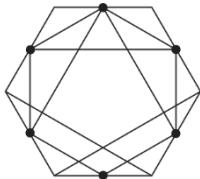
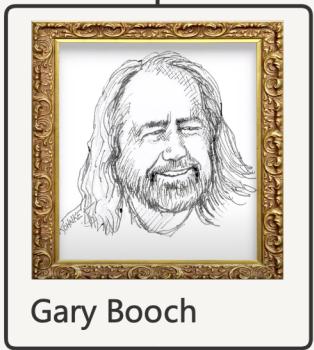


LISKOV SUBSTITUTION PRINCIPLE



Clean Code

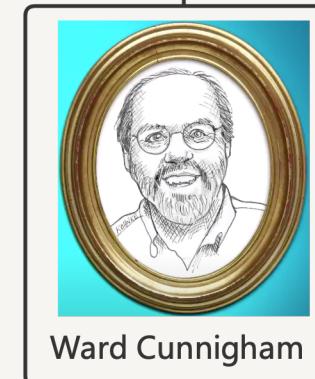


Simple & Direct

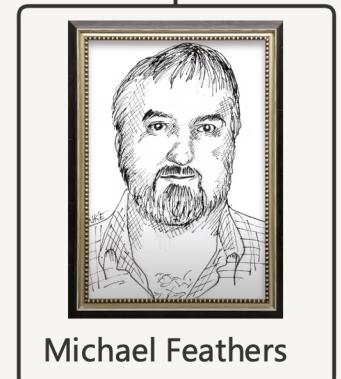
Reads as well-written prose



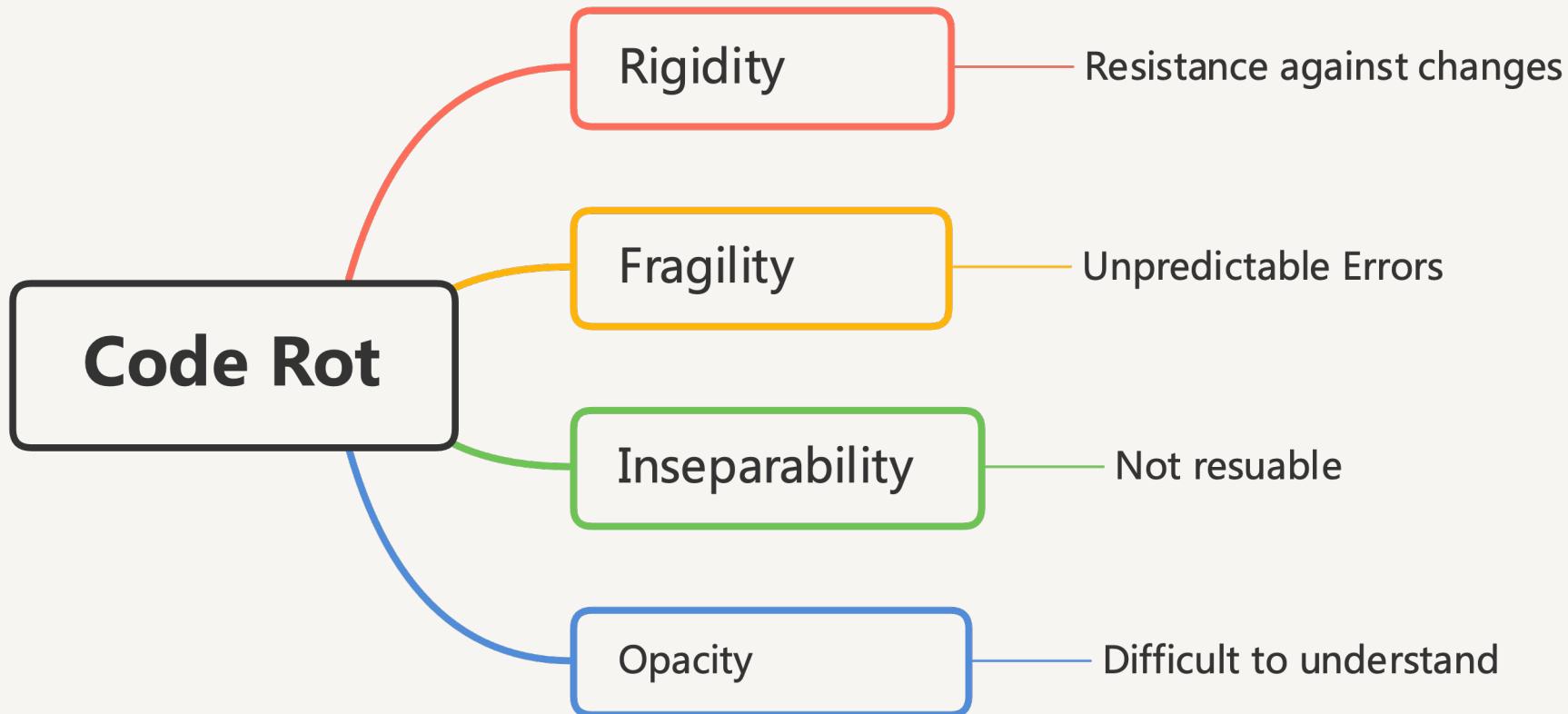
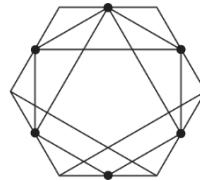
Elegant & Efficient. Should do one thing

