ShapeLib

Generated by Doxygen 1.9.3

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 ShapeLib Namespace Reference

Classes

- class Rectangle

  Simple class representing a rectangle defined by two points.
- class Shape

Functions

- std::ostream & operator<< (std::ostream &stream, Rectangle &rect)

5.1.1 Function Documentation

5.1.1.1 operator<<()

```
std::ostream & ShapeLib::operator<< (
            std::ostream & stream,
            ShapeLib::Rectangle & rect )
```

Definition at line 42 of file rectangle.cpp.

Chapter 6

Class Documentation

6.1 Point Class Reference

Simple class representing one point in cartesian coordinates.

`#include <point.h>`

Public Member Functions

- Point (int x, int y)
  First parametric constructor.

Static Public Member Functions

- static int horizontalDistance (const Point &p1, const Point &p2)
  Static function for horizontal distance of two point estimation.
- static int verticalDistance (const Point &p1, const Point &p2)
  Static function for vertical distance of two point estimation.

Public Attributes

- int x
- int y

6.1.1 Detailed Description

Simple class representing one point in cartesian coordinates.

Definition at line 11 of file point.h.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 Point()

```
Point::Point (
            int x,
            int y )  [inline]
```

First parametric constructor.

Because at least one parametric constructor is created, the compiler doesn't create the default constructor.

Parameters

x	coordinate
y	coordinate

Definition at line 21 of file point.h.

6.1.3 Member Function Documentation

6.1.3.1 horizontalDistance()

```
int Point::horizontalDistance (
            const Point & p1,
            const Point & p2 )  [static]
```

Static function for horizontal distance of two point estimation.

See also

Point

Parameters

p1	first input point
p2	second input point

Returns

horizontal distance between provided pints

Definition at line 14 of file point.cpp.

6.1.3.2 verticalDistance()

```
int Point::verticalDistance (
            const Point & p1,
            const Point & p2 )  [static]
```

Static function for vertical distance of two point estimation.

See also

Point

Parameters

p1	first input point
p2	second input point

Returns

vertical distance between provided points

Definition at line 24 of file point.cpp.

6.1.4 Member Data Documentation

6.1.4.1 x

```
int Point::x
```

coordinate X

Definition at line 26 of file point.h.

6.1.4.2 y

```
int Point::y
```

coordinate Y

Definition at line 27 of file point.h.

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

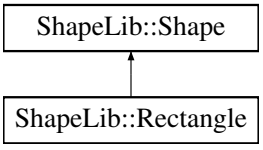
- /Users/peterjanku/SourcesCpp/AK5PC_Shape2D/ShapeLib/include/point.h
- /Users/peterjanku/SourcesCpp/AK5PC_Shape2D/ShapeLib/src/point.cpp

6.2 ShapeLib::Rectangle Class Reference

Simple class representing a rectangle defined by two points.

```
#include <rectangle.h>
```

Inheritance diagram for ShapeLib::Rectangle:



Public Member Functions

- Rectangle (int id, int x1, int y1, int x2, int y2)

  The rectangle constructor.
- Rectangle (int id, const Point &p1, const Point &p2)

  The main rectangle constructor.
- void printInfo () const override
- int getWidth ()

  This function returns the current width of the rectangle. If the already calculated width is incorrect(the rectangle points were redefined), the dimensions are recalculated at first.
- int getHeight ()

  This function returns the current height of the rectangle. If the already calculated height is incorrect(the rectangle points were redefined), the dimensions are recalculated at first.
- const Point & getP1 () const

  Return the first point in 2D space, which defines the rectangle.
- const Point & getP2 () const

  Return the second point in 2D space, which defines the rectangle.

Protected Member Functions

- void recalculateDim ()

Protected Attributes

- Point p1
- Point p2
- int width
- int height
- bool dimValid = false

6.2.1 Detailed Description

Simple class representing a rectangle defined by two points.

