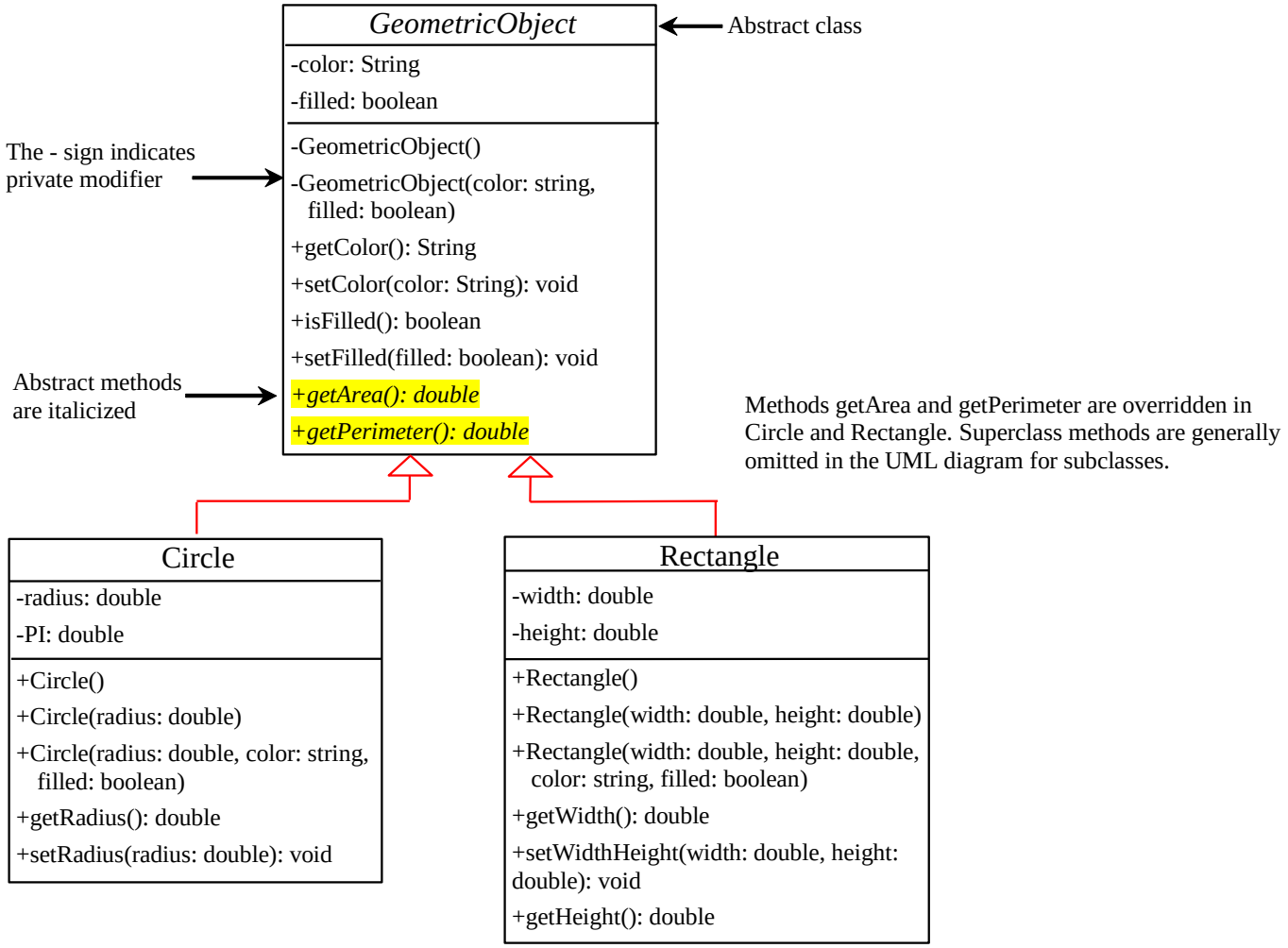
LAB 8

Exercise 1: Create a GeometricObject



Exercise 1: Create a GeometricObject Class

- Create an abstract GeometricObject class that contains:
  - 2 attributes
    - color (String)
    - filled (boolean)
  - 2 constructors:
    - GeometricObject() which set the default to “blue”
    - GeometricObject(String color, boolean filled) which set the color and filled attributes according to the input
  - 4 methods
    - void setColor(String color)
    - void setFilled(boolean filled)
    - String getColor()
    - boolean isFilled()
  - 2 abstract methods
    - double getArea()
    - double getPerimeter()

