|  |
| --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  федеральное государственное АВТОНОМНОЕ образовательное учреждение высшего образования  «Национальный исследовательский ядерный университет «МИФИ» |
| **Обнинский институт атомной энергетики –**  филиал федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский ядерный университет «МИФИ»  **(ИАТЭ НИЯУ МИФИ)** |

Отделение ядерной физики и технологий

**Лабораторная работа 10**

«Динамические структуры в С++. Текстовые и двоичные файлы»

Выполнил:

студент гр. ЯРМ-С24 Долинов Д.А.

Проверил:

Неведин А.В.

**Обнинск, 2024 г.**

Цель работы : Знакомство с динамическими структурами в С++. Работа с текстовыми и двоичными файлами.

Задание:

В программах необходимо использовать только динамические структуры. Согласно своему варианту выполнить следующие действия:

1. С помощью текстового редактора создать текстовый документ, в который занести исходную информацию. Рекомендуется вначале занести количество записей, а затем последовательно все записи, причем тестовые поля вводить в отдельной строке, а числовые – можно в отдельной строке или в одной через пробел или разделитель табуляцию.

2. Написать первую программу, которая считывает информацию из созданного текстового файла и записывает ее в двоичный файл

3. Написать вторую программу, которая считывает информацию из двоичного файла, реализует поставленную задачу. После повторного чтения двоичного файла результат работы выводится в результирующий текстовый файл. Все текстовые и двоичный файлы должны располагаться или в личной папке или в папке проекта.

Создать двоичный файл с информацией о продаже путевок некоторой туристической фирмой (см. табл. 3.5). Добавить поле «Среднее количество проданных путевок». Упорядочить файл в порядке убывания информации в поле «Продано путевок на сумму».

Программа 1:

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

#include <sstream>

using namespace std;

struct Tour

{

string country;

double price;

int sales[6]; // Static array for sales

Tour(const string &countryName, double tourPrice, const int salesArray[6]) {

country = countryName;

price = tourPrice;

for (int i = 0; i < 6; i++)

{

sales[i] = salesArray[i];

}

}

};

int main()

{

ifstream inputFile("tours.txt");

if (!inputFile.is\_open())

{

cerr << "Error opening input file!" << endl;

return 1;

}

ofstream binaryFile("tours.dat", ios::binary);

if (!binaryFile.is\_open())

{

cerr << "Error opening binary file for writing!" << endl;

return 1;

}

string line;

while (getline(inputFile, line))

{

istringstream iss(line);

string countryName;

double tourPrice;

int salesArray[6];

// Read country name, price, and sales from the line

getline(iss, countryName, ','); // Read country name

iss >> tourPrice; // Read price

iss.ignore(); // Ignore the comma

for (int i = 0; i < 6; i++)

{

iss >> salesArray[i];

if (i < 5)

{

iss.ignore(); // Ignore the comma after each sales number

}

}

// Create a Tour object and write it to the binary file

Tour tour(countryName, tourPrice, salesArray);

size\_t countryLength = tour.country.size();

binaryFile.write(reinterpret\_cast<char \*>(&countryLength), sizeof(countryLength)); // Write length

binaryFile.write(tour.country.c\_str(), countryLength); // Write country name

binaryFile.write(reinterpret\_cast<const char \*>(&tour.price), sizeof(tour.price)); // Write price

binaryFile.write(reinterpret\_cast<const char \*>(tour.sales), sizeof(tour.sales)); // Write sales array

}

inputFile.close();

binaryFile.close();

return 0;

}

Программа 2:

#include <iostream>

#include <fstream>

#include <vector>

#include <string>

#include <algorithm>

#include <numeric>

using namespace std;

struct Tour

{

string country;

double price;

int \*sales; // Pointer to dynamic array

int salesCount;

double average\_sales;

Tour(int count) : salesCount(count)

{

sales = new int[salesCount]; // Dynamic memory allocation for array

}

~Tour()

{

delete[] sales; // Free allocated memory

}

void calculate\_average\_sales()

{

average\_sales = accumulate(sales, sales + salesCount, 0) / static\_cast<double>(salesCount);

}

};

bool compare(const Tour \*a, const Tour \*b)

{

double total\_a = a->price \* accumulate(a->sales, a->sales + a->salesCount, 0);

double total\_b = b->price \* accumulate(b->sales, b->sales + b->salesCount, 0);

return total\_a > total\_b;

}

int main()

{

ifstream binaryFile("tours.dat", ios::binary);

ofstream resultFile("result.txt");

if (!binaryFile.is\_open() || !resultFile.is\_open())

{

cerr << "Error opening files!" << endl;

return 1;

}

vector<Tour \*> tours;

while (true)

{

Tour \*tour = new Tour(6); // Create new Tour object with dynamic array

// Read country name length first

size\_t countryLength;

binaryFile.read(reinterpret\_cast<char \*>(&countryLength), sizeof(countryLength));

if (binaryFile.eof())

{

delete tour; // Free memory if end of file is reached

break;

}

// Read country name

tour->country.resize(countryLength);

binaryFile.read(&tour->country[0], countryLength);

binaryFile.read(reinterpret\_cast<char \*>(&tour->price), sizeof(tour->price));

binaryFile.read(reinterpret\_cast<char \*>(tour->sales), sizeof(int) \* 6); // Read array

tour->calculate\_average\_sales();

tours.push\_back(tour); // Add pointer to vector

}

// Sort by descending order

sort(tours.begin(), tours.end(), compare);

// Write results to text file

for (const auto &t : tours)

{

resultFile << "Country: " << t->country << ", Price: " << t->price

<< ", Average Sales: " << t->average\_sales << "\n";

}

binaryFile.close();

resultFile.close();

// Free allocated memory

for (auto tour : tours)

{

delete tour;

}

return 0;

}

Исходный txt файл с информацией:

France,1200.50,100,200,150,300,250,400

Italy,950.75,80,160,120,240,200,320

Spain,1100.00,90,180,140,270,230,360

Выходной файл с информацией:

Country: France, Price: 1200.5, Average Sales: 233.333

Country: Spain, Price: 1100, Average Sales: 211.667

Country: Italy, Price: 950.75, Average Sales: 186.667

Вывод: Я познакомился с динамическими структурами в С++. Поработал с текстовыми и двоичными файлами.