See also

Point

Definition at line 16 of file rectangle.h.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 Rectangle() [1/2]

```
ShapeLib::Rectangle::Rectangle (
            int id,
            int x1,
            int y1,
            int x2,
            int y2 )  [inline]
```

The rectangle constructor.

This constructor is calling the second one with modified parameters. This can be done since C++11 standard and it's called constructor delegation.

Parameters

id	Rectangle ID
x1	X coordinate of first rectangle's point
y1	Y coordinate of first rectangle's point
x2	X coordinate of second rectangle's point
y2	Y coordinate of second rectangle's point

Definition at line 29 of file rectangle.h.

6.2.2.2 Rectangle() [2/2]

```
ShapeLib::Rectangle::Rectangle (
            int id,
            const Point & p1,
            const Point & p2 )  [inline]
```

The main rectangle constructor.

This constructor stores all Rectangle's internal members as well as members defined in Shape class. It also calls constructor of Shape class. Finally, it recalculates the value dimensions of the rectangle.

See also

recalculateDim()

Parameters

id	
p1	
p2	

Definition at line 40 of file rectangle.h.

6.2.3 Member Function Documentation

6.2.3.1 getHeight()

```
int ShapeLib::Rectangle::getHeight ( )
```

This function returns the current height of the rectangle. If the already calculated height is incorrect(the rectangle points were redefined), the dimensions are recalculated at first.

See also

recalculateDim()

Returns

current height of the rectangle.

Definition at line 23 of file rectangle.cpp.

6.2.3.2 getP1()

```
const Point & ShapeLib::Rectangle::getP1 ( ) const  [inline]
```

Return the first point in 2D space, which defines the rectangle.

Returns

Const reference to the point

See also

Point

Definition at line 52 of file rectangle.h.

6.2.3.3 getP2()

```
const Point & ShapeLib::Rectangle::getP2 ( ) const  [inline]
```

Return the second point in 2D space, which defines the rectangle.

Returns

Const reference to the point

See also

Point

Definition at line 58 of file rectangle.h.

6.2.3.4 getWidth()

```
int ShapeLib::Rectangle::getWidth ( )
```

This function returns the current width of the rectangle. If the already calculated width is incorrect(the rectangle points were redefined), the dimensions are recalculated at first.

See also

recalculateDim()

Returns

current width of the rectangle.

Definition at line 12 of file rectangle.cpp.

6.2.3.5 printInfo()

```
void ShapeLib::Rectangle::printInfo ( ) const  [override], [virtual]
```

Implements ShapeLib::Shape.

Definition at line 36 of file rectangle.cpp.

6.2.3.6 recalculateDim()

```
void ShapeLib::Rectangle::recalculateDim ( )  [protected]
```

Definition at line 30 of file rectangle.cpp.

6.2.4 Member Data Documentation

6.2.4.1 dimValid

```
bool ShapeLib::Rectangle::dimValid = false  [mutable], [protected]
```

Information if width and height is correct or not.

See also

recalculateDim()

Definition at line 68 of file rectangle.h.

6.2.4.2 height

```
int ShapeLib::Rectangle::height  [mutable], [protected]
```

Temporary storage of rectangle height. Don't have to be correct!!!!

See also

dimValic

Definition at line 67 of file rectangle.h.

6.2.4.3 p1

```
Point ShapeLib::Rectangle::p1  [protected]
```

The first point in 2D space

Definition at line 64 of file rectangle.h.

6.2.4.4 p2

```
Point ShapeLib::Rectangle::p2  [protected]
```

The second point in 2D space

Definition at line 65 of file rectangle.h.

6.2.4.5 width

```
int ShapeLib::Rectangle::width  [mutable], [protected]
```

Temporary storage of rectangle width. Don't have to be correct!!!!

See also

dimValid

Definition at line 66 of file rectangle.h.

