ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ

УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Кафедра Інформатики

**Звіт**

з лабораторної роботи № 1

Виконав Перевірила

ст.гр.ІТІНФ-20-1 Кириченко I. Ю.

Самченко С.О.

Харків 2021

ЗАВДАННЯ:

21) Описати структуру з іменем ORDER, яка містить наступні поля: ім’я та призвіще платника, розрахунковий рахунок платника; розрахунковий рахунок одержувача; сума переказу. Написати програму, яка виконує наступні дії:

- зчитування даних з рядкового потоку в масив, який складається з структур типу PRICE

- впорядкувати записи за розрахунковими рахунками платників в алфавітному порядку;

- знаходження та виведення на дисплей і у файл інформації про суму, яка знята з розрахункового рахунку платника, введеного з клавіатури; якщо такого розрахункового рахунку немає, вивести відповідне повідомлення.

- Пошук першого ордера призвіще, якого збігається з першими введеними літерами

РЕАЛІЗАЦІЯ:

order.h

#include <iostream>

#include <stdlib.h>

#include <iomanip>

#include <sstream>

#include <string>

#include <fstream>

using namespace std;

namespace NOrder {

struct ORDER {

string surname;

int p = 0;

int r = 0;

int price = 0;

};

// Перевантаження потоків

std::ostream& operator<<(std::ostream& out, const ORDER& ord);

std::istream& operator>>(std::istream& in, ORDER& ord);

// Створюемо власні маніпулятори

std::ostream& manipStr(std::ostream& out);

std::ostream& manipInt(std::ostream& out);

// Функції обробки данних

void input(ORDER\* gr, int n, std::istream& in); //введення з потоку

void output(const ORDER\* gr, int n, std::ostream& out); //введення у потік

void Sort(ORDER\* gr, int n);

const ORDER\* findOrd(const ORDER\* gr, int n); //пошук студента по першим літерам

const ORDER\* findPay(const ORDER\* gr, int n);

};

order.cpp

#include "order.h"

#include <iostream>

#include <iomanip>

#include <sstream>

using namespace std;

namespace NOrder {

// створення власних маніпуляторів

ostream& manipStr(ostream& out) {

out << setw(15) << setfill('.')

<< resetiosflags(ios::right)

<< setiosflags(ios::left);

return out;

}

ostream& manipInt(ostream& out) {

out << setw(4) << setfill('\_')

<< resetiosflags(ios::left)

<< setiosflags(ios::right);

return out;

}

// перевантаження потоків

ostream& operator<<(ostream& out, const ORDER& ord) {

out << manipStr << ord.surname

<< manipInt << ord.p

<< manipInt << ord.r

<< manipInt << ord.price;

return out;

}

istream& operator>>(istream& in, ORDER& ord) {

string str;

getline(in, ord.surname, ';');

getline(in, str, ';');

ord.p = atoi(str.c\_str());

getline(in, str, ';');

ord.r = atoi(str.c\_str());

getline(in, str, ';');

ord.price = atoi(str.c\_str());

return in;

}

// Функції обробки данних

void input(ORDER\* gr, int n, std::istream& in) // Введення з потоку {

int i = 0;

while (i < n && !in.eof()) {

in >> gr[i++];

}

}

void output(const ORDER\* gr, int n, std::ostream& out) //Введення у потік {

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

out << gr[i] << endl;

out << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

}

//Сортування за датою народження

void Sort(ORDER\* gr, int n) {

for (int j = 0; j < n - 1; j++) {

for (int i = 0; i < n - j - 1; i++) {

if (gr[i].surname > gr[i + 1].surname) {

ORDER tmp = gr[i];

gr[i] = gr[i + 1];

gr[i + 1] = tmp;

}

}

}

}

//Пошук студента по першим літерам

const ORDER\* findOrd(const ORDER\* gr, int n) {

string name;

cout << endl << "Input letters: ";

cin >> name;

const ORDER\* result = NULL;

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

if (gr[i].surname.find(name) != std::string::npos) {

result = &gr[i];

break;

}

}

return result;

}

const ORDER\* findPay(const ORDER\* gr, int n) {

ofstream fout;

string path = "myFile.txt";

fout.open(path);

int pay;

cout << "Input current account of the payer: ";

cin >> pay;

const ORDER\* result = NULL;

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

if (gr[i].p == pay) {

result = &gr[i];

fout << "The sum belongs to the user: " << gr[i].surname;

break;

}

}

if (result == NULL) {

cout << "User not found!!!";

fout << "User not found!!!";

}

else {

// cout << \*result << endl;

}

fout.close();

return result;

}

}

Source.cpp

#include "order.h"

using namespace NOrder;

//фукнція підрахунку не пустих рядків у потоці

int countString(istream& in)

{

int count = 0;

std::string line;

while (!in.eof())

{

std::getline(in, line);

if (in && !line.empty()) ++count;

}

return count;

}

int main()

{

std::string inputDataStr = "\nPetrov;200;300;250;\nDmitrov;100;200;150;\nIvanov;300;100;200;\nSidorov;400;220;210;\nOstapov;350;120;325";

std::istringstream inputData(inputDataStr);

int n = countString(inputData);

NOrder::ORDER\* mas = new NOrder::ORDER[n];

inputData.seekg(0);//відновлюемо данні у потоці

inputData.clear(); //скинути прапорці помилок

std::ostringstream resultData;

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

NOrder::input(mas, n, inputData);

NOrder::output(mas, n, resultData);

NOrder::Sort(mas, n);

NOrder::output(mas, n, resultData);

std::ofstream fout("myFile.txt");

cout << resultData.str();

fout << resultData.str();

fout.close();

resultData.str("");//очищення потоку

cout << \*findOrd(mas, n);

cout << endl << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";

cout << endl << endl;

cout << \*findPay(mas, n);

cout << endl << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";

cout << endl << endl;

delete[] mas;

system("pause");

return 0;

}

РЕЗУЛЬТАТИ:

