EXTRA CLASS

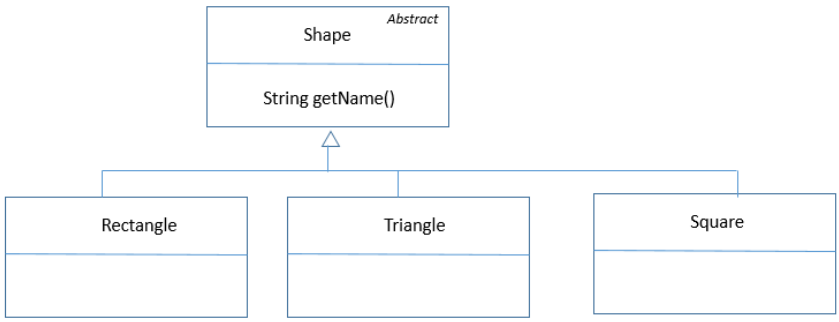
JAVA EXTRA CLASS - OBJECTS

CHALLENGE 1

To make the project work with no errors, you need to do the following

1. Create the following 3 classes which extends from Shape.
   - Rectangle
   - Triangle
   - Square

   Note*: create each class in a separate file.*



2. Implement the required method getName() which shall return the name of the shape.

Now the Main should work and display this output:

```
Rectangle
Rectangle
Triangle
Square
```

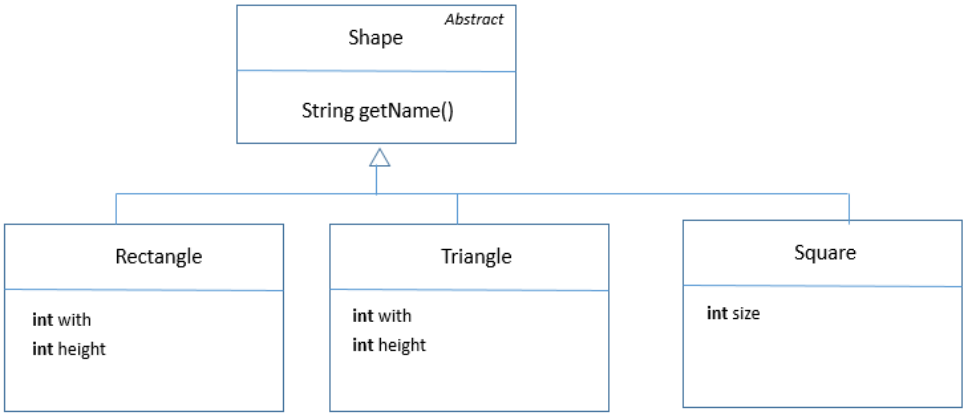
CHALLENGE 2

Now the shapes have some dimensions!!

But each shape manage dimension in a different way.

1. Add the following attributes to each shapes
2. Also add a constructor on each class to initialize those parameter
3. Update main :: replace shapes initialization code with the following code :

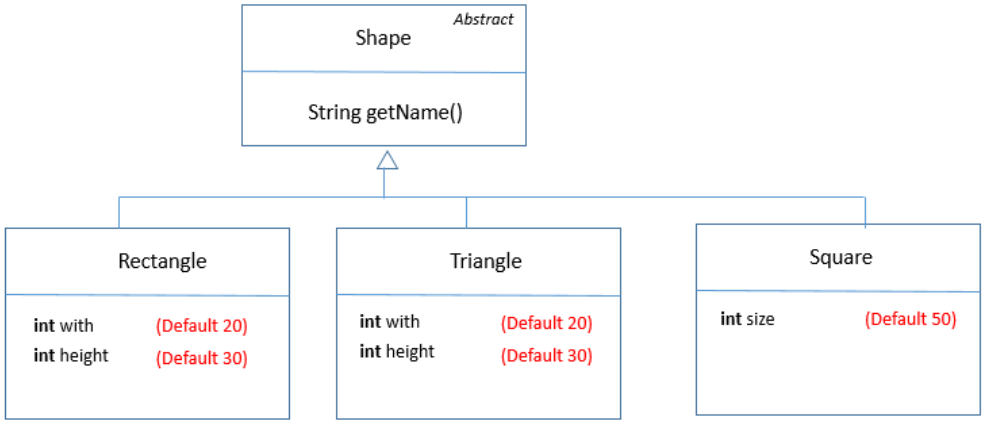
```
shapes.add(new Rectangle(10, 15));
shapes.add(new Rectangle(20, 30));
shapes.add(new Triangle(15, 15));
shapes.add(new Square(60));
```



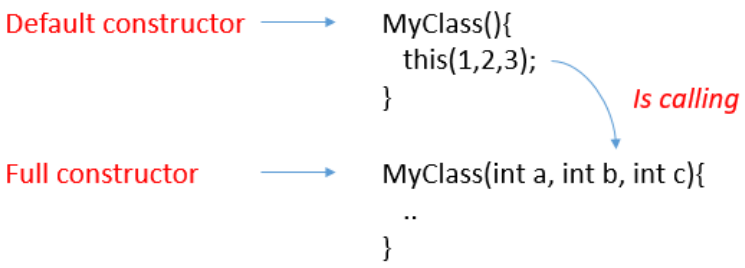
CHALLENGE 3

Now we want to be able to create shapes with default values.