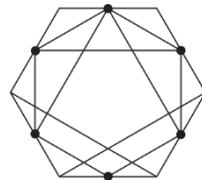


Every Routine is pretty
much what you expected



Written by someone who cares





SOLID

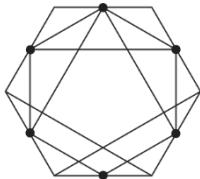
Single Responsibility Principle

Open Close Principle

Liskov Substitution Principle

Interface Segregation Principle

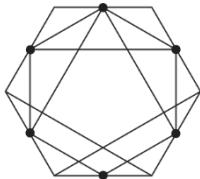
Dependency Inversion Principle



LISKOV SUBSTITUTION PRINCIPLE



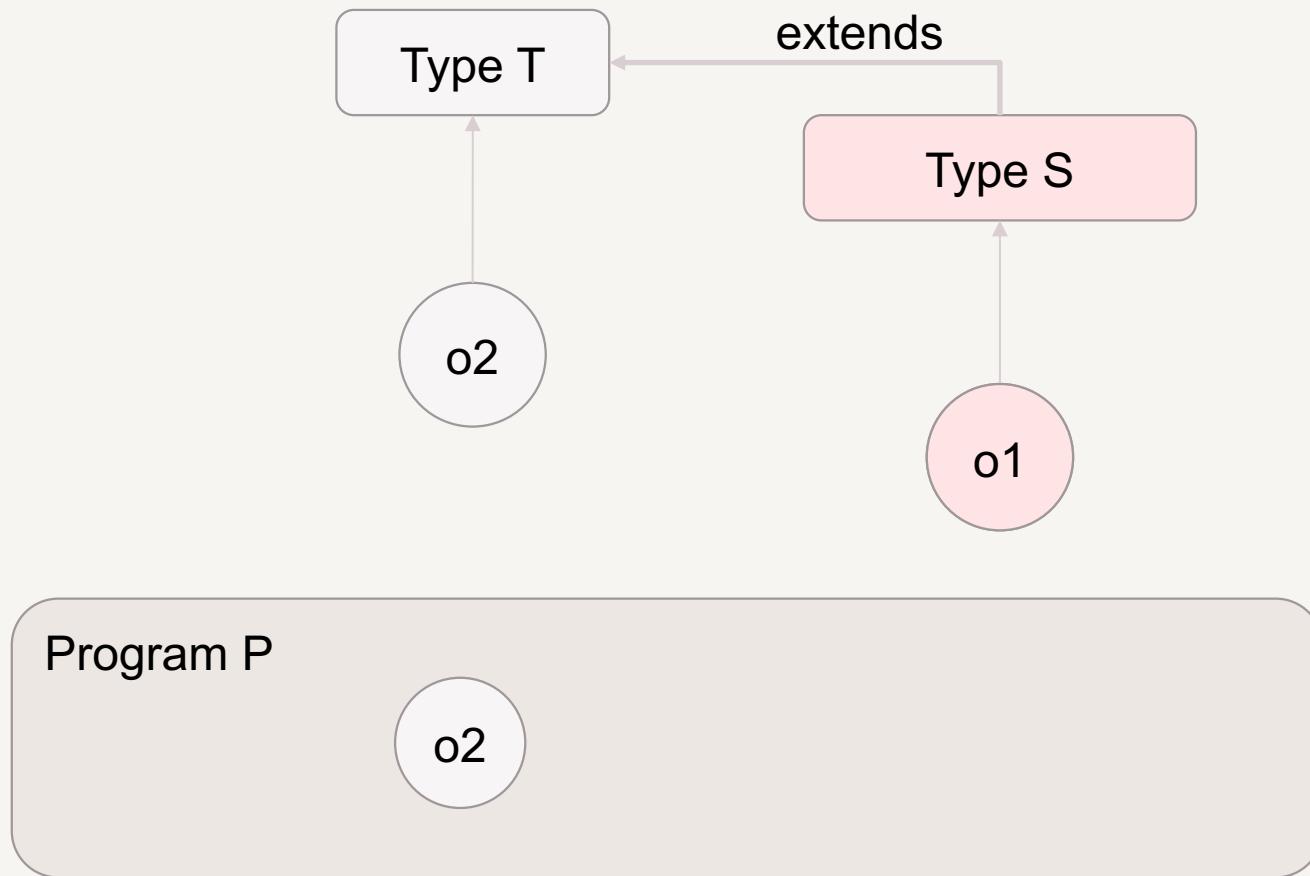
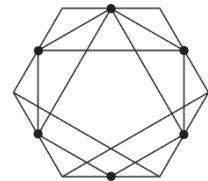
- Instances of subclasses
- Can take the role of superclasses
- Without changing program behavior

