Shapes & Rectangles

CPSC 1181 – O.O.

Jeremy Hilliker

Summer 2017

Langara.
THE COLLEGE OF HIGHER LEARNING.

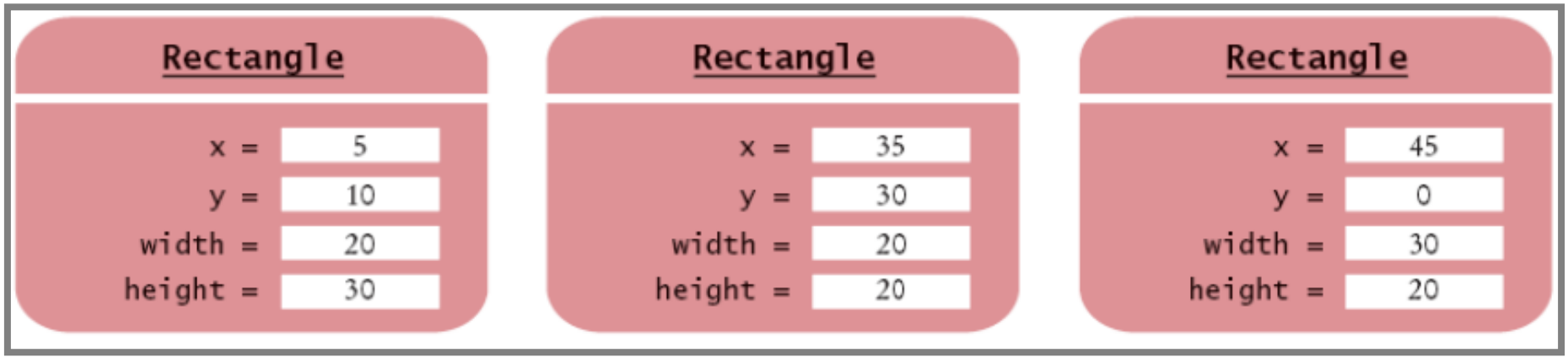
Objectives

- Learn more about methods and constructors
- Identify:
  - Objects
  - Constructors
  - Methods
  - Params: explicit/implicit
  - Params: formal/actual
  - Accessors
  - Mutators
- Learn to read the API's documentation
  - JSE 8 javadoc

First off

- Is a rectangle an object?
  - Identity
    - Singular whole
    - One is “different” from another
    - Has some kind of identifier
  - State
    - Has properties / attributes that belong to the identity
  - Behaviour
    - Can do things, or
    - Can have things done to it

- Instances of the Rectangle class
  - (Objects of type Rectangle)
  - Describe rectangular shapes
- A Rectangle object isn't a rectangular shape
  - it is an object with attributes that describe the rectangle



Mover.ja

```java
import java.awt.Rectangle;

/**
 * Moves some boxes.
 */
public class Mover {

   public static void main(String[] args) {

      Rectangle box = new Rectangle(5, 10, 20, 30);
      System.out.println("Top left: " + box.getX() + ",\t" + box.getY());

      box.translate(15,25);
      System.out.println("Top left: " + box.getX() + ",\t" + box.getY());
   }
}
```

Identify:
Objects
Constructors
Methods
Params: explicit/implicit
Params: formal/actual
Accessors
Mutators

```
$ javac Mover.java ; java Mover
Top left: 5.0,  10.0
Top left: 20.0, 35.0
```

Javadoc

- Learn to love javadoc !
- The full java API is documented
  - (Application Programming Interface)
  - Classes, attributes, methods, interfaces (later)

- https://docs.oracle.com/javase/8/docs/api/

java.awt

Class Rectangle

java.lang.Object
    java.awt.geom.RectangularShape
        java.awt.geom.Rectangle2D
            java.awt.Rectangle

All Implemented Interfaces:

`Shape, Serializable, Cloneable`

Direct Known Subclasses:

`DefaultCaret`

```
public class Rectangle
extends Rectangle2D
implements Shape, Serializable
```

A `Rectangle` specifies an area in a coordinate space that is enclosed by the `Rectangle` object's upper-left point `(x,y)` in the coordinate space, its width, and its height.

A `Rectangle` object's `width` and `height` are `public` fields. The constructors that create a `Rectangle`, and the methods that can modify one, do not prevent setting a negative value for width or height.

A `Rectangle` whose width or height is exactly zero has location along those axes with zero dimension, but is otherwise considered empty. The `isEmpty()` method will return true for such a `Rectangle`. Methods which test if an empty `Rectangle` contains or intersects a point or rectangle will always return false if either dimension is zero. Methods which combine such a `Rectangle` with a point or rectangle will include the location of the `Rectangle` on that axis in the result as if the `add(Point)` method were being

Constructor Summary

Constructors

Constructor and Description
`Rectangle()` Constructs a new `Rectangle` whose upper-left corner is at (0, 0) in the coordinate space, and whose width and height are both zero.
`Rectangle(Dimension d)` Constructs a new `Rectangle` whose top left corner is (0, 0) and whose width and height are specified by the `Dimension` argument.
`Rectangle(int width, int height)` Constructs a new `Rectangle` whose upper-left corner is at (0, 0) in the coordinate space, and whose width and height are specified by the arguments of the same name.
`Rectangle(int x, int y, int width, int height)` Constructs a new `Rectangle` whose upper-left corner is specified as `(x,y)` and whose width and height are specified by the arguments of the same name.
`Rectangle(Point p)` Constructs a new `Rectangle` whose upper-left corner is the specified `Point`, and whose width and height are both zero.
`Rectangle(Point p, Dimension d)` Constructs a new `Rectangle` whose upper-left corner is specified by the **`Point`** argument, and whose width and height are specified by the **`Dimension`** argument.
`Rectangle(Rectangle r)` Constructs a new `Rectangle`, initialized to match the values of the specified `Rectangle`.

Method Summary

All Methods | Instance Methods | Concrete Methods | Deprecated Methods

Modifier and Type	Method and Description
void	**add**(int newx, int newy) Adds a point, specified by the integer arguments newx,newy to the bounds of this Rectangle.
void	**add**(Point pt) Adds the specified Point to the bounds of this Rectangle.
void	**add**(Rectangle r) Adds a Rectangle to this Rectangle.
boolean	**contains**(int x, int y) Checks whether or not this Rectangle contains the point at the specified location (x,y).
boolean	**contains**(int X, int Y, int W, int H) Checks whether this Rectangle entirely contains the Rectangle at the specified location (X,Y) with the specified dimensions (W,H).
boolean	**contains**(Point p) Checks whether or not this Rectangle contains the specified Point.
boolean	**contains**(Rectangle r) Checks whether or not this Rectangle entirely contains the specified Rectangle.
Rectangle2D	**createIntersection**(Rectangle2D r) Returns a new Rectangle2D object representing the intersection of this Rectangle2D with the specified Rectangle2D.

translate

```
public void translate(int dx,
                      int dy)
```

Translates this `Rectangle` the indicated distance, to the right along the X coordinate axis, and downward along the Y coordinate axis.

Parameters:

`dx - the distance to move this Rectangle along the X axis`

`dy - the distance to move this Rectangle along the Y axis`

See Also:

`setLocation(int, int), setLocation(java.awt.Point)`

Recap

- Learned more about methods and constructors
- More about instantiating an object

- javadoc !