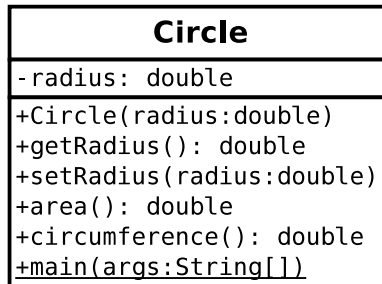


Model 1 The Circle Class

Unified Modeling Language (UML) provides a way of graphically illustrating a class's design, independent of the programming language.



Questions (15 min)

Start time:

1. Consider the Circle class:

a) How many attributes does it have?

b) How many methods does it have?

2. Based on ?? and Model 1, what is typically **public** and what is typically **private**?

Methods are typically public, and attributes are typically private.

The following questions will have you implement the Circle class exactly as shown in the UML diagram above. Do not worry about writing Javadoc comments for this activity.

3. Write the code that declares the radius attribute. An outline of *Circle.java* is provided below for context.

```
public class Circle {  
    private double radius;  
  
    // constructor goes here  
  
    // other methods go here  
}
```

4. Write the code for the Circle constructor. Notice that, in contrast to ??, the Circle constructor has a parameter. Assign the parameter radius to the attribute **this.radius**.

```
public Circle(double radius) {  
    this.radius = radius;  
}
```

5. Write the code for `getRadius`. (Refer to ?? for an example.)

```
public double getRadius() {  
    return this.radius;  
}
```

6. Write the code for `setRadius`. Like the constructor, it should assign the parameter to the corresponding attribute.

```
public void setRadius(double radius) {  
    this.radius = radius;  
}
```

7. Write the code for `area`. The area of a circle is πr^2 .

```
public double area() {  
    return Math.PI * radius * radius;  
}
```

8. Write the code for `circumference`. The circumference of a circle is $2\pi r$.

```
public double circumference() {  
    return 2.0 * Math.PI * radius;  
}
```

9. Write a main method that creates a `Circle` object with a radius of 2.0 and displays its area and circumference (using `println`).

```
public static void main(String[] args) {  
    Circle circle = new Circle(2);  
    System.out.println(circle.area());  
    System.out.println(circle.circumference());  
}
```