ГУАП

КАФЕДРА № 14

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ассистент |  |  |  | А.Ю. Петров |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №2 |
|  |
| по курсу: ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 1141 |  |  |  | Г.С. Коржов |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2023

1) Постановка задачи

***Задание 1. Стандартные потоки:***

Определить класс с именем ROUTE, содержащий следующие поля:

● название начального пункта маршрута;

● название конечного пункта маршрута;

● номер маршрута.

Определить методы доступа к этим полям и перегруженные операции

извлечения и вставки для объектов типа ROUTE.

Написать программу, выполняющую следующие действия:

● ввод с клавиатуры данных в массив, состоящий из восьми объектов

типа ROUTE; записи должны быть упорядочены по номерам

маршрутов;

● вывод на экран информации о маршрутах, которые начинаются или

кончаются в пункте, название которого введено с клавиатуры;

● если таких маршрутов нет, выдать на дисплей соответствующее

сообщение

***Задание 2. Файловые и строковые потоки:***

С использованием файловых и строковых потоков написать программу,

которая считывает текст из файла и выводит на экран только строки, не

содержащие двузначные числа.

2) Формализация

***Задание 1:***

Сначала необходимо создать класс Route, содержащий поля: startingPoint, finalDestination, number. Реализовать методы, включая конструкторы, методы установки и получения данных. И перегрузить операторы << и >> для вывода и ввода объектов данного класса.

В завершении работы необходимо с помощью функций создать удобный интерфейс для взаимодействия с классами, который реализует возможность сортировки, ввода, вывода, удаления и поиска элементов.

На протяжении всей работы нужно создавать отчёты в системе контроля версий Git.

***Задание 2:***

Необходимо создать программу, которая считывает текст из файла , а затем выводит только выводит на экран только строки, не содержащие двузначные числа.

Для этого нужно:

* Если файл не может быть открыт, программа должна вывести сообщение об ошибке.
* Считывать из файла строку, создавать для нее строковый поток, из него вычленять слова (последовательности символов, разделенные пробелом), каждое слово переводить в тип integer в том случае, если оно переводится, проверять его на двузначность, если оно двузначное, строку не выводить на экран.
* Закрыть файл после завершения чтения.

3) Таблица описания классов

***Задание 1:***

4) Листинг

***Задание 1:***

**Файл main.cpp:**

#include <iostream>

#include "Route.h"

int main(void)

{

int n;

Route\* routes;

while (true)

{

cout << "0. Create routes." << endl;

cout << "1. Print information about the route. " << endl;

cout << "2. Print routes." << endl;

cout << "3. Edit route. " << endl;

cout << "4. Change route to new. " << endl;

cout << "5. Delete route. " << endl;

cout << "6. Exit." << endl;

int choice;

cin >> choice;

switch (choice)

{

case 0:

{

cout << "How many routes? ";

cin >> n;

routes = new Route[n];

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

{

cin >> routes[i];

}

//sorting

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

{

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

{

if (routes[j].GetNumber() > routes[j + 1].GetNumber())

{

swap(routes[j], routes[j + 1]);

}

}

}

break;

}

case 1:

{

std::string searchPoint;

std::cout << "Enter the point to search for: ";

std::cin >> searchPoint;

bool found = false;

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

if (routes[i].GetStartingPoint() == searchPoint ||

routes[i].GetStartingPoint() == searchPoint)

{

std::cout << routes[i] << std::endl;

found = true;

}

}

if (!found)

std::cout << "No routes found for the specified point." << std::endl;

}

break;

case 2:

{

cout << "Printing routes..." << endl;

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

{

cout << routes[i];

}

}

break;

case 3:

{

int routeToEdit;

cout << "Route to edit? ";

cin >> routeToEdit;

cin >> routes[routeToEdit];

}

break;

case 4:

{

int routeToChange;

cout << "Route to change? ";

cin >> routeToChange;

cin >> routes[routeToChange];

}

break;

case 5:

{

int routeToDelete;

cout << "Route to delete? ";

cin >> routeToDelete;

cin >> routes[routeToDelete];

