

Projections

1.0

Generated by Doxygen 1.9.1

1 Projection of line	1
2 Data Structure Index	3
2.1 Data Structures	3
3 File Index	5
3.1 File List	5
4 Data Structure Documentation	7
4.1 Point Class Reference	7
4.1.1 Detailed Description	7
4.1.2 Constructor & Destructor Documentation	7
4.1.2.1 Point() [1/2]	7
4.1.2.2 Point() [2/2]	8
4.1.3 Member Function Documentation	8
4.1.3.1 operator[]()	8
4.1.3.2 printPoint()	8
4.1.3.3 setPoint()	8
4.1.4 Friends And Related Function Documentation	8
4.1.4.1 operator-	8
5 File Documentation	9
5.1 include/point.h File Reference	9
5.1.1 Macro Definition Documentation	9
5.1.1.1 DIM	9
5.2 src/main.cpp File Reference	9
5.2.1 Macro Definition Documentation	10
5.2.1.1 ACCUR	10
5.2.1.2 COS	10
5.2.1.3 DIST	10
5.2.1.4 DIST_BETWEEN	11
5.2.2 Function Documentation	11
5.2.2.1 calculate_dims()	11
5.2.2.2 main()	11
5.2.2.3 read_line()	11
5.3 src/point.cpp File Reference	12
5.3.1 Function Documentation	12
5.3.1.1 operator-()	12
Index	13

Chapter 1

Projection of line

Author

M.M. Pugavko

The program takes three arguments: name_file x y z name_file is file with line x, y and z are coordinates of the point
The program prints output of the following form segment n parameter s point x y z n is number of segment of line s
is a parameter that shows the part of the segment that the projection falls on. This parameter ranges from 0 to 1

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Point	7
-----------------------	-------	-------------------

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

include/point.h	9
src/main.cpp	9
src/point.cpp	12

Chapter 4

Data Structure Documentation

4.1 Point Class Reference

```
#include <point.h>
```

Public Member Functions

- [Point](#) ()
- [Point](#) (double x, double y, double z)
- void [setPoint](#) (double x, double y, double z)
- void [printPoint](#) () const
- double & [operator\[\]](#) (const int)

Friends

- [Point operator-](#) (const [Point](#) &, const [Point](#) &)

4.1.1 Detailed Description

Definition at line 4 of file point.h.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 [Point\(\)](#) [1/2]

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

Definition at line 6 of file point.cpp.

4.1.2.2 Point() [2/2]

```
Point::Point (
    double x = 0,
    double y = 0,
    double z = 0 )
```

Definition at line 11 of file point.cpp.

4.1.3 Member Function Documentation

4.1.3.1 operator[]()

```
double & Point::operator[] (
    const int index )
```

Definition at line 32 of file point.cpp.

4.1.3.2 printPoint()

```
void Point::printPoint ( ) const
```

Definition at line 23 of file point.cpp.

4.1.3.3 setPoint()

```
void Point::setPoint (
    double x = 0,
    double y = 0,
    double z = 0 )
```

Definition at line 17 of file point.cpp.

4.1.4 Friends And Related Function Documentation

4.1.4.1 operator-

```
Point operator- (
    const Point & left,
    const Point & right ) [friend]
```

Definition at line 27 of file point.cpp.

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

- [include/point.h](#)
- [src/point.cpp](#)

Chapter 5

File Documentation

5.1 include/point.h File Reference

Data Structures

- class [Point](#)

Macros

- `#define` [DIM](#) 3

5.1.1 Macro Definition Documentation

5.1.1.1 DIM

```
#define DIM 3
```

Definition at line 3 of file point.h.

5.2 src/main.cpp File Reference

```
#include "point.h"  
#include <iostream>  
#include <vector>  
#include <fstream>  
#include <string>  
#include <cmath>  
#include <stdexcept>  
#include <sstream>
```

Macros

- `#define ACCUR 1e-7`
- `#define COS(top, bottom) top / bottom`
- `#define DIST(X, Y, Z) sqrt(X * X + Y * Y + Z * Z)`
- `#define DIST_BETWEEN(x1, x2, y1, y2, z1, z2) sqrt((x1 - x2) * (x1 - x2) + (y1 - y2) * (y1 - y2) + (z1 - z2) * (z1 - z2))`

Functions

- `void read_line (vector< Point > &points, string namefile)`
This function read file. Structure of file: x0 y0 z0 x1 y1 z1 ... xn yn zn.
- `void calculate_dims (vector< Point > &points, Point &input_point)`
- `int main (int argc, char *argv[])`
This main.

5.2.1 Macro Definition Documentation

5.2.1.1 ACCUR

```
#define ACCUR 1e-7
```

Definition at line 33 of file main.cpp.

5.2.1.2 COS

```
#define COS(  
    top,  
    bottom ) top / bottom
```

Definition at line 34 of file main.cpp.

5.2.1.3 DIST

```
#define DIST(  
    X,  
    Y,  
    Z ) sqrt(X * X + Y * Y + Z * Z)
```

Definition at line 35 of file main.cpp.

5.2.1.4 DIST_BETWEEN

```
#define DIST_BETWEEN(  
    x1,  
    x2,  
    y1,  
    y2,  
    z1,  
    z2 ) sqrt((x1 - x2) * (x1 - x2) + (y1 - y2) * (y1 - y2) + (z1 - z2) * (z1 - z2))
```

Definition at line 36 of file main.cpp.

5.2.2 Function Documentation

5.2.2.1 calculate_dims()

```
void calculate_dims (  
    vector< Point > & points,  
    Point & input_point )
```

Definition at line 69 of file main.cpp.

5.2.2.2 main()

```
int main (  
    int argc,  
    char * argv[] )
```

This main.

Definition at line 140 of file main.cpp.

5.2.2.3 read_line()

```
void read_line (  
    vector< Point > & points,  
    string namefile )
```

This function read file. Structure of file: x0 y0 z0 x1 y1 z1 ... xn yn zn.

Definition at line 48 of file main.cpp.

5.3 src/point.cpp File Reference

```
#include "point.h"  
#include <iostream>
```

Functions

- [Point operator-](#) (const [Point](#) &left, const [Point](#) &right)

5.3.1 Function Documentation

5.3.1.1 operator-()

```
Point operator- (   
    const Point & left,   
    const Point & right )
```

Definition at line 27 of file point.cpp.

Index

ACCUR
 main.cpp, [10](#)

calculate_dims
 main.cpp, [11](#)

COS
 main.cpp, [10](#)

DIM
 point.h, [9](#)

DIST
 main.cpp, [10](#)

DIST_BETWEEN
 main.cpp, [10](#)

include/point.h, [9](#)

main
 main.cpp, [11](#)

main.cpp
 ACCUR, [10](#)
 calculate_dims, [11](#)
 COS, [10](#)
 DIST, [10](#)
 DIST_BETWEEN, [10](#)
 main, [11](#)
 read_line, [11](#)

operator-
 Point, [8](#)
 point.cpp, [12](#)

operator[]
 Point, [8](#)

Point, [7](#)
 operator-, [8](#)
 operator[], [8](#)
 Point, [7](#)
 printPoint, [8](#)
 setPoint, [8](#)

point.cpp
 operator-, [12](#)

point.h
 DIM, [9](#)

printPoint
 Point, [8](#)

read_line
 main.cpp, [11](#)

setPoint
 Point, [8](#)
 src/main.cpp, [9](#)
 src/point.cpp, [12](#)