«Московский авиационный институт»   
(Национальный исследовательский университет)

Лабораторная работа по курсу

«Технологии программирования»

Часть 2

Вариант 24

Выполнил:   
студент группы

 М7О-207С-19

Понкращенков Дмитрий

Проверил:

Барчев Н.Б.

Москва

**Содержание**

1. Задание. ........................................................................................................ 3
2. Псевдокод. ................................................................................................... 4
3. Сведения о программной реализации. ...................................................... 6
4. Инструкция пользователя. ......................................................................... 8
5. Листинг. ..................................................................................................... 10
6. **Задание**
   1. Подготовить программу, формирующую на основе информации, вводимой пользователем с клавиатуры, два внешних файла:
   2. Файл портов: состоит из записей, каждая из которых включает три поля - название порта, страны, моря;
   3. Файл рейсов: состоит из записей, каждая из которых включает три поля - названия судна, порта отправления и прибытия.
   4. Подготовить вторую программу, формирующие сведенья о перевозках, осуществляемых судами, с указанием стран и морей на основе данных из двух внешних файлов, сформированных в результате работы программы, подготовленной в п.1.
   5. В процессе проектирования предусмотреть необходимые по смыслу задания проверки корректности данных, а также адекватное задаче взаимодействие  с пользователем.   
      Во второй программе реализовать возможность вывода итоговой информации в двух режимах: избирательно на экран (по запросу пользователем конкретных данных с клавиатуры) и полностью - в отдельный текстовый файл. Оба режима вывода должны предоставлять сгруппированную по смыслу информацию из обоих входных файлов в виде, удобном для восприятия пользователем.   
      Учесть, что ограничения на размеры файлов отсутствуют, файлы в общем случае могут быть различной длины.   
      В обязательном порядке использовать языковые средства организации программных единиц.   
      Не использовать программные средства, поддерживающие работу с базами данных.
   6. Программы подготовить и использованием средств одной из реализаций языка программирования С++.
7. **Псевдокод.**

Объявление функции поиска файла.

Если файл открыт, то конец функции.

Объявление функции поиска портов

Пока не достигнут конец файла Портов

Сверять искомую строку со строкой в файле.

Объявление функции поиска рейсов

Пока не достигнут конец файла Рейсов

Сверять искомую строку со строкой в файле.

Объявление функции поиска Портов в файле Рейсов.

Пока не достигнут конец файла Рейсов

Сверять искомую строку со строкой в файле.

* Если найден файл портов и файл рейсов

Создание главного меню программы:

* Начать поиск Перевозок, осуществляемых судами (1)
* Редактировать файл Рейсов (2)
* Выйти из программы. (esc)
* Если выбран 1-ый вариант:

Считать строку для поиска.

Поиск пробела в веденой строке и замена на тире.

Если введённая строка найдена среди названий рейсов в файле рейсов

Вывести информацию о рейсе и портах отправления/прибытия.

Если введённая строка не найдена среди названий рейсов в файле рейсов

Осведомить пользователя и запросить новый ввод строки.

Организовать выход в главное меню программы или продолжить поиск рейсов в файле рейсов

* Если выбран 2-ой вариант:

Пока не достигнут конец файла рейсов

Вывести: название судна, Информацию порта отправления;

Информацию порта прибытия;

Пока не достигнут конец файла портов

Если название порта из файла портов не найдено среди портов отправления/прибытия из файла рейсов

Вывести: Название Порта, Страны, моря;

Произвести анимацию сохранения файла

Выйти в главное меню.

* Если выбран 3-ий вариант:

Произвести анимацию завершения программы

Выйти из программы.

* Иначе сообщение об ошибке
* Если файлы портов и рейсов не были найдены

Подсказать пользователю где расположить файлы.

