

Class Triangle

java.lang.Object
└ **Triangle**

```
public class Triangle  
extends java.lang.Object
```

Triangle. The main function takes 3 positive whole-number lengths to be typed in as command line arguments. The program responds with a description of the triangle, as follows:

- **equilateral** - if all three sides have equal length
- **isosceles** - if two sides have equal length
- **right-angled** - if one angle is a right angle
- **scalene** - all sides different lengths, no right angles
- **impossible** - if the given side lengths do not form a triangle

Area and perimeter of the triangle are calculated, too.

Constructor Summary

Triangle (int s1, int s2, int s3) Constructor (without error checking)	
--	--

Method Summary

java.lang.String	classify () Classifies the triangle.
double	getArea () Gets the area of the triangle.
int	getPerimeter () Gets the perimeter of the triangle.
java.lang.String	getSideLengths () Gets the side lengths.
boolean	isEquilateral () Checks if the triangle is equilateral.
boolean	isImpossible () Checks whether the given side lengths form a valid triangle.
boolean	isIsosceles () Checks if the triangle is isosceles.

boolean	isRightAngled() Checks if the triangle is right-angled.
boolean	isScalene() Checks if the triangle is scalene.
static void	main (java.lang.String[] args) Usage: java Quadrangle <side1:int> <side2:int> <side3:int> Main method is only used for testing purposes, no unit tests need to be written for this method.
Triangle	setSideLengths (int s1, int s2, int s3) Sets the lengths of the sides of this triangle.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Triangle

```
public Triangle(int s1,
                int s2,
                int s3)
```

Constructor (without error checking)

Parameters:

- s1 - length of the side1 as an integer.
- s2 - length of the side2 as an integer.
- s3 - length of the side3 as an integer.

Method Detail

setSideLengths

```
public Triangle setSideLengths(int s1,  
                                int s2,  
                                int s3)
```

Sets the lengths of the sides of this triangle.

Parameters:

s1 - length of the side1

s2 - length of the side2

s3 - length of the side3

Returns:

a reference to this triangle.

getSideLengths

```
public java.lang.String getSideLengths()
```

Gets the side lengths.

Returns:

a comma separated list of side lengths

getPerimeter

```
public int getPerimeter()
```

Gets the perimeter of the triangle.

Returns:

-1 if input values are invalid, otherwise the perimeter.

getArea

```
public double getArea()
```

Gets the area of the triangle.

Returns:

area of the triangle, -1.0 if triangle is impossible.

classify

```
public java.lang.String classify()
```

Classifies the triangle. Possible types, returned as a string, are:

- equilateral - if all three sides have equal length
- isosceles - if two sides have equal length
- right-angled - if one angle is a right angle
- scalene - all sides different lengths, no right angles
- impossible - if the lengths can't form a triangle

Returns:

type of the triangle as a string.

isIsosceles

```
public boolean isIsosceles()
```

Checks if the triangle is isosceles. Note: isosceles triangle may also be equilateral and right-angled.

Returns:

true if two sides have equal length

isEquilateral

```
public boolean isEquilateral()
```

Checks if the triangle is equilateral.

Returns:

true if all three sides have equal length.

isRightAngled

```
public boolean isRightAngled()
```

Checks if the triangle is right-angled. Note: right-angled triangle may also be isosceles.

Returns:

true if one angle is a right angle, otherwise false.

isScalene

```
public boolean isScalene()
```

Checks if the triangle is scalene.

Returns:
true if all sides different lengths, no right angles.

isImpossible

```
public boolean isImpossible()
```

Checks whether the given side lengths form a valid triangle.

Returns:
true if the given side lengths do not form a triangle.

main

```
public static void main(java.lang.String[] args)
```

Usage: java Quadrangle <side1:int> <side2:int> <side3:int>

Main method is only used for testing purposes, no unit tests need to be written for this method.