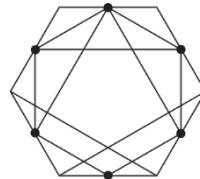


LISKOV SUBSTITUTION PRINCIPLE



- More relaxed version
- Without causing problems
- This is a fundamental feature of class hierarchies

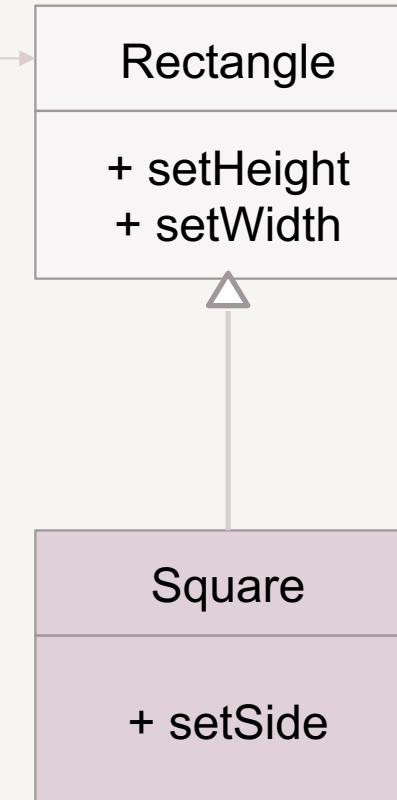


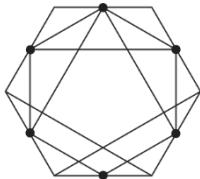


LSP VIOLATION

User

```
Rectangle rectangle = new Square()  
rectangle.setHeight(10);  
rectangle.setWidth(5);  
  
assert(rectangle.area() == 50);
```

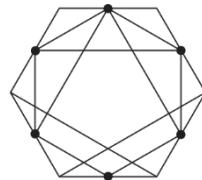




LSP VIOLATION

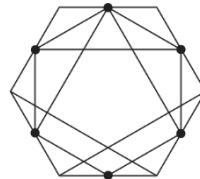
```
3 ⚡ public class Rectangle {  
4  
5     private double height;  
6     private double width;  
7  
8     ⚡ public void setHeight(double height) {...}  
13    ⚡ public void setWidth(double width) {...}  
18  
19  
20     + public double area() { return height * width; }  
23 }
```

```
3  
4  
5  
6  
7     + public class Square extends Rectangle {  
8  
9         private double side;  
10  
11         public void setSide(double side){  
12             this.side = side;  
13             super.setHeight(side);  
14             super.setWidth(side);  
15         }  
16  
17         @Override  
18         public void setHeight(double side){  
19             super.setHeight(side);  
20             super.setWidth(side);  
21         }  
22  
23         @Override  
24         public void setWidth(double side){  
25             super.setHeight(side);  
26             super.setWidth(side);  
27     }
```



LSP VIOLATION

```
3 ► public class Main {  
4  
5 ►   □ public static void main(String[] args) {  
6  
7     Rectangle rectangle = new Rectangle();  
8     rectangle.setHeight(10);  
9     rectangle.setWidth(5);  
10    System.out.println(rectangle.area()); // Liefert 50.  
11  
12    rectangle = new Square();  
13    rectangle.setHeight(10);  
14    rectangle.setWidth(5);  
15    System.out.println(rectangle.area()); // Liefert 25  
16  
17  }
```



ÖSUNGEN UND DISKUSSIONEN





VIELEN DANK FÜR IHRE AUFMERKSAMKEIT.