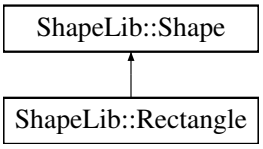
The documentation for this class was generated from the following files:

- /Users/peterjanku/SourcesCpp/AK5PC_Shape2D/ShapeLib/include/rectangle.h
- /Users/peterjanku/SourcesCpp/AK5PC_Shape2D/ShapeLib/src/rectangle.cpp

6.3 ShapeLib::Shape Class Reference

`#include <shape.h>`

Inheritance diagram for ShapeLib::Shape:



Public Member Functions

- Shape (int id)
- virtual void printInfo () const =0
- int getID () const

Protected Attributes

- int ID

6.3.1 Detailed Description

Definition at line 12 of file shape.h.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 Shape()

```
ShapeLib::Shape::Shape (
            int id )  [inline], [explicit]
```

Definition at line 14 of file shape.h.

6.3.3 Member Function Documentation

6.3.3.1 getID()

```
int ShapeLib::Shape::getID ( ) const  [inline]
```

Definition at line 18 of file shape.h.

6.3.3.2 printInfo()

```
virtual void ShapeLib::Shape::printInfo ( ) const  [pure virtual]
```

Implemented in ShapeLib::Rectangle.

6.3.4 Member Data Documentation

6.3.4.1 ID

```
int ShapeLib::Shape::ID  [protected]
```

Definition at line 21 of file shape.h.

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

- /Users/peterjanku/SourcesCpp/AK5PC_Shape2D/ShapeLib/include/shape.h

Chapter 7

File Documentation

7.1 /Users/peterjanku/SourcesCpp/AK5PC_Shape2D/ShapeLib/include/point.h File Reference

Classes

- class Point

  Simple class representing one point in cartesian coordinates.

7.2 point.h

Go to the documentation of this file.

```
00001 //
00002 // Created by Peter Janků on 08.10.2022.
00003 //
00004
00005 #ifndef SHAPE2D_POINT_H
00006 #define SHAPE2D_POINT_H
00007
00011 class Point {
00012 public:
00013
00021     Point(int x, int y) : x(x), y(y) {}
00022
00023     static int horizontalDistance(const Point& p1, const Point& p2);
00024     static int verticalDistance(const Point& p1, const Point& p2);
00025
00026     int x;
00027     int y;
00028 };
00029
00030
00031 #endif //SHAPE2D_POINT_H
```

7.3 /Users/peterjanku/SourcesCpp/AK5PC_Shape2D/ShapeLib/include/rectangle.h File Reference

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

Classes

- class ShapeLib::Rectangle

  Simple class representing a rectangle defined by two points.

Namespaces

- namespace ShapeLib

Functions

- std::ostream & ShapeLib::operator<< (std::ostream &stream, Rectangle &rect)

7.4 rectangle.h

Go to the documentation of this file.

```
00001 //
00002 // Created by Peter Janků on 08.10.2022.
00003 //
00004
00005 #ifndef SHAPE2D_RECTANGLE_H
00006 #define SHAPE2D_RECTANGLE_H
00007
00008 #include <iostream>
00009 #include "shape.h"
00010 #include "point.h"
00011
00012 namespace ShapeLib {
00016     class Rectangle : public Shape {
00017     public:
00018
00029         Rectangle(int id, int x1, int y1, int x2, int y2) : Rectangle(id, Point(x1, y1), Point(x2,
       y2)) {}
00030
00040         Rectangle(int id, const Point &p1, const Point &p2) : Shape(id), p1(p1), p2(p2) {
       recalculateDim(); }
00041
00042         void printInfo() const override;
00043
00044         int getWidth();
00045
00046         int getHeight();
00047
00052         const Point &getP1() const { return p1; }
00053
00058         const Point &getP2() const { return p2; }
00059
00060     protected:
00061
00062         void recalculateDim();
00063
00064         Point p1;
00065         Point p2;
00066         mutable int width;
00067         mutable int height;
00068         mutable bool dimValid = false;
00071     };
00072
00073
00074     std::ostream &operator«(std::ostream &stream, Rectangle &rect);
00075 }
00076
00077
00078 #endif //SHAPE2D_RECTANGLE_H
```