}

break;

case 6:

cout << "Exiting..." << endl;

return 0;

default:

return 0;

}

}

return 0;

}

**Файл Route.h:**

#ifndef ROUTE\_H

#define ROUTE\_H

#include <string>

#include <fstream>

#include <iostream>

using std::cin;

using std::cout;

using std::endl;

using std::string;

using std::istream;

using std::ostream;

using std::swap;

class Route

{

private:

string startingPoint;

string finalDestination;

int number;

public:

Route();

~Route();

string GetStartingPoint();

string GetFinalDestination();

int GetNumber();

void SetStartingPoint(string startingPoint);

void SetFinalDestination(string finalDestination);

void SetNumber(int number);

friend istream& operator >>(istream& is, Route& route);

friend ostream& operator <<(ostream& os, const Route& route);

};

#endif

**Файл Route.cpp:**

#include "Route.h"

Route::Route()

{

};

Route::~Route()

{

};

string Route::GetStartingPoint()

{

return startingPoint;

};

string Route::GetFinalDestination()

{

return finalDestination;

};

int Route::GetNumber()

{

return number;

};

void Route::SetStartingPoint(string startingPoint)

{

this->startingPoint = startingPoint;

};

void Route::SetFinalDestination(string finalDestination)

{

this->finalDestination = finalDestination;

};

void Route::SetNumber(int number)

{

this->number = number;

};

istream& operator >>(istream& is, Route& route)

{

string startingPoint;

string finalDestination;

int number;

cout << "Starting point: ";

is >> startingPoint;

route.SetStartingPoint(startingPoint);

cout << "Final destination: ";

is >> finalDestination;

route.SetFinalDestination(finalDestination);

cout << "Number: ";

is >> number;

route.SetNumber(number);

return is;

};

ostream& operator <<(ostream& os, const Route& route)

{

os << "Starting point: " << route.startingPoint << endl;

os << "Final destination: " << route.finalDestination << endl;

os << "Number: " << route.number << endl;

return os;

};

}

***Задание 2:***

**Файл main.cpp:**

#include <iostream>

#include <fstream>

#include <sstream>

#include <string>

int main() {

std::ifstream inputFile("input.txt");

if (!inputFile.is\_open()) {

std::cerr << "Unable to open the input file." << std::endl;

return 1;

}

// Строка для считывания каждой строки из файла

std::string line;

// Считываем строки из файла

while (std::getline(inputFile, line)) {

bool containsTwoDigitNumber = false;

// Создаем строковый поток для анализа содержимого строки

std::istringstream iss(line);

// Проверяем каждое слово в строке на наличие двузначных чисел

std::string word;

while (iss >> word) {

try {

// Пробуем преобразовать слово в число

int number = std::stoi(word);

if (number >= 10 && number <= 99) {

containsTwoDigitNumber = true;

break;

}

} catch (const std::invalid\_argument& e) {

// Игнорируем слова, которые нельзя преобразовать в число

}

}

if (!containsTwoDigitNumber) {

std::cout << line << std::endl;

}

}

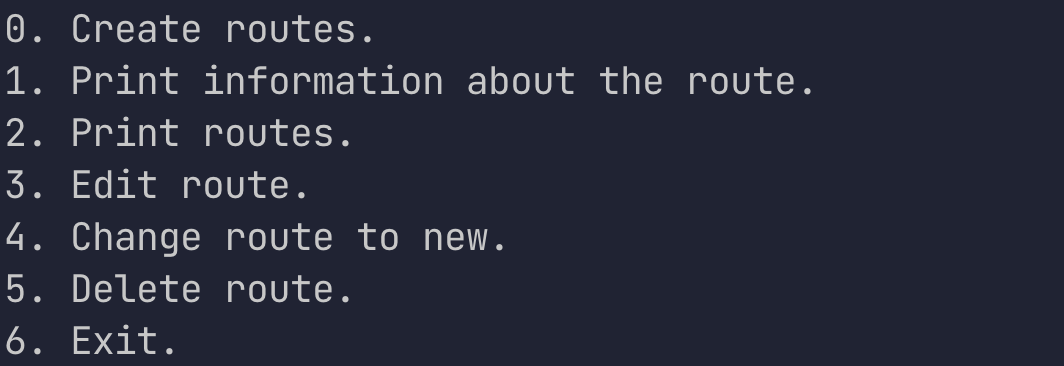
inputFile.close();

return 0;