1. **Сведения о программной реализации:**

Версия системы программирования:   
Microsoft Visual Studio Community 2019 Version 16.3.8

Требуемые для работы ресурсы вычислительной системы:

|  |  |
| --- | --- |
| Поддерживаемые операционные системы: | * Windows 7 и выше |
| Процессор | * Процессор с тактовой частотой не ниже 1,8 ГГц. |
| Оперативная память | * 2 ГБ ОЗУ, рекомендуется 4 ГБ ОЗУ |

**Описание входных и выходных данных:**

Входными данными будут два текстовых файла:

Файл Портов:

* Название Порта;
* Название Страны;
* Название Моря;

Файл Рейсов:

* Название судна;
* Порт отправления;
* Порт прибытия;

Выходными данными будет текстовый файл под названием Input.txt, содержащий всю информацию, описанную выше.

Файл будет автоматически создан в одной папке с программой.

Входные файлы должны лежать по тому же пути, что и программа.

|  |  |
| --- | --- |
| Диагностические сообщения | Причина возникновения |
| «Введена неправильная клавиша» | При выборе варианта была нажата неверная клавиша, т.е. была нажата любая клавиша, кроме «1», «2», «Esc» |
| «Данный маршрут не найден» | В файле рейсов не нашлось схожих названий Рейсов. |
| «Файлы Ports.txt и Rase.txt не обнаружены по пути:» | В папке с программой не найдены файлы Port.txt и Rase.txt |
| «Файл Ports.txt не обнаружен по пути:» | В папке с программой не найдены файл Port.txt |
| «Файл Rase.txt не обнаружен по пути:» | В папке с программой не найдены файл Rase.txt |

**Программные единицы:**

Функция: **isfile** (*string filepatch*).

Назначение: поиск файла.

Тип: bool;

Входные параметры: filepatch путь к файлу.

Тип: string;

Возвращаемое значение checks;

Тип: bool;

--------------------------------------------------------------

Функция **FindPort** (*char FindLocalPort , string Path*)

Назначение: Поиск порта в файле портов.

Входные параметры: FindLocalPort, Тип: char;

Path, Тип string;

Возвращаемое значение Local;

Тип: Sport;

--------------------------------------------------------------

Функция **FindRase** (*char FindLocalRase , string Path*)

Назначение: Поиск порта в файле портов.

Входные параметры: FindLocalRase, Тип: char;

Path, Тип string;

Возвращаемое значение Local;

Тип: Sport;

--------------------------------------------------------------

Функция **FindPortInRase** (*char FindLocalPort , string Path*)

Назначение: Поиск порта в файле портов.

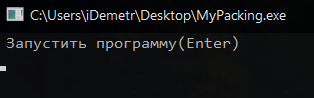
Входные параметры: FindLocalPort, Тип: char;

Path, Тип string;

Возвращаемое значение Local;

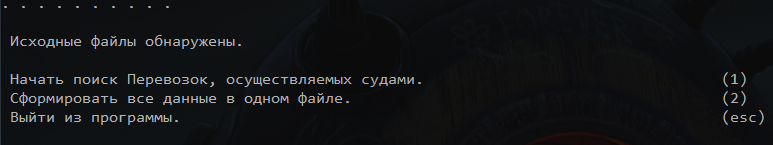
Тип: Sport;

1. **Инструкция пользователя:**

После запуска программы на экране монитора появляется сообщение с предложением загрузить программу:

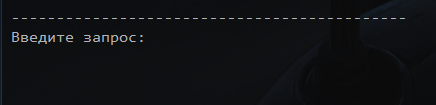
Для продолжения программы необходимо нажать клавишу «Enter».

После проигрывания анимации загрузки программы на экране монитора появляется меню с выбором действий:

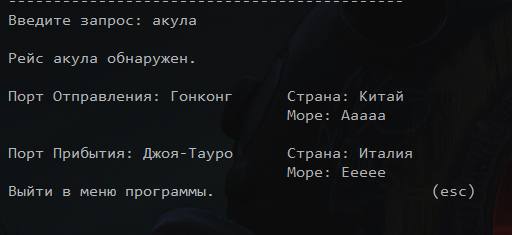


Выбор пунктов осуществляется с помощью ввода нужного номера пункта с клавиатуры.