The GeometricObject class (completed)

```
public abstract class GeometricObject {
  private String color;
  private boolean filled;

  public GeometricObject() {
        this.color = "blue";
  }
  public GeometricObject(String color, boolean filled) {
        this.color = color;
        this.filled = filled;
  }

  public String getColor() {
        return color;
  }
  public void setColor(String color) {
        this.color = color;
  }
  public boolean isFilled() {
        return filled;
  }
  public void setFilled(boolean filled) {
        this.filled = filled;
  }

  public abstract double getArea();

  public abstract double getPerimeter();
}
```

Exercise 1: Create a GeometricObject Class

- Try to remove the keyword `abstract` from the class header
  - Compile
    - Error – a class that contains abstract methods is required to be declared explicitly as an abstract class

Exercise 2: Create a Circle Class

- Create a Circle class that extends the GeometricObject and contains:
  - 2 attributes
    - radius (double)
    - a constant pi (double) from Math class (Math.PI)
  - 3 constructors
    - Circle() \\ call the second constructor with radius 1.0
    - Circle(double radius) \\ call the third constructor with the input radius, color = “white”, and filled = true
    - Circle(double radius, String color, boolean filled) \\ set radius and call a constructor of GeometricObject with the input of color and filled
  - 2 methods
    - void setRadius(double radius)
    - double getRadius() // return radius

- The Circle class

```
public class Circle extends GeometricObject {
  private double radius;
  private double PI = Math.PI;

  public Circle() {
          this(1.0);
  }
  public Circle(double radius) {
          this(radius, "white", true);
  }
  public Circle(double radius, String color, boolean filled) {
          super(color, filled);
          this.radius = radius;
  }

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

}
```

- Save and Compile
  - Error – abstract methods are not yet implemented

- The Circle class
  - Implement 2 abstract methods

```
public class Circle extends GeometricObject {
  private double radius;
  private double PI = Math.PI;

  public Circle() {
          this(1.0);
  }
  public Circle(double radius) {
          this(radius, "white", true);
  }
  public Circle(double radius, String color, boolean filled) {
          super(color, filled);
          this.radius = radius;
  }

  public double getRadius() {
          return radius;
  }
  public void setRadius(double radius) {
          this.radius = radius;
  }
  public double getArea() {
          return radius*radius*PI;

  }
  public double getPerimeter() {
          return 2*radius*PI;
  }
}
```

Exercise 3: Create a Rectangle Class

- Create a Rectangle class that extends the GeometricObject and contains:
  - 2 attributes
    - width (double)
    - height (double)
  - 3 constructors
    - Rectangle() \\ call the second constructor with width = 1.0 and height = 1.0
    - Rectangle(double width, double height) \\ call the third constructor with the input radius, color = “green”, and filled = true
    - Rectangle(double width, double height, String color, boolean filled) \\ set width, height and call a constructor of GeometricObject with the input of color and filled
  - 3 methods
    - void setWidthHeight(double width, double height)
    - double getWidth() // return width
    - double getHeight() // return height

- The Rectangle class

```
public class Rectangle extends GeometricObject{
  private double width;
  private double height;

  public Rectangle() {
          this(1.0, 1.0);
  }
  public Rectangle(double width, double height) {
          this(width, height,"green", true);
  }
  public Rectangle(double width, double height, String color, boolean filled) {
          super(color, filled);
          this.width = width;
          this.height = height;
  }

  public double getWidth() {
          return width;
  }
  public double getHeight() {
          return height;
  }
  public void setWidthHeight(double width, double height) {
          this.width = width; this.height = height;
  }
}
```

