OCP

OPEN/CLOSED PRINCIPLE

Is our design ready to be easily extended?

The Open/Closed Principle (OCP)

“

Software entities (classes, modules, functions, etc.) should be open for extension but closed for modification

— *Bertrand Meyer*

”

wrong

```
public class Shape {}

class Square extends Shape {}

class Circle extends Shape {}
```

```
public class Painter {

  void paint (Collection<Shape> shapes) {
    for (Shape shape: shapes) {
      if (shape instanceof Square) {
        paint((Square) shape);
      } else if (shape instanceof Circle) {
        paint((Circle) shape);
      }
    }
  }

  void paint (Square square) {
    // ...
  }

  void paint (Circle circle) {
    // ...
  }
}
```

```
interface Shape {
  void paint();
}

class Square implements Shape {

  @Override
  public void paint() {
    // ...
  }
}

class Circle implements Shape {

  @Override
  public void paint() {
    // ...
  }
}
```

right

```
class Painter {

  void paint (Collection<Shape> shapes) {
    for (Shape shape: shapes) {
      shape.paint();
    }
  }
}
```

OCP

OPEN/CLOSED PRINCIPLE