1. Add a **default constructor** on each shapes to initialize shapes with the following default values (in red)



2. Update you code a bit : on each shape class, the default constructor shall call the full constructor with the default values, as show below



3. Update main : replace shapes initialization code with the following code :

```
shapes.add(new Rectangle(10, 15));
shapes.add(new Rectangle());                // rectangle with default constructor

shapes.add(new Triangle(15, 15));
shapes.add(new Triangle());                 // Triangle with default constructor

shapes.add(new Square(60));
shapes.add(new Square());                   // Square with default constructor
```

CHALLENGE 4

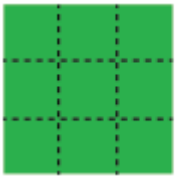
We want the shapes to provide their **perimeter.**

1. Add an abstract method getPerimeter() to the abstract Shape
2. Implement this method in all your shape classes, doing the perimeter computation of the related shape
3. Update the main :
    a. You shall display as output the sum of all the perimeters of the defined shapes

CHALLENGE 5

Same as challenge 4, but regarding the area of the shapes.

You shall display as output the sum of all the areas of the defined shapes



AREA

The amount of space inside a shape

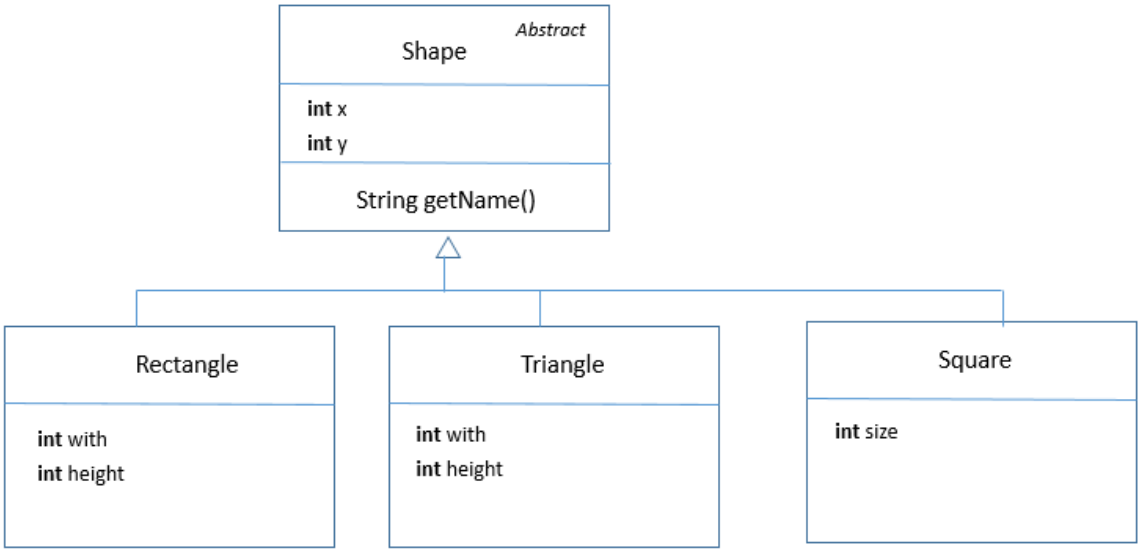
CHALLENGE 6

All shapes have positions X and Y!!

Update:

- Abstract class Shape
- All concrete shapes classes
- The Main class

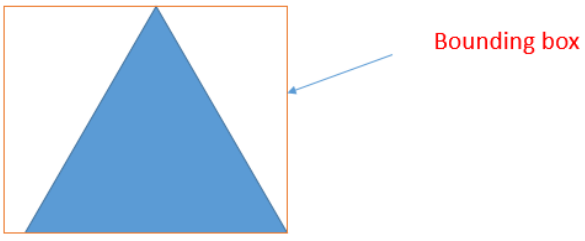
To manage the position of the shapes



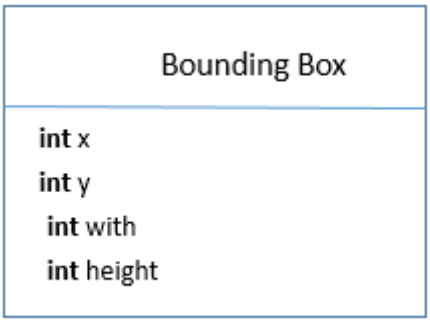
CHALLENGE 7

We want each shape to provide its bounding box

Why: the bounding box can be useful to detect collision/overlap s easily between shapes



- Define the class BoundingBox and its constructor as follow



- As the following method to the abstract class shape and implement this method on all Shapes