Courier.cpp

The most recent version

Generated by Doxygen 1.9.6

Chapter 1

Introduction

This program create a file with routes between company centre to all other cities.

1.1 Manual

1) Download all files from my repository called Courier_cpp on the GitHub;
2) Launch command prompt;
3) Use command 'cd' and go to files "Courier_cpp", "Courier", "x64", "Debug";
4) Then type in arguments from point at the number 5.;
5) Courier.exe -i input.txt -o output.txt -c centre; (Remember that instead input.txt, output.txt and centre type in proper params)
For examples: Courier.exe -i data_file.txt -o answer_file.txt -c Poznan

Author

Karol Pitera

Date

23.02.2023

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 4

Class Documentation

4.1 edge Struct Reference

The scructure includes a information about a given neighbouring city.

`#include <struct.h>`

Public Attributes

- double **range**
- std::string **end**

4.1.1 Detailed Description

The scructure includes a information about a given neighbouring city.

Parameters

range	The distance between the cities.
end	Neighbouring city.

The documentation for this struct was generated from the following file:

- C:/Users/Ryzen/Desktop/Projekty/Courier_cpp/Courier/struct.h

4.2 vertex Struct Reference

The structure include a information about the given city.

`#include <struct.h>`

Public Attributes

- double **distance** = std::numeric_limits<double>::max()
- std::string **previous**
- std::vector< edge > **neighbors**
- bool **visited** = false

4.2.1 Detailed Description

The structure include a information about the given city.

Parameters

previous	The earlier city that was determined by the algorithm
distance	Dictance from the given city to the center.
neighbors	Vector of nieghbouring cities stcrutures.
visited	Bool value, that include the information about visiting the city.

The documentation for this struct was generated from the following file:

- C:/Users/Ryzen/Desktop/Projekty/Courier_cpp/Courier/struct.h

Chapter 5

File Documentation

5.1 C:/Users/Ryzen/Desktop/Projekty/Courier_cpp/Courier/features.cpp File Reference

complete features file

```
#include <iostream>
#include <cmath>
#include <vector>
#include <unordered_map>
#include <fstream>
#include <deque>
#include "features.h"
#include "struct.h"
```

Functions

- void service_cmd (std::string &input, std::string &output, std::string &centre, int argc, char *argv[])

  Feature assign a proper params to variables input, output and centre.
- void read_data (std::unordered_map< std::string, vertex > &graph, std::string input)

  Feature reads a information from a data file and save the data to the map.
- void Dijkstra (std::unordered_map< std::string, vertex > &graph, std::string &centre, std::vector< std::string > &unavailable)

  The feature searches the shortest routes from the central city to all different cities.
- void typing_result (std::unordered_map< std::string, vertex > graph, std::string center, std::vector< std::string > &unavailable, std::string output)

  The feature sorts the cities and previous citis, and saves the program result to the new create file.

5.1.1 Detailed Description

complete features file

5.1.2 Function Documentation

5.1.2.1 Dijkstra()

```
void Dijkstra (
            std::unordered_map< std::string, vertex > & graph,
            std::string & centre,
            std::vector< std::string > & unavailable )
```

The feature searches the shortest routes from the central city to all different cities.

Parameters

graph	unordered map, which is indexed with city names.
unavailable	The cities, which haven't got any route connecting to the center.

5.1.2.2 read_data()

```
void read_data (
            std::unordered_map< std::string, vertex > & graph,
            std::string input )
```

Feature reads a information from a data file and save the data to the map.

The map includes structures with information about cities.

Parameters

graph	unordered map, which is indexed with city names.
input	the variable includes a file name with input data.

5.1.2.3 service_cmd()

```
void service_cmd (
            std::string & input,
            std::string & output,
            std::string & centre,
            int argc,
            char * argv[] )
```

Feature assign a proper params to variables input, output and centre.

Parameters

input	the variable includes a input file name with data
output	the variable includes output file name with a program result
centre	The varaible includes name of central city

5.1.2.4 typing_result()

```
void typing_result (
            std::unordered_map< std::string, vertex > graph,
            std::string center,
            std::vector< std::string > & unavailable,
            std::string output )
```

The feature sorts the cities and previous citis, and saves the program result to the new create file.

When the function finished sort, then all cities visited in one route are typed in the right order with finish distance.

