break;

case 0:

cout << "Выход из программы." << endl;

break;

default:

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

break;

}

} while (task != 0);

return 0; // завершение программы

}

Результаты работы программы

Задание 1:

Матрица для n = 3:

3 \*

2 \* \*

1 \* \* \*

Манипуляторы:

- `setw(3)` выравнивает элементы на 3 позиции для аккуратного вывода.

Задание 2:

Исходная строка: Zero Two Null

Преобразованная строка: Zero Null Two

Манипуляторы:

- В этом задании манипуляторы не используются напрямую, но используется `getline()` и `stringstream` для работы со строкой.

Задание 3:

Данные о предметах записаны в файл.

Содержимое файла:

Имя: Предмет 1, Цена: 100.0

Имя: Предмет 2, Цена: 200.5

Имя: Предмет 3, Цена: 300.75

Имя: Предмет 4, Цена: 150.25

Имя: Предмет 5, Цена: 400.0

Манипуляторы:

- `setw(15)` выравнивает текст в столбцах.

- `left` выравнивает по левому краю.

- `fixed` и `setprecision()` управляют количеством знаков после запятой.

Задание 4:

Чтение данных из файла успешно выполнено.

Программа выполняет задачи корректно. В задании 1 используется `setw` для выравнивания элементов матрицы. В задании 2 применяется `stringstream` для обработки строки и замены слов, без манипуляторов. Задание 3 записывает данные о предметах в файл с использованием `setw`, `fixed`, `setprecision` для форматирования чисел и выравнивания текста. В задании 4 программа успешно читает данные из файла.