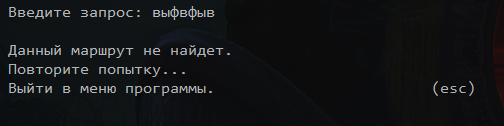
После выбора пользователем, например, первого пункта меню, на экране появится следующее сообщение:



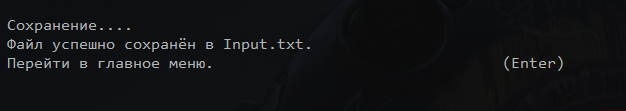
После ввода пользователем названия нужного ему издания, и успешного поиска информация выведется на экран:

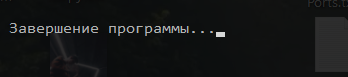


Для продолжения поиска достаточно нажать любую клавишу.

При неудачном поиске программа выведет на экран:   
 После перехода в главное меню программы, будет произведена очистка экрана программы.

После выбора пользователем варианта №2 программа выведет на экран уведомление об успешном сохранении файла:



При выходе из программы будет проигранна анимация завершения программы и последующий выход из нее:

Успешной эксплуатации!

1. **Листинг.**

#include <iostream>

#include <fstream>

#include <string>

#include <conio.h>

#include <Windows.h>

#include <string.h>

#include <stdio.h>

#include <cstdio>

using namespace std;

bool isfile(std::string filepath) //Функция проверки наличия файла.

{

bool a = false;

std::ifstream b(filepath.c\_str());

if (b.is\_open())

a = true;

b.close();

return a;

}

struct SPort //Структура Портов.

{

char NamePort[15]; //Название Порта.

char Contry[15]; //Название Страны.

char Sea[15]; //Название моря.

};

struct SRase //Структура Рейсов.

{

char NameSudna[20]; //Название Судна.

char PortOtpr[15]; //Порт отправления.

char PortPribit[15]; //Порт прибытия.

};

SPort FindPort(char FindLocalPort[15], string Path) //Процедура поиска в файле портов.

{

SPort Local;

ifstream f1(Path);

//f1.open(Path, ios::in);

while (!f1.eof())

{

f1 >> Local.NamePort;

f1 >> Local.Contry;

f1 >> Local.Sea;

if (\_stricmp(FindLocalPort, Local.NamePort) == 0)

break;

}

f1.close();

return Local;

}

SRase FindRase(char FindLocalRase[15], string Path) //Процедура поиска в файле рейсов.

{

SRase Local;

ifstream f2(Path);

while (!f2.eof())

{

f2 >> Local.NameSudna;

f2 >> Local.PortOtpr;

f2 >> Local.PortPribit;

if (\_stricmp(FindLocalRase, Local.NameSudna) == 0)

break;

}

f2.close();

return Local;

}

SRase FindPortInRase(char FindLocalPort[15], string Path) //Процедура поиска портов в файле рейсов.

{

SRase Local;

ifstream f2(Path);

while (!f2.eof())

{

f2 >> Local.NameSudna;

f2 >> Local.PortOtpr;

f2 >> Local.PortPribit;

if ((!\_stricmp(FindLocalPort, Local.PortOtpr) == 0) && (!\_stricmp(FindLocalPort, Local.PortPribit) == 0))

continue;

else

break;

}

f2.close();

return Local;

}

int main()

{

setlocale(LC\_ALL, "Russian");

SetConsoleCP(1251); //Русский язык в консоли.

SetConsoleOutputCP(1251); //Русский язык в файле.

cin.sync();

char Rase[15]; //Поиск типа char.

char ch = 0; //Буфер ввода.

int i = 0; //Счетчик.

int Flag1 = 0; //Цикл всей программы.

int Flag2 = 0; //Цикл подпрограммы.

SPort MySPort; //Переменная Структуры Портов.

SRase MySRase; //Переменная Структуры Рейсов.

string Buffer; //Поиск типа string.

string PortName = "Ports.txt"; //Путь к файлу Портов.

string RaseName = "Rase.txt"; //Путь к файлу Рейсов.

string PathName = "C:Users/User/Desktop/"; //Путь к рабочему столу.

cout << " Запустить программу(Enter)\n";

\_getch();

cout << " Запуск программы:\n";

do

{ //Анимация загрузки.

i++;

cout << ". ";

Sleep(200);

} while (i != 10);

if (isfile(PortName) && isfile(RaseName)) //Проверка наличия исходных файлов.