7.5 /Users/peterjanku/SourcesCpp/AK5PC_Shape2D/ShapeLib/include/shape.h File Reference

```
#include "point.h"
```

Classes

- class ShapeLib::Shape

Namespaces

- namespace ShapeLib

7.6 shape.h

Go to the documentation of this file.

```
00001 //
00002 // Created by Peter Janků on 08.10.2022.
00003 //
00004
00005 #ifndef SHAPE2D_SHAPE_H
00006 #define SHAPE2D_SHAPE_H
00007
00008 #include "point.h"
00009
00010 namespace ShapeLib {
00011
00012     class Shape {
00013     public:
00014         explicit Shape(int id):ID(id){}
00015
00016         virtual void printInfo() const = 0;
00017
00018         int getID()const { return ID; }
00019
00020     protected:
00021         int ID;
00022     };
00023
00024 } // ShapeLib
00025
00026 #endif //SHAPE2D_SHAPE_H
```

7.7 /Users/peterjanku/SourcesCpp/AK5PC_Shape2D/ShapeLib/src/point.cpp File Reference

```
#include "../include/point.h"
#include <cmath>
```

7.8 point.cpp

Go to the documentation of this file.

```
00001 //
00002 // Created by Peter Janků on 08.10.2022.
00003 //
00004
00005 #include "../include/point.h"
00006 #include <cmath>
00007
00014 int Point::horizontalDistance(const Point &p1, const Point &p2) {
00015     return std::abs(p1.x - p2.x);
00016 }
00017
00024 int Point::verticalDistance(const Point &p1, const Point &p2) {
00025     return std::abs(p1.y - p2.y);
00026 }
```

7.9 /Users/peterjanku/SourcesCpp/AK5PC_Shape2D/ShapeLib/src/rectangle.cpp File Reference

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

7.10 rectangle.cpp

Go to the documentation of this file.

```
00001 //
00002 // Created by Peter Janků on 08.10.2022.
00003 //
00004 #include <iostream>
00005 #include "rectangle.h"
00006
00012 int ShapeLib::Rectangle::getWidth() {
00013     if (!dimValid){
00014         recalculateDim();
00015     }
00016     return width;
00017 }
00023 int ShapeLib::Rectangle::getHeight() {
00024     if (!dimValid){
00025         recalculateDim();
00026     }
00027     return height;
00028 }
00029
00030 void ShapeLib::Rectangle::recalculateDim() {
00031     width = Point::horizontalDistance(p1,p2);
00032     height = Point::verticalDistance(p1,p2);
00033     dimValid = true;
00034 }
00035
00036 void ShapeLib::Rectangle::printInfo() const {
00037     std::cout « "Rectangle ID:" «  ID « std::endl;
00038     std::cout « "\tP1 (" « p1.x « "," « p1.y « ")" « std::endl;
00039     std::cout « "\tP2 (" « p2.x « "," « p2.y « ")" « std::endl;
00040 }
00041
00042 std::ostream &ShapeLib::operator«(std::ostream &stream, ShapeLib::Rectangle &rect) {
00043     std::cout « "Rectangle ID:" «  rect.getID() « std::endl;
00044     std::cout « "\tP1 (" « rect.getP1().x « "," « rect.getP1().y « ")" « std::endl;
00045     std::cout « "\tP2 (" « rect.getP2().x « "," « rect.getP2().y « ")" « std::endl;
00046     std::cout « "\tWidth: " « rect.getWidth() « std::endl;
00047     std::cout « "\tHeight:" « rect.getHeight() « std::endl;
00048
00049
00050     return stream;
00051 }
```

7.11 /Users/peterjanku/SourcesCpp/AK5PC_Shape2D/ShapeLib/src/shape.cpp File Reference

```
#include "shape.h"
```

Namespaces

- namespace ShapeLib

7.12 shape.cpp

Go to the documentation of this file.

```
00001 //
00002 // Created by Peter Janků on 08.10.2022.
00003 //
00004
00005 #include "shape.h"
00006
00007 namespace ShapeLib {
00008 } // ShapeLib
```

Index