

ALG01-semProject

1.0

Generated by Doxygen 1.10.0

1 Class Index	1
1.1 Class List	1
2 File Index	3
2.1 File List	3
3 Class Documentation	5
3.1 Map Class Reference	5
3.1.1 Detailed Description	6
3.1.2 Constructor & Destructor Documentation	6
3.1.2.1 Map()	6
3.1.2.2 ~Map()	6
3.1.3 Member Function Documentation	6
3.1.3.1 AddPoint()	6
3.1.3.2 CalculatePath()	6
3.1.3.3 GetFinish()	7
3.1.3.4 GetPath()	7
3.1.3.5 GetPoint()	7
3.1.3.6 GetStart()	7
3.1.3.7 LoadFromFile()	8
3.1.3.8 PrintMap()	8
3.2 Point Class Reference	8
3.2.1 Detailed Description	9
3.2.2 Constructor & Destructor Documentation	9
3.2.2.1 Point()	9
3.2.2.2 ~Point()	9
3.2.3 Member Function Documentation	9
3.2.3.1 AddPrevious()	9
3.2.3.2 GetDistance()	10
3.2.3.3 GetLevel()	10
3.2.3.4 GetPrevious()	10
3.2.3.5 GetX()	10
3.2.3.6 GetY()	11
3.2.3.7 PrintLevel()	11
3.2.3.8 SetDistance()	11
4 File Documentation	13
4.1 ALG01-semProject.cpp File Reference	13
4.1.1 Function Documentation	13
4.1.1.1 main()	13
4.2 Map.cpp File Reference	13
4.3 Map.h File Reference	14
4.4 Map.h	14

4.5 Point.cpp File Reference	14
4.6 Point.h File Reference	14
4.7 Point.h	15
Index	17

Chapter 1

Class Index

1.1 Class List

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

Map	Map class used for load, store and return path find results	5
Point	Point class which is composite for Map Point represents single place in map	8

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

ALG01-semProject.cpp	13
Map.cpp	13
Map.h	14
Point.cpp	14
Point.h	14

Chapter 3

Class Documentation

3.1 Map Class Reference

`Map` class used for load, store and return path find results.

```
#include <Map.h>
```

Public Member Functions

- `Map` (int rows, int cols)
`Map` object constructor for specified size.
- `~Map` ()
Destructor for deleting all objects in map object.
- `Point * GetStart` ()
Start point getter.
- `Point * GetFinish` ()
End point getter.
- void `AddPoint` (`Point *point`)
Loads point into map.
- `Point * GetPoint` (int x, int y)
Getting point from map.
- std::string `PrintMap` ()
Print actual state of `Map`.
- void `CalculatePath` ()
Calculating shortest path Implementation Breadth-first search for each neighbour Ends when reach end or check all points.
- std::vector< `Point * >` `GetPath` ()
Return results from calculated path.

Static Public Member Functions

- static `Map * LoadFromFile` (const std::string &filename)
Reading map version from file.

3.1.1 Detailed Description

[Map](#) class used for load, store and return path find results.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 Map()

```
Map::Map (
    int rows,
    int cols )
```

[Map](#) object constructor for specified size.

Parameters

<i>rows</i>	Row count
<i>cols</i>	Column count

3.1.2.2 ~Map()

```
Map::~~Map ( )
```

Destructor for deleting all objects in map object.

3.1.3 Member Function Documentation

3.1.3.1 AddPoint()

```
void Map::AddPoint (
    Point * point )
```

Loads point into map.

Parameters

<i>point</i>	point to be added
--------------	-------------------

3.1.3.2 CalculatePath()

```
void Map::CalculatePath ( )
```

Calculating shortest path Implementation Breadth-first search for each neighbour Ends when reach end or check all points.

3.1.3.3 GetFinish()

```
Point * Map::GetFinish ( )
```

End point getter.

Returns

finish point

3.1.3.4 GetPath()

```
std::vector< Point * > Map::GetPath ( )
```

Return results from calculated path.

Returns

array of Points from start to end

3.1.3.5 GetPoint()

```
Point * Map::GetPoint (
    int x,
    int y )
```

Getting point from map.

Parameters

<i>x</i>	column index in map
<i>y</i>	row index in map

Returns

selected point from map

3.1.3.6 GetStart()

```
Point * Map::GetStart ( )
```

Start point getter.

Returns

start point

3.1.3.7 LoadFromFile()

```
Map * Map::LoadFromFile (
    const std::string & filename ) [static]
```

Reading map version from file.

Parameters

<i>filename</i>	path to file with map input
-----------------	-----------------------------

Returns

new [Map](#) generated from text

3.1.3.8 PrintMap()

```
std::string Map::PrintMap ( )
```

Print actual state of [Map](#).

Returns

console printable string

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

- [Map.h](#)
- [Map.cpp](#)

3.2 Point Class Reference

[Point](#) class which is composite for [Map Point](#) represents single place in map.

```
#include <Point.h>
```

Public Member Functions

- [Point](#) (int x, int y, char level)
Creates nnew point.
- [~Point](#) ()
Destroys actual point and its references.
- int [GetX](#) ()
Getter for x directory.
- int [GetY](#) ()
Getter for y directory.
- int [GetDistance](#) ()

- Getter for distance.*
- char [GetLevel](#) ()
Getter for level replacing start with 'a' value and end with 'z'.
- char [PrintLevel](#) ()
Getter for level with real char value from map.
- void [SetDistance](#) (int distance)
Setter for distance.
- void [AddPrevious](#) ([Point](#) *point)
Method for adding reference to previous [Point](#).
- [Point](#) * [GetPrevious](#) ()
Getter for actual stored previous [Point](#).