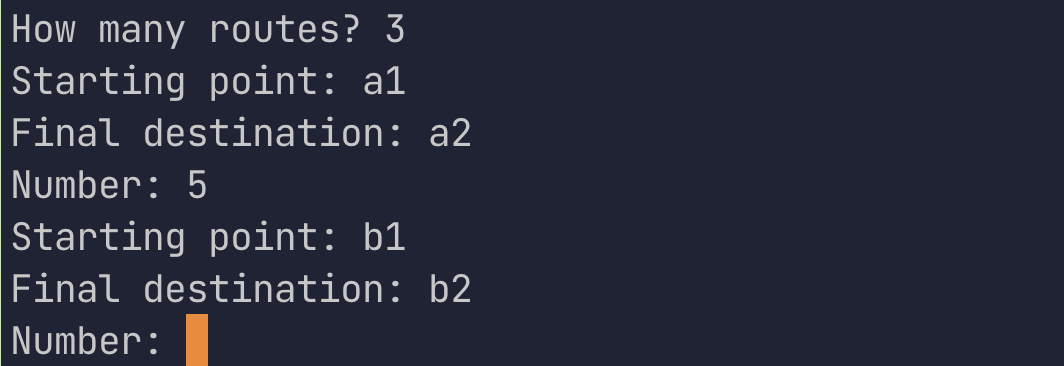
}

5) Результаты работы программы

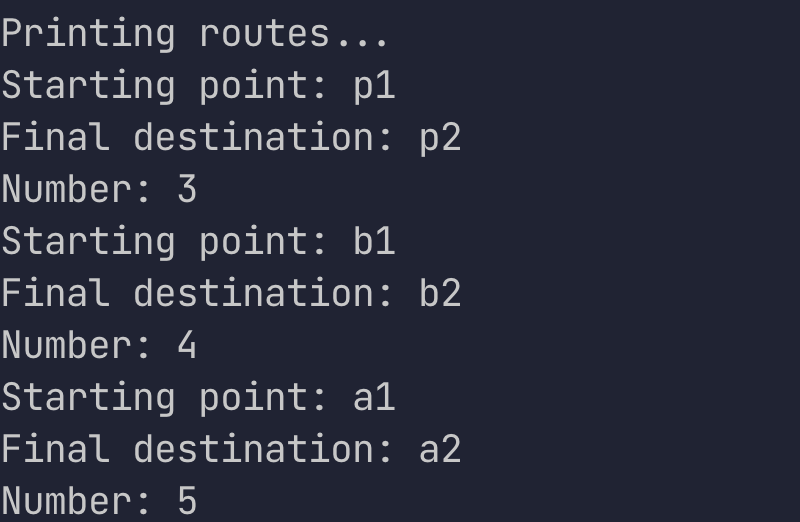
***Задание 1:***



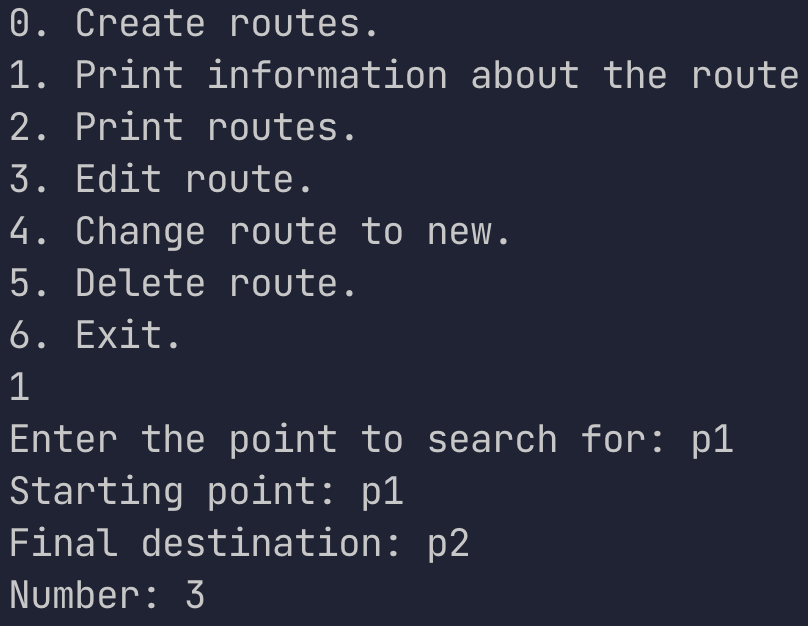
*Рисунок 2. Результат работы программы 1*



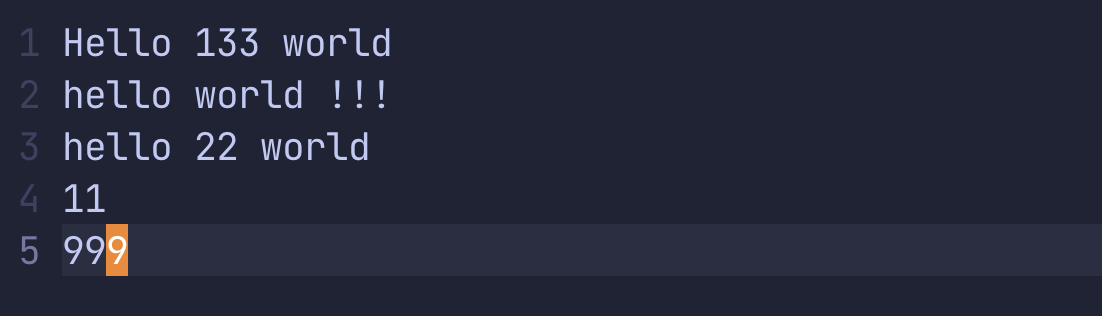
*Рисунок 3. Результат работы программы 1*



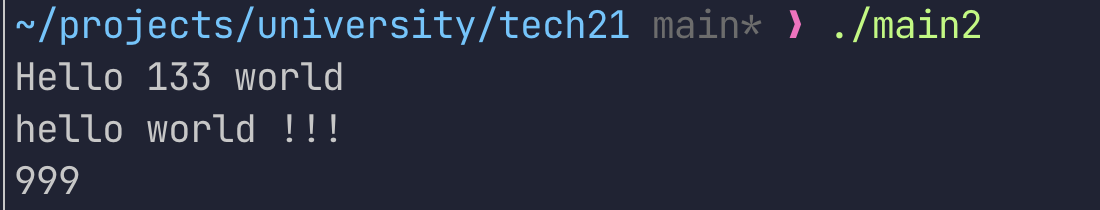
*Рисунок 4. Результат работы программы 1*

**

*Рисунок 5. Результат работы программы 1*

***Задание 2:*** **

*Рисунок 6. Входной файл input.txt*

**

*Рисунок 7. Результат работы программы 2*

6) Вывод

***Задание 1. Стандартные потоки:***

Мною был определён класс с именем Route, содержащий поля. Определены методы доступа к этим полям и перегружены операции извлечения и вставки для объектов типа Route. Записи сортируются по номерам. Реализован вывод на экран информации о маршруте, если одна из точек маршрута совпадает с запрошенной. Мною были изучены и освоены стандартные потоки.

Репозиторий с проектом - https://github.com/rellarda/tech2.git

***Задание 2. Файловые и строковые потоки:***

Мною с использованием файловых и строковых потоков была написана программа, которая считывает текст из файла и выводит на экран только те строки, в которых нет двухзначных чисел. Мною были изучены и освоены файловые и строковые потоки.

Репозиторий с проектом - https://github.com/rellarda/tech2.git