SWEN20003

Workshop 3, Week 4 (2: Electric Boogaloo)

Eleanor McMurtry, University of Melbourne

Information Hiding

What is information hiding?

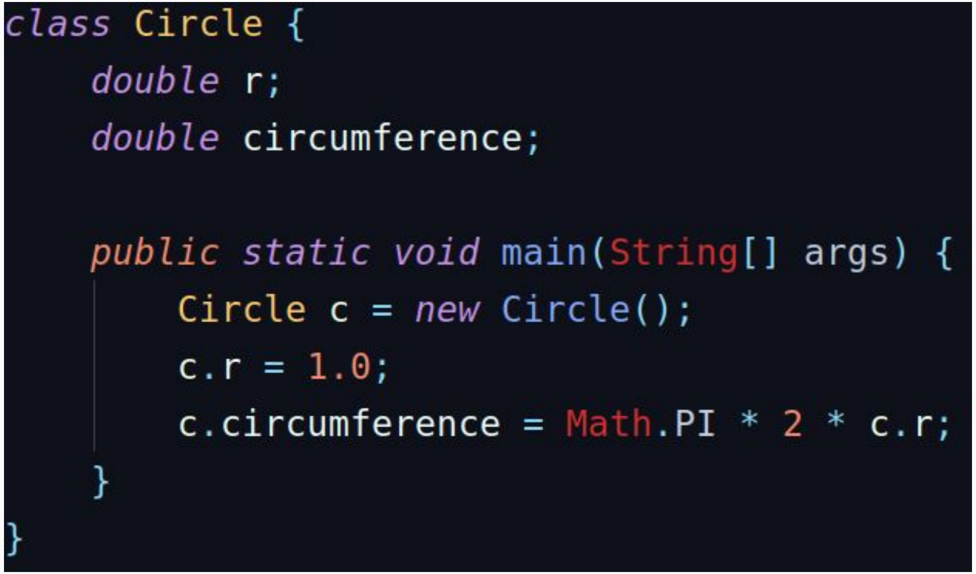
- Make methods and attributes only visible *inside this class*
- No other class can see or use them

How to do information hiding

How to do information hiding

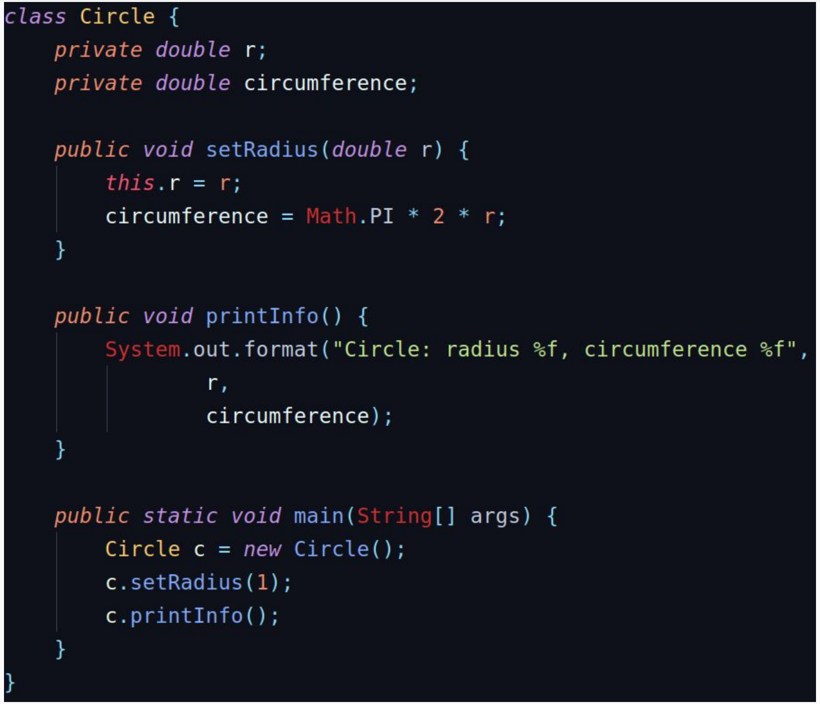
- Visibility modifiers: `public`, `private`, ...

Compare...



```
class Circle {
    double r;
    double circumference;

    public static void main(String[] args) {
        Circle c = new Circle();
        c.r = 1.0;
        c.circumference = Math.PI * 2 * c.r;
    }
}
```



```
class Circle {
    private double r;
    private double circumference;

    public void setRadius(double r) {
        this.r = r;
        circumference = Math.PI * 2 * r;
    }

    public void printInfo() {
        System.out.format("Circle: radius %f, circumference %f",
                r,
                circumference);
    }

    public static void main(String[] args) {
        Circle c = new Circle();
        c.setRadius(1);
        c.printInfo();
    }
}
```

Every (non-final) attribute should be private.