3.2.1 Detailed Description

[Point](#) class which is composite for [Map Point](#) represents single place in map.

3.2.2 Constructor & Destructor Documentation

3.2.2.1 Point()

```
Point::Point (
    int x,
    int y,
    char level )
```

Creates nnew point.

Parameters

x	
-------------------	--

3.2.2.2 ~Point()

```
Point::~~Point ( )
```

Destroys actual point and its references.

3.2.3 Member Function Documentation

3.2.3.1 AddPrevious()

```
void Point::AddPrevious (
    Point * point )
```

Method for adding reference to previous [Point](#).

Parameters

<i>point</i>	
--------------	--

3.2.3.2 GetDistance()

```
int Point::GetDistance ( )
```

Getter for distance.

Returns

distance as integer

3.2.3.3 GetLevel()

```
char Point::GetLevel ( )
```

Getter for level replacing start with 'a' value and end with 'z'.

Returns**3.2.3.4 GetPrevious()**

```
Point * Point::GetPrevious ( )
```

Getter for actual stored previous [Point](#).

Returns

pointer to previous [Point](#)

3.2.3.5 GetX()

```
int Point::GetX ( )
```

Getter for x directory.

Returns

index x

3.2.3.6 GetY()

```
int Point::GetY ( )
```

Getter for y directory.

Returns

index y

3.2.3.7 PrintLevel()

```
char Point::PrintLevel ( )
```

Getter for level with real char value from map.

Returns

3.2.3.8 SetDistance()

```
void Point::SetDistance (
    int distance )
```

Setter for distance.

Parameters

<i>distance</i>	
-----------------	--

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

- [Point.h](#)
- [Point.cpp](#)

Chapter 4

File Documentation

4.1 ALG01-semProject.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <string>
#include "Map.h"
```

Functions

- `int main ()`
Solution for shortest-path find.

4.1.1 Function Documentation

4.1.1.1 main()

```
int main ( )
```

Solution for shortest-path find.

Returns

4.2 Map.cpp File Reference

```
#include "Map.h"
#include <iostream>
#include <queue>
```

4.3 Map.h File Reference

```
#include "Point.h"
#include <string>
#include <vector>
#include <iostream>
#include <fstream>
```

Classes

- class [Map](#)
Map class used for load, store and return path find results.

4.4 Map.h

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002 #include "Point.h"
00003 #include <string>
00004 #include <vector>
00005 #include <iostream>
00006 #include <fstream>
00010 class Map
00011 {
00012 private:
00013     std::vector<Point*> mapArray;
00014     Point* start = nullptr;
00015     Point* finish = nullptr;
00016     int rows;
00017     int cols;
00018     Point* ProcessPoint(Point* actual, Point* next);
00019 public:
00020     Map(int rows, int cols);
00021     ~Map();
00022     Point* GetStart();
00023     Point* GetFinish();
00024     void AddPoint(Point* point);
00025     Point* GetPoint(int x, int y);
00026     std::string PrintMap();
00027     void CalculatePath();
00028     std::vector<Point*> GetPath();
00029     static Map* LoadFromFile(const std::string& filename);
00030
00031
00032 };
```

4.5 Point.cpp File Reference

```
#include "Point.h"
```

4.6 Point.h File Reference

Classes

- class [Point](#)
Point class which is composite for Map Point represents single place in map.

4.7 Point.h

[Go to the documentation of this file.](#)

```
00001 #pragma once
00006 class Point
00007 {
00008 private:
00009     int x;
00010     int y;
00011     char level;
00012     int distance = 2147483647; //int maxvalue
00013     Point* previous = nullptr;
00014 public:
00015     Point(int x, int y, char level);
00016     ~Point();
00017     int GetX();
00018     int GetY();
00019     int GetDistance();
00020     char GetLevel();
00021     char PrintLevel();
00022     void SetDistance(int distance);
00023     void AddPrevious(Point* point);
00024     Point* GetPrevious();
00025
00026
00027 };
```


Index

- ~Map
 - Map, [6](#)
- ~Point
 - Point, [9](#)
- AddPoint
 - Map, [6](#)
- AddPrevious
 - Point, [9](#)
- ALG01-semProject.cpp, [13](#)
 - main, [13](#)
- CalculatePath
 - Map, [6](#)
- GetDistance
 - Point, [10](#)
- GetFinish
 - Map, [6](#)
- GetLevel
 - Point, [10](#)
- GetPath
 - Map, [7](#)
- GetPoint
 - Map, [7](#)
- GetPrevious
 - Point, [10](#)
- GetStart
 - Map, [7](#)
- GetX
 - Point, [10](#)
- GetY
 - Point, [10](#)
- LoadFromFile
 - Map, [7](#)
- main
 - ALG01-semProject.cpp, [13](#)
- Map, [5](#)
 - ~Map, [6](#)
 - AddPoint, [6](#)
 - CalculatePath, [6](#)
 - GetFinish, [6](#)
 - GetPath, [7](#)
 - GetPoint, [7](#)
 - GetStart, [7](#)
 - LoadFromFile, [7](#)
 - Map, [6](#)
 - PrintMap, [8](#)
- Map.cpp, [13](#)

- Map.h, [14](#)
- Point, [8](#)
 - ~Point, [9](#)
 - AddPrevious, [9](#)
 - GetDistance, [10](#)
 - GetLevel, [10](#)
 - GetPrevious, [10](#)
 - GetX, [10](#)
 - GetY, [10](#)
 - Point, [9](#)
 - PrintLevel, [11](#)
 - SetDistance, [11](#)
- Point.cpp, [14](#)
- Point.h, [14](#)
- PrintLevel
 - Point, [11](#)
- PrintMap
 - Map, [8](#)
- SetDistance
 - Point, [11](#)