Projections

1.1

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. Example:

```
./main data.dat 1 1 1
```

Output:

```
Segment 2 parameter 0.75 point 1.75 0.75 0
Segment 3 parameter 0.25 point 2.25 1 0.25
```

Version

1.1

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 sum_coordinates () const
- double & operator[ ] (const int)

Friends

- Point operator- (const Point &, const Point &)
- Point operator∗ (const Point &, const Point &)
- Point operator∗ (const Point &, const double)
- Point operator/ (const Point &, const Point &)
- Point operator/ (const Point &, const double)
- Point operator/ (const Point &, const double)

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 115 of file point.cpp.

4.1.3.2 printPoint()

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

The method prints point.

Parameters

left	left operand.
right	right operand.

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.

4.1.3.4 sum_coordinates()

```
double Point::sum_coordinates ( ) const
```

Computes sum of coordinates point.

Returns

sum of coordinates point.

Definition at line 51 of file point.cpp.

4.1.4 Friends And Related Function Documentation

4.1.4.1 operator* [1/2]

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

The operator calculates the multiplication point by double.

Parameters

left	left operand (Point).
right	right operand (double).

Definition at line 84 of file point.cpp.

4.1.4.2 operator* [2/2]

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

The operator calculates the multiplication two points.

Parameters

left	left operand (Point).
right	right operand (Point).

Definition at line 73 of file point.cpp.

4.1.4.3 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.

Parameters

left	left operand.
right	right operand.

Definition at line 63 of file point.cpp.

4.1.4.4 operator/ [1/3]

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

The operator alculates left per double number

Parameters

left	left operand (Point).
right	right operand (double).

Definition at line 104 of file point.cpp.

4.1.4.5 operator/ [2/3]

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

The operator alculates left per double number

Parameters

left	left operand (Point).
right	right operand (double).

Definition at line 104 of file point.cpp.

4.1.4.6 operator/ [3/3]

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

The operator alculates the division of the coordinates of points

Parameters

left	left operand (Point).
right	right operand (Point).

Definition at line 93 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>
#include <float.h>
```

Macros

- #define ACCUR 1e-7

  Distance measurement accuracy.
- #define DIST(x, y, z) sqrt(x ∗ x + y ∗ y + z ∗ z)

  Calculates the sum of the squares of the coordinates of a point.
- #define DIST_BETWEEN(x1, x2, y1, y2, z1, z2) sqrt((x1 - x2) ∗ (x1 - x2) + (y1 - y2) ∗ (y1 - y2) + (z1 - z2) ∗ (z1 - z2))

  Computes the distance between two input points.

Functions

- void read_line (vector< Point > &points, string namefile)
- void calculate_projections (vector< Point > &points, Point &input_point)
- void projection_print (vector< Point > &all_projections, vector< Point > &points, vector< unsigned int > &segments)
- int main (int argc, char ∗argv[ ])

5.2.1 Macro Definition Documentation

5.2.1.1 ACCUR

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

Distance measurement accuracy.

Definition at line 46 of file main.cpp.

5.2.1.2 DIST

```
#define DIST(
            x,
            y,
            z ) sqrt(x * x + y * y + z * z)
```

Calculates the sum of the squares of the coordinates of a point.

Parameters

x,y,z	– Point coordinates

Returns

the sum of the squares of the coordinates of a point

Definition at line 52 of file main.cpp.

5.2.1.3 DIST_BETWEEN

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

Computes the distance between two input points.

Parameters

(x1,x2,y1,y2,z1,z2)	two input points.

Returns

distance between two input point.

Definition at line 58 of file main.cpp.

5.2.2 Function Documentation

5.2.2.1 calculate_projections()

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

Computes all projections.

Parameters

points	a line.
input_point	an input point.

Definition at line 143 of file main.cpp.

5.2.2.2 main()

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

Definition at line 67 of file main.cpp.

5.2.2.3 projection_print()

```
void projection_print (
            vector< Point > & all_projections,
            vector< Point > & points,
            vector< unsigned int > & segments )
```

Prinnts all projections, parameters and segments.

Parameters

all_projections	all found projections.
points	a line.
segments	all found projections.

Definition at line 119 of file main.cpp.

5.2.2.4 read_line()

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

Read a line from a file

Parameters

points	the vector of dots. The data from the file is written to this vector.
namefile	the name input file.

Definition at line 92 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)
- Point operator∗ (const Point &left, const Point &right)
- Point operator∗ (const Point &left, const double right)
- Point operator/ (const Point &left, const Point &right)
- Point operator/ (const Point &left, const double right)

5.3.1 Detailed Description

Implementing the Point interface.

5.3.2 Function Documentation

5.3.2.1 operator∗() [1/2]

```
Point operator* (
            const Point & left,
            const double right )
```

The operator calculates the multiplication point by double.

Parameters

left	left operand (Point).
right	right operand (double).

Definition at line 84 of file point.cpp.

5.3.2.2 operator*() [2/2]

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

The operator calculates the multiplication two points.

Parameters

left	left operand (Point).
right	right operand (Point).

Definition at line 73 of file point.cpp.

5.3.2.3 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.

Parameters

left	left operand.
right	right operand.

Definition at line 63 of file point.cpp.

5.3.2.4 operator/() [1/2]

```
Point operator/ (
            const Point & left,
            const double right )
```

The operator alculates left per double number

Parameters

left	left operand (Point).
right	right operand (double).

Definition at line 104 of file point.cpp.

5.3.2.5 operator/() [2/2]

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

The operator alculates the division of the coordinates of points

Parameters

left	left operand (Point).
right	right operand (Point).

Definition at line 93 of file point.cpp.