(unless you’re really, *really* sure you know what you’re doing)

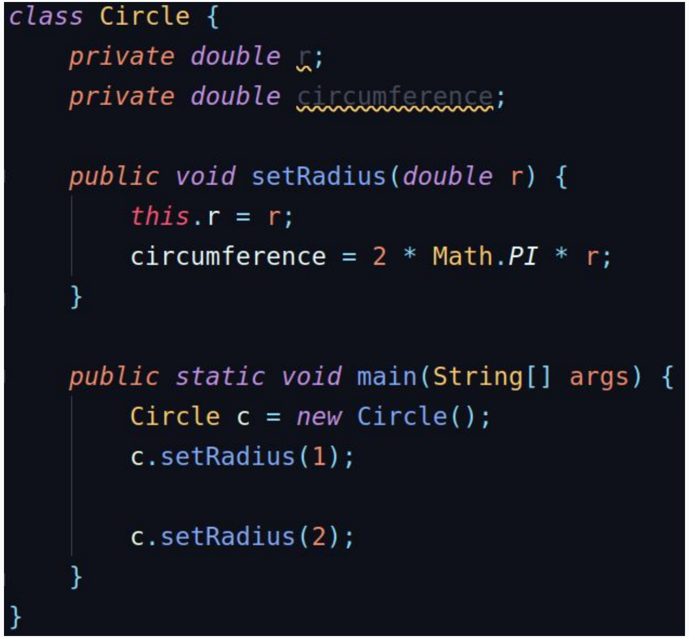
Immutability

Immutability

- Attributes cannot change after creating instance

Compare...

Mutable



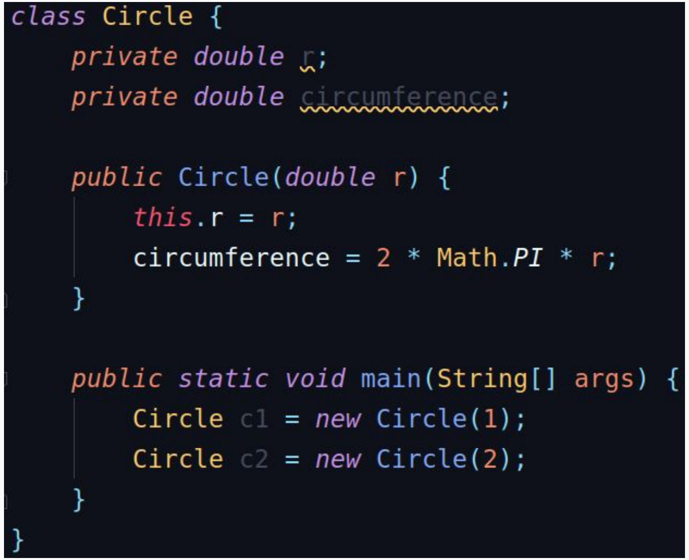
```java
class Circle {
    private double r;
    private double circumference;

    public void setRadius(double r) {
        this.r = r;
        circumference = 2 * Math.PI * r;
    }

    public static void main(String[] args) {
        Circle c = new Circle();
        c.setRadius(1);

        c.setRadius(2);
    }
}
```

Immutable



```java
class Circle {
    private double r;
    private double circumference;

    public Circle(double r) {
        this.r = r;
        circumference = 2 * Math.PI * r;
    }

    public static void main(String[] args) {
        Circle c1 = new Circle(1);
        Circle c2 = new Circle(2);
    }
}
```

Immutability

- Attributes cannot change after creating instance

- **Not the same as `final`**

Arrays

Arrays

- Homogeneous fixed-length list
- `int[] nums = new int {0, 1, 2, 3}`

Arrays

- Homogeneous fixed-length list
- `int[] nums = new int[] {0, 1, 2, 3}`

- How many instances of Circle do we create with:

```
Circle[] circles = new Circle[10]
```

Array operations

- `circles.length`

Array operations

- `circles.length`
- `Arrays.toString(circles)`

Array operations

- `circles.length`
- `Arrays.toString(circles)`
- `Arrays.sort(nums)`

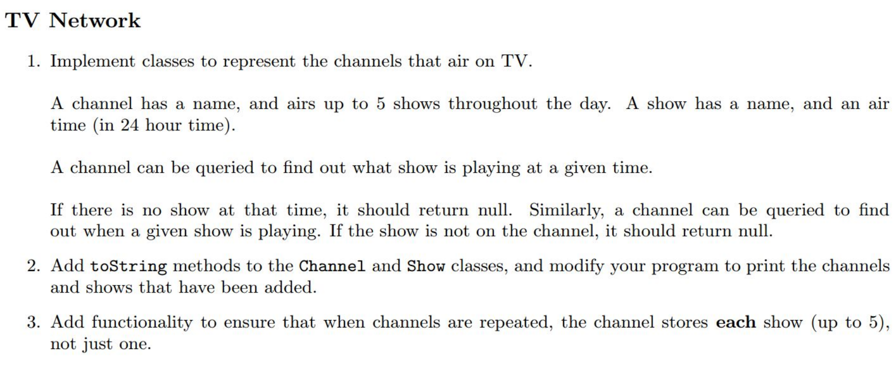
Array operations

- `circles.length`
- `Arrays.toString(circles)`
- `Arrays.sort(nums)`
- `Arrays.copyOf(nums)`

Array operations

- `circles.length`
- `Arrays.toString(circles)`
- `Arrays.sort(nums)`
- `Arrays.copyOf(nums)`
- `Arrays.equals(a, b)`

Worked Problem



TV Network

1. Implement classes to represent the channels that air on TV.

   A channel has a name, and airs up to 5 shows throughout the day. A show has a name, and an air time (in 24 hour time).

   A channel can be queried to find out what show is playing at a given time.

   If there is no show at that time, it should return null. Similarly, a channel can be queried to find out when a given show is playing. If the show is not on the channel, it should return null.

2. Add `toString` methods to the `Channel` and `Show` classes, and modify your program to print the channels and shows that have been added.

3. Add functionality to ensure that when channels are repeated, the channel stores **each** show (up to 5), not just one.