problem13

Generated by Doxygen 1.8.14

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

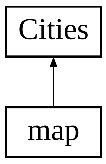
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Class Documentation

3.1 Cities Class Reference

Inheritance diagram for Cities:



Public Member Functions

- Cities ()
- bool getVaildDest (std::string)

    Checks if the city named entered is one of the cities that the company serves.
- void setName (std::string)

    It is going to set the name for the city.
- void setMap (std::string, std::string, int, int)

    Will set the names of the cities for the neighboring cities in the flightFile.txt.
- bool getVisited (std::string)

Protected Attributes

- Node ∗ **top** [40]
- Node ∗ **cityList** [40]
- Node ∗ **tryNext** [40]
- int **total**

3.1.1 Constructor & Destructor Documentation

3.1.1.1 Cities()

```
Cities::Cities ( )
```

Default

Parameters

none	

Returns

none

Precondition

called to make a default city

Postcondition

will have created a default city

3.1.2 Member Function Documentation

3.1.2.1 getVaildDest()

```
bool Cities::getVaildDest (
            std::string dest )
```

Checks if the city named entered is one of the cities that the company serves.

Parameters

dest,name	of city

Returns

validDest, which is a bool value that tells the program if the city entered is valid

Precondition

Takes in a city string to test the validity

Postcondition

Will give a bool value based on the validity of the city name

3.1.2.2 getVisited()

```
bool Cities::getVisited (
             std::string n_city )
```

Default constructor

Precondition

:

Postcondition

:

Parameters

Returns

:

3.1.2.3 setMap()

```
void Cities::setMap (
             std::string origin,
             std::string dest,
             int number,
             int cost )
```

Will set the names of the cities for the neighboring cities in the flightFile.txt.

Parameters

origin	
dest	

Returns

void

Precondition

Will take in both city names to set to the cityList post Will set the city names to the CityList

3.1.2.4 setName()

```
void Cities::setName (
             std::string name )
```

It is going to set the name for the city.

Parameters

name,name	of city

Returns

void

Precondition

Is going to take in a city name to set it

Postcondition

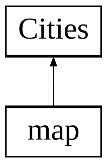
Will have set the name of the city to cityList

The documentation for this class was generated from the following files:

- problem13.h
- problem13.cpp

3.2 map Class Reference

Inheritance diagram for map:



Public Member Functions

- void markVisited (City)

  When finding the cities, this function is called to set a flag, so it doesn't come back to this city. When we visit this city it will get a value of true, meaning that this city has been cisited.
- void resetVisited ()
- std::pair< int, int > getNextCity (City, City &)
- bool check (City, City)
- void traverse (std::list< City >, std::list< int >, std::list< int >)

Additional Inherited Members

3.2.1 Member Function Documentation

3.2.1.1 check()

```
bool map::check (
             City origin,
             City dest )
```

It is going to check if there is a path between the to cities.

Parameters

origin	
dest	

Returns

bool

Precondition

to check if there is a path between cities

Postcondition

marks cities visited every run through. If there it gets NO_CITY it will pop

begin loop

get the next city adjacent to top city

check if next city is valid, NO_CITY

< backtrack

< backtrack

< backtrack

push next city onto stack, mark as visited

3.2.1.2 getNextCity()

```
std::pair< int, int > map::getNextCity (
             City palce,
             City & nextCity )
```

gets the next available city, will save also into the trNext array

Parameters

t_city	

Returns

the city

Precondition

Takes in a city to get next

Postcondition

Will check if the cities are equal and if the have already been cisited. IF all is true it will return that city. If not the will return NO_CITY

initialize variable

3.2.1.3 markVisited()

```
void map::markVisited (
            City CityVisited )
```

When finding the cities, this function is called to set a flag, so it doesn't come back to this city. When we visit this city it will get a value of true, meaning that this city has been cisited.

Parameters

CityVisited	

Returns

void

Precondition

Takes in a string to find the city in the list

Postcondition

Will loop through the names of the cities to find the city and then will set that cities private member ,visited, to true.

3.2.1.4 resetVisited()

```
void map::resetVisited ( )
```

is going to reset all the cities visited variable to false to check again for wrong and new paths.

Parameters

none	

Returns

void

Precondition

none

Postcondition

resets the bools of all the cities

3.2.1.5 traverse()

```
void map::traverse (
            std::list< City > flight,
            std::list< int > fNumber,
            std::list< int > fCost )
```

prints out all the data for the flights

Parameters

flight	
fNumber	
fCost	

Returns

void

Precondition

take in flights

Postcondition

prints the right info for the flights

opens the log3 to record the cities it goes too.

The documentation for this class was generated from the following files:

- problem13.h
- problem13.cpp

3.3 Node Class Reference

Friends

- class **Cities**
- class **map**

The documentation for this class was generated from the following files:

- problem13.h
- problem13.cpp

Index