- Compile and fix errors

- The Rectangle class (Fixed)

```
public class Rectangle extends GeometricObject {
     …
          public double getWidth() {
                    return width;
          }
          public double getHeight() {
                    return height;
          }
          public void setWidthHeight(double width, double height) {
                    this.width = width;
                    this.height = height;
          }
          public double getArea() {
                    return width*height;
          }
          public double getPerimeter() {
                    return (2*width)+(2*height);
          }
}
```

Exercise 4: Abstraction

- Create an AbstractionTester class
  - add a main() method

```
public class AbstractionTester {
	public static void main(String args[]){

	}
}
```

Exercise 4: Abstraction

- Create an object of GeometricObject class: g1

```
public class AbstractionTester {
	public static void main(String args[]){
		GeometricObject g1 = new GeometricObject();
	}
}
```

- Compile
  - Error – the abstract class cannot be initiated as an object

Exercise 4: Abstraction

- Create a Circle object: c1 and try to call different methods

```
public class AbstractionTester {
	public static void main(String args[]){
		Circle c1 = new Circle();
		c1.setRadius(5.0);
		c1.setColor("green");
		c1.setFilled(false);
		System.out.println(c1.getRadius());
		System.out.println(c1.getColor());
		System.out.println(c1.getArea());
		System.out.println(c1.getPerimeter());
	}
}
```

Exercise 4: Abstraction

- Create a Rectangle object: r1 and try to call different methods

```
public class AbstractionTester {
	public static void main(String args[]){
		Rectangle r1 = new Rectangle();
		r1.setWidthHeight(5, 10);
		r1.setColor("yellow");
		r1.setFilled(true);
		System.out.println(r1.getWidth());
		System.out.println(r1.getHeight());
		System.out.println(r1.getColor());
		System.out.println(r1.getArea());
		System.out.println(r1.getPerimeter());
	}
}
```

- Try to use the method setRadius(2.5) and getRadius()

Exercise 5: Create an Interface Class

- Create a new class of GeometricObjectInterface that contains
  - 1 attribute
    - color (String)
  - 2 abstract methods
    - double getArea()
    - double getPerimeter()

Exercise 5: Create an Interface Class

```
public interface GeometricObjectInterface {
      public static final String color = "white";
      public  double getArea();
      public  double getPerimeter();
}
```

- Try
  - add the keyword `abstract` in front of method name
    - Work fine – with or without the `abstract` keyword the methods are abstract anyway
  - add variable String color (comment the first statement out first)
    - Error – an interface class does not allow to have any variable
    - Only a constant with `public static final` modifier is allowed

Exercise 5: Create a Circle2 Class

- Create a Circle2 class that implements the GeometricObjectInterface and contains:
  - 2 attributes
    - radius (double)
    - a constant pi (double) from Math class (Math.PI)
  - 2 constructors
    - Circle2() \\ call the second constructor with radius 1.0
    - Circle2(double radius) \\ set radius
  - 4 methods
    - void setRadius(double radius)
    - double getRadius() // return radius
    - double getArea()
    - double getPerimeter()

Exercise 5: Create a Circle2 Class

```
public class Circle2 implements GeometricObjectInterface {
  private double radius;
  private final double PI = Math.PI;
  public Circle2() {
        this(1.0);
  }
  public Circle2(double radius) {
        this.radius = radius;
  }

  public void setRadius(double radius){
        this.radius = radius;
  }
  public double getRadius() {
        return radius;
  }
  public double getArea(){
        return radius*radius*Math.PI;
  }
  public double getPerimeter(){
        return 2*radius*Math.PI;
  }
}
```

Exercise 6: Interface

- Create an InterfaceTester class
  - add a main() method
  - create an object of Circle2: c2
  - try
    - call methods of c2
    - change color of c2 – cannot change, why?

Exercise 6: Interface

```
public class InterfaceTester {
  public static void main(String args[]){
       System.out.println(Circle2.color);
       Circle2.color = "green";
       Circle2 c2 = new Circle2();
       System.out.println(c2.color);
       c2.setRadius(3);
       System.out.println(c2.getRadius());
       System.out.println(c2.getArea());
       System.out.println(c2.getPerimeter());
  }
}
```