Parameters

graph	unordered map, which is indexed with city names.
centre	The varaible includes name of central city
unavailable	The cities, which haven't got any route connecting to the center.
output	the variable includes output file name with a program result

5.2 C:/Users/Ryzen/Desktop/Projekty/Courier_cpp/Courier/features.h File Reference

Header file.

```
#include <iostream>
#include <cmath>
#include <vector>
#include <unordered_map>
#include <fstream>
#include "struct.h"
```

Functions

- void service_cmd (std::string &input, std::string &output, std::string &centre, int argc, char ∗argv[ ])

  Feature assign a proper params to variables input, output and centre.
- void read_data (std::unordered_map< std::string, vertex > &graph, std::string input)

  Feature reads a information from a data file and save the data to the map.

- void Dijkstra (std::unordered_map< std::string, vertex > &graph, std::string &centre, std::vector< std::string > &unavailable)

  The feature searches the shortest routes from the central city to all different cities.
- void typing_result (std::unordered_map< std::string, vertex > graph, std::string center, std::vector< std::string > &unavailable, std::string output)

  The feature sorts the cities and previous citis, and saves the program result to the new create file.

5.2.1 Detailed Description

Header file.

5.2.2 Function Documentation

5.2.2.1 Dijkstra()

```
void Dijkstra (
            std::unordered_map< std::string, vertex > & graph,
            std::string & centre,
            std::vector< std::string > & unavailable )
```

The feature searches the shortest routes from the central city to all different cities.

Parameters

graph	unordered map, which is indexed with city names.
unavailable	The cities, which haven't got any route connecting to the center.

5.2.2.2 read_data()

```
void read_data (
            std::unordered_map< std::string, vertex > & graph,
            std::string input )
```

Feature reads a information from a data file and save the data to the map.

The map includes structures with information about cities.

Parameters

graph	unordered map, which is indexed with city names.
input	the variable includes a file name with input data.

5.2.2.3 service_cmd()

```
void service_cmd (
              std::string & input,
              std::string & output,
              std::string & centre,
              int argc,
              char * argv[] )
```

Feature assign a proper params to variables input, output and centre.

Parameters

input	the variable includes a input file name with data
output	the variable includes output file name with a program result
centre	The varaible includes name of central city

5.2.2.4 typing_result()

```
void typing_result (
              std::unordered_map< std::string, vertex > graph,
              std::string center,
              std::vector< std::string > & unavailable,
              std::string output )
```

The feature sorts the cities and previous citis, and saves the program result to the new create file.

When the function finished sort, then all cities visited in one route are typed in the right order with finish distance.

Parameters

graph	unordered map, which is indexed with city names.
centre	The varaible includes name of central city
unavailable	The cities, which haven't got any route connecting to the center.
output	the variable includes output file name with a program result

5.3 features.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 #include <iostream>
00004 #include <cmath>
00005 #include <vector>
00006 #include <unordered_map>
00007 #include <fstream>
00008
00009 #include "struct.h"
00013 void service_cmd(std::string& input, std::string& output, std::string & centre, int argc, char*
      argv[]);
00014
```

```
00015 void read_data(std::unordered_map <std::string, vertex> & graph, std::string input);
00016
00017 void Dijkstra(std::unordered_map <std::string, vertex> & graph, std::string & centre, std::vector
      <std::string> & unavailable);
00018
00019 void typing_result(std::unordered_map <std::string, vertex> graph, std::string center, std::vector
      <std::string>& unavailable,
00020                  std::string output);
```

5.4 C:/Users/Ryzen/Desktop/Projekty/Courier_cpp/Courier/main.cpp File Reference

File with main feature.

```
#include <iostream>
#include <cmath>
#include <vector>
#include <unordered_map>
#include <fstream>
#include <deque>
#include "features.h"
#include "struct.h"
```

Functions

- int **main** (int argc, char ∗argv[ ])

5.4.1 Detailed Description

File with main feature.

Parameters

argc	number of typed arguments.
argv	params contents.

5.5 C:/Users/Ryzen/Desktop/Projekty/Courier_cpp/Courier/struct.h File Reference

Structures file.

```
#include <iostream>
#include <cmath>
#include <vector>
#include <unordered_map>
```

Classes

- struct edge

  The scructure includes a information about a given neighbouring city.
- struct vertex

  The structure include a information about the given city.

5.5.1 Detailed Description

Structures file.

5.6 struct.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 #include <iostream>
00004 #include <cmath>
00005 #include <vector>
00006 #include <unordered_map>
00007
00019 struct edge {
00020     double range;
00021     std::string end;
00022 };
00023
00024
00033 struct vertex {
00034
00035     double distance = std::numeric_limits<double>::max();
00036     std::string previous; ;
00037     std::vector <edge> neighbors;
00038     bool visited = false;
00039 };
```

Index