



Project ▾
Abstraction C:\Users\lrush\IdeaPro
 > .idea
 > out
 ▾ src
 @ Circle
 @ Main
 @ Rectangle
 @ Shape
 @ Square
 ⊗ .gitignore
 Ⓜ Abstraction.iml
 > External Libraries
 Scratches and Consoles

Shape.java Circle.java × Rectangle.java Square.java Main.java
2 usages
1 public class Circle extends Shape {
 9 usages
2 protected double radius;
3
 no usages
4 public Circle() {
5 super();
6 this.radius = 1.0;
7 }
8
 no usages
9 public Circle(double radius) {
10 super();
11 this.radius = radius;
12 }
13
 1 usage
14 public Circle(double radius, String color, boolean filled) {
15 super(color, filled);
16 this.radius = radius;
17 }
18
 no usages
19 public double getRadius() {

Project ▾
Abstraction C:\Users\lrush\IdeaPro
 > .idea
 > out
 ▾ src
 @ Circle
 @ Main
 @ Rectangle
 @ Shape
 @ Square
 ⊗ .gitignore
 Ⓜ Abstraction.iml
 > External Libraries
 Scratches and Consoles

Shape.java Circle.java × Rectangle.java Square.java Main.java
14 public Circle(double radius, String color, boolean filled) {
15 super(color, filled);
16 this.radius = radius;
17 }
18
 no usages
19 public double getRadius() {
20 return radius;
21 }
22
 no usages
23 public void setRadius(double radius) {
24 this.radius = radius;
25 }
26
 3 usages
27 @Override
28 public double getArea() {
29 return Math.PI * radius * radius;
30 }
31
 3 usages
32 @Override
33 public double getPerimeter() {
34 return 2 * Math.PI * radius;

Project ▾
Abstraction C:\Users\lrush\IdeaPro
 > .idea
 > out
 ▾ src
 @ Circle
 @ Main
 @ Rectangle
 @ Shape
 @ Square
 ⊗ .gitignore
 Ⓜ Abstraction.iml
 > External Libraries
 Scratches and Consoles

Shape.java Circle.java × Rectangle.java Square.java Main.java
24 this.radius = radius;
25 }
26
 3 usages
27 @Override
28 public double getArea() {
29 return Math.PI * radius * radius;
30 }
31
 3 usages
32 @Override
33 public double getPerimeter() {
34 return 2 * Math.PI * radius;
35 }
36
 3 usages
37 @Override
38 public String toString() {
39 return "A Circle with radius=" + radius + ", which is a subclass of " + super.toString();
40 }
41 }

Project ▾
Abstraction C:\Users\irush\IdeaPro
 > .idea
 > out
 > src
 @ Circle
 @ Main
 @ Rectangle
 @ Shape
 @ Square
 ⊗ .gitignore
 Ⓜ Abstraction.iml
 > External Libraries
 ≡ Scratches and Consoles

Shape.java Circle.java Rectangle.java × Square.java Main.java
1 3 usages 1 inheritor
2 public class Rectangle extends Shape {
3 9 usages
4 protected double width;
5 8 usages
6 protected double length;
7
8 1 usage
9 public Rectangle() {
10 super();
11 this.width = 1.0;
12 this.length = 1.0;
13 }
14
15 1 usage
16 public Rectangle(double width, double length) {
17 super();
18 this.width = width;
19 this.length = length;
20 }
21
22 2 usages
23 public Rectangle(double width, double length, String color, boolean filled) {
24 super(color, filled);
25 this.width = width;
26 }
27 }
28
29
30
31
32
33
34
35
36
37

Project ▾
Abstraction C:\Users\irush\IdeaPro
 > .idea
 > out
 > src
 ⊗ .gitignore
 Ⓜ Abstraction.iml
 > External Libraries
 ≡ Scratches and Consoles

Shape.java Circle.java Rectangle.java × Square.java Main.java
19 this.width = width;
20 this.length = length;
21 }
22
23 no usages
24 public double getWidth() {
25 return width;
26 }
27
28 3 usages 1 override
29 public void setWidth(double width) {
30 this.width = width;
31 }
32
33 no usages
34 public double getLength() {
35 return length;
36 }
37
38 3 usages 1 override
39 public void setLength(double length) {
40 this.length = length;
41 }
42 }
43
44
45
46
47
48
49
50
51
52
53
54

Project ▾
Abstraction C:\Users\irush\IdeaPro
 > .idea
 > out
 > src
 ⊗ .gitignore
 Ⓜ Abstraction.iml
 > External Libraries
 ≡ Scratches and Consoles

Shape.java Circle.java Rectangle.java × Square.java Main.java
35 this.length = length;
36 }
37
38 3 usages
39 @Override
40 public double getArea() {
41 return width * length;
42 }
43
44 3 usages
45 @Override
46 public double getPerimeter() {
47 return 2 * (width + length);
48 }
49
50 1 override
51 @Override
52 public String toString() {
53 return "A Rectangle with width=" + width + " and length=" + length + ", which is a subclass of " + super.toString();
54 }
55 }
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

Project ▾

Abstraction C:\Users\irush\IdeaPro
 > .idea
 > out
 > src
 .gitignore
 Abstraction.iml
 > External Libraries
 Scratches and Consoles

Shape.java

Circle.java

Rectangle.java

Square.java ×

Main.java

2 usages

1 class Square extends Rectangle {
2 no usages
3 public Square() {
4 super();
5 }

6 no usages
7 public Square(double side) {
8 super(side, side);
9 }

10 1 usage
11 public Square(double side, String color, boolean filled) {
12 super(side, side, color, filled);
13 }

14 no usages
15 public double getSide() {
16 return width;
17 }

18 no usages
19 public void setSide(double side) {
20 setWidth(side);
21 }

Project ▾

Abstraction C:\Users\irush\IdeaPro
 > .idea
 > out
 > src
 .gitignore
 Abstraction.iml
 > External Libraries
 Scratches and Consoles

Shape.java

Circle.java

Rectangle.java

Square.java ×

Main.java

no usages

17
18 public void setSide(double side) {
19 setWidth(side);
20 setLength(side);
21 }

22 3 usages
23 @Override
24 public void setWidth(double side) {
25 super.setWidth(side);
26 super.setLength(side);
27 }

28 3 usages
29 @Override
30 public void setLength(double side) {
31 super.setWidth(side);
32 super.setLength(side);
33 }

34 @Override
35 public String toString() {
36 return "Square[" + super.toString()+"]";
37 }

