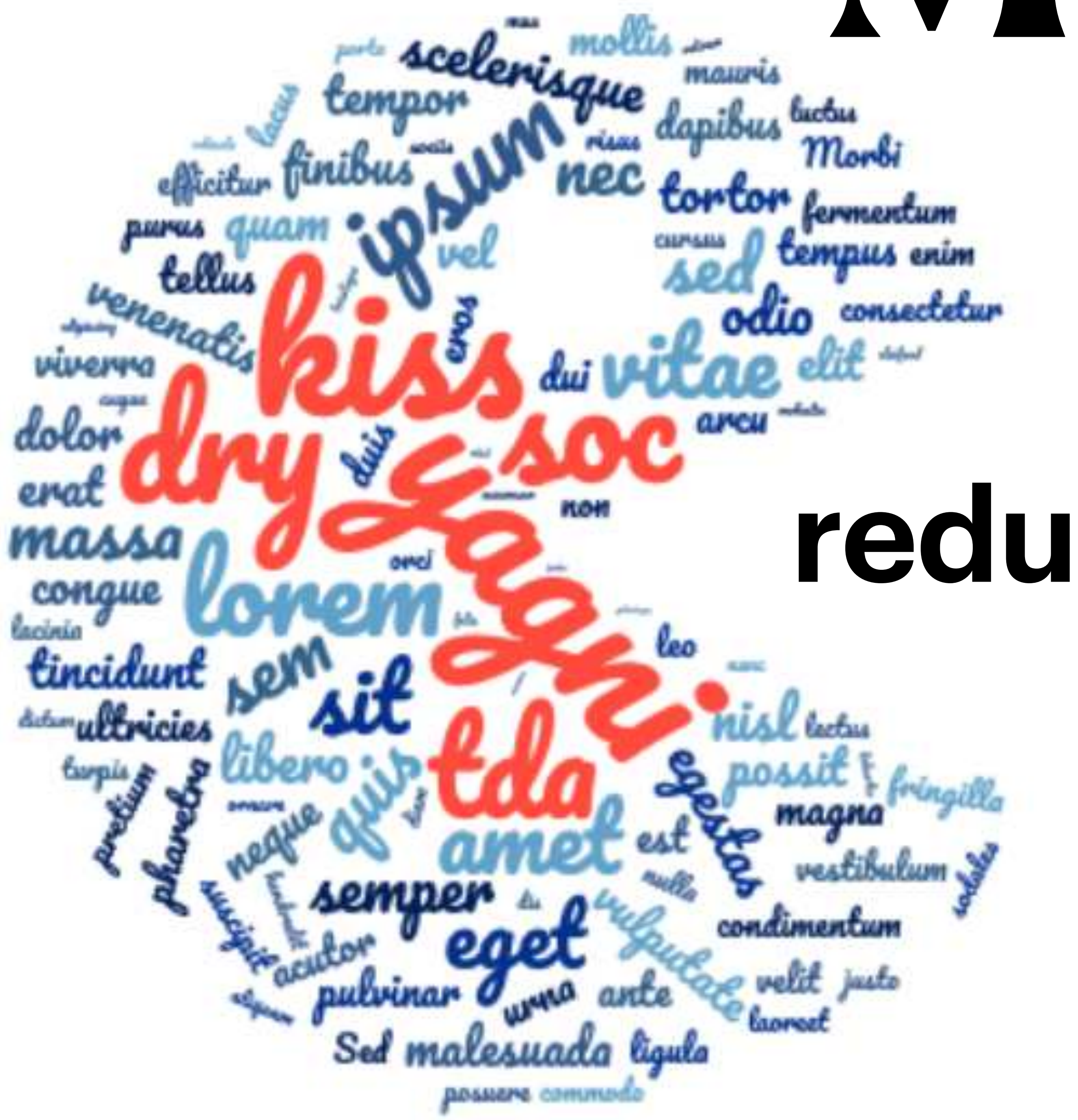
