Projections

1.0

Generated by Doxygen 1.9.1

Chapter 1

Projection of line

main file 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

Version

1

Date

2021-06-21

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

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 8 of file point.h.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Point() [1/2]

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

Default constructor that defines a point at the origin.

Definition at line 14 of file point.cpp.

4.1.2.2 Point() [2/2]

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

The constructor defines point.

Parameters

x,y,z	are coordinates of input point.

Definition at line 22 of file point.cpp.

4.1.3 Member Function Documentation

4.1.3.1 operator[]()

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

Indexing operator. It returns the x, y, z coordinate depending on the index from the range [0, 2].

Parameters

index	0 – x, 1 – y, 2 – z.

Definition at line 57 of file point.cpp.

4.1.3.2 printPoint()

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

The method prints point.

Definition at line 42 of file point.cpp.

4.1.3.3 setPoint()

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

The method sets the coordinates of the point.

Parameters

x,y,z	are coordinates of input.

Definition at line 32 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]
```

The operator calculates a point that is the difference between all coordinates of the other two points.

Definition at line 49 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

Point class interface.

Data Structures

- class Point

Macros

- #define DIM 3

5.1.1 Detailed Description

Point class interface.

5.1.2 Macro Definition Documentation

5.1.2.1 DIM

```
#define DIM 3
```

Definition at line 7 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 34 of file main.cpp.

5.2.1.2 COS

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

Definition at line 35 of file main.cpp.

5.2.1.3 DIST

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

Definition at line 36 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 37 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 70 of file main.cpp.

5.2.2.2 main()

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

This main.

Definition at line 142 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 49 of file main.cpp.

5.3 src/point.cpp File Reference

Implementing the Point interface.

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

Functions

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

5.3.1 Detailed Description

Implementing the Point interface.

5.3.2 Function Documentation

5.3.2.1 operator-()

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

The operator calculates a point that is the difference between all coordinates of the other two points.

Definition at line 49 of file point.cpp.

Index