{

cout << "\n\n Исходные файлы обнаружены.\n ";

while (Flag1 < 1) //Переход в меню.

{

cout << "\n Начать поиск Перевозок, осуществляемых судами. (1)";

cout << "\n Сформировать все данные в одном файле. (2)";

cout << "\n Выйти из программы. (esc) \n";

ch = \_getch();

if (ch == 49) //Выбран 1 вариант.

{

system("cls");

Flag2 = 0;

while (Flag2 < 1)

{

cout << "\n --------------------------------------------";

cout << "\n Введите запрос: ";

cin >> Rase; //Ввод запроса.

for (size\_t r = 0; r < strlen(Rase) + 1; r++) //Замена пробелов в воде на Тире(-).

{

if (Rase[r] == ' ')

Rase[r] = '-';

};

MySRase = FindRase(Rase, RaseName); //Сканирование файла рейсов.

if (\_stricmp(Rase, MySRase.NameSudna) == 0) //Если рейс найден.

{

cout << "\n Рейс " << Rase << " обнаружен.\n";

cout << "\n Порт Отправления: " << MySRase.PortOtpr; //Вывод порта отправления.

MySPort = FindPort(MySRase.PortOtpr, PortName);

if (\_stricmp(MySRase.PortOtpr, MySPort.NamePort) == 0) //Вывод информации о порте.

{

cout << " Страна: " << MySPort.Contry;

cout << "\n Море: " << MySPort.Sea;

}

else

cout << "\n Данные о " << MySRase.PortOtpr << " отсутсвуют..."; //Если информация отсутствует.

cout << "\n\n Порт Прибытия: " << MySRase.PortPribit;

MySPort = FindPort(MySRase.PortPribit, PortName);

if (\_stricmp(MySRase.PortPribit, MySPort.NamePort) == 0) //Вывод информации о порте.

{

cout << " Страна: " << MySPort.Contry;

cout << "\n Море: " << MySPort.Sea;

}

else

cout << "\n Данные о " << MySRase.PortPribit << " отсутсвуют..."; //Если информация отсутствует.

cout << "\n Выйти в меню программы. (esc)\n";

if (\_getch() == 27)

{

Flag1 = 0;

Flag2 = 1;

system("cls");

}

else

Flag2 = 0;

}

else

{

cout << "\n Данный маршрут не найден.\n Повторите попытку...\n Выйти в меню программы. (esc)\n\n";

ch = \_getch();

if (ch == 27)

{

Flag2 = 1;

Flag1 = 0;

system("cls");

}

else

Flag2 = 0;

}

}

}

else

if (ch == 50)

{

system("cls");

ofstream f3("Input.txt"); //Файл выхода;

ifstream f1(RaseName); //файл Рейсов;

ifstream f2(PortName); //Файл Портов;

f3 << " Рейсы: \tПорт Отправления/Прибытия: \tСтрана: \tМоре: \n";

while (!f1.eof())

{

f1 >> MySRase.NameSudna;

f1 >> MySRase.PortOtpr;

f1 >> MySRase.PortPribit;

if (strlen(MySRase.NameSudna) >= strlen(" ")) //Корректировка вывода по столбцу Рейсов.

f3 << "\n" << MySRase.NameSudna << "\t\t " << MySRase.PortOtpr;

else

f3 << "\n" << MySRase.NameSudna << "\t\t\t " << MySRase.PortOtpr;

MySPort = FindPort(MySRase.PortOtpr, PortName);

if (\_stricmp(MySRase.PortOtpr, MySPort.NamePort) == 0) //Вывод информации о порте.

{

if (strlen(MySRase.PortOtpr) >= strlen(" ")) //Корректировка вывода по столбцу Портов.

{

if (strlen(MySPort.Contry) >= strlen(" "))

f3 << "\t\t" << MySPort.Contry << "\t" << MySPort.Sea << "\n";

else

f3 << "\t\t" << MySPort.Contry << "\t\t" << MySPort.Sea << "\n";

}

else

{

if (strlen(MySPort.Contry) >= strlen(" "))

f3 << "\t\t\t" << MySPort.Contry << "\t" << MySPort.Sea << "\n";

else

f3 << "\t\t\t" << MySPort.Contry << "\t\t" << MySPort.Sea << "\n";

}

}

else

f3 << "\t\t\t---- \t\t---- \n";

if (strlen(MySRase.NameSudna) >= strlen(" ")) //Корректировка вывода по столбцу Рейсов.

f3 << "\t\t\t " << MySRase.PortPribit;

else

f3 << "\t\t\t " << MySRase.PortPribit;

MySPort = FindPort(MySRase.PortPribit, PortName);

if (\_stricmp(MySRase.PortPribit, MySPort.NamePort) == 0) //Вывод информации о порте.

{

if (strlen(MySRase.PortPribit) >= strlen(" ")) //Корректировка вывода по столбцу Портов.

{

if (strlen(MySPort.Contry) >= strlen(" "))

f3 << "\t\t" << MySPort.Contry << "\t" << MySPort.Sea << "\n";

else

f3 << "\t\t" << MySPort.Contry << "\t\t" << MySPort.Sea << "\n";

}

else

{

if (strlen(MySPort.Contry) >= strlen(" "))

f3 << "\t\t\t" << MySPort.Contry << "\t" << MySPort.Sea << "\n";

else

f3 << "\t\t\t" << MySPort.Contry << "\t\t" << MySPort.Sea << "\n";

}

}

else

f3 << "\t\t\t---- \t\t---- \n";

}

while (!f2.eof())

{

f2 >> MySPort.NamePort;

f2 >> MySPort.Contry;

f2 >> MySPort.Sea;

MySRase = FindPortInRase(MySPort.NamePort, RaseName);

if (!\_stricmp(MySPort.NamePort, MySRase.PortOtpr)==0 && !\_stricmp(MySPort.NamePort, MySRase.PortPribit)==0)

{

f3 << "----\t\t\t" << MySPort.NamePort;

if (strlen(MySPort.NamePort) >= strlen(" ")) //Корректировка вывода по столбцу Портов.

{

if (strlen(MySPort.Contry) >= strlen(" "))

f3 << "\t\t" << MySPort.Contry << "\t" << MySPort.Sea << "\n";

else

f3 << "\t\t" << MySPort.Contry << "\t\t" << MySPort.Sea << "\n";

}

else

{

if (strlen(MySPort.Contry) >= strlen(" "))

f3 << "\t\t\t" << MySPort.Contry << "\t" << MySPort.Sea << "\n";

else

f3 << "\t\t\t" << MySPort.Contry << "\t\t" << MySPort.Sea << "\n";

}

}

}

f3.close();

f2.close();

f1.close();

cout << "\n Сохранение";

i = 0;

do

{ //Анимация загрузки

i++;

cout << ".";

Sleep(300);

} while (i != 4);

cout << "\n Файл успешно сохранён в Input.txt. \n Перейти в главное меню. (Enter)\n";

\_getch();

system("cls");

}

else

if (ch == 27) //Выход из программы.

{

cout << "\n Завершение программы";

i = 0;

do

{ //Анимация загрузки.

i++;

cout << ".";

Sleep(500);

} while (i != 4);

return 1;

}

else

cout << "\n\n Введена неправильная клавиша...\n";

}

}

else // Если файлы не обнаружены.

{

if (isfile(RaseName) == false && isfile(PortName) == false)

cout << "\n Файлы Ports.txt и Rase.txt не обнаружены по пути: " << PathName;

if (isfile(RaseName) && isfile(PortName) == false)

cout << "\n Файл Ports.txt не обнаружен по пути: " << PathName;

if (isfile(RaseName) == false && isfile(PortName))

cout << "\n Файл Rase.txt не обнаружен по пути: " << PathName;

}

\_getch();

cout << "\n Завершение программы";

i = 0;

do

{ //Анимация загрузки.

i++;

cout << ".";

Sleep(500);

} while (i != 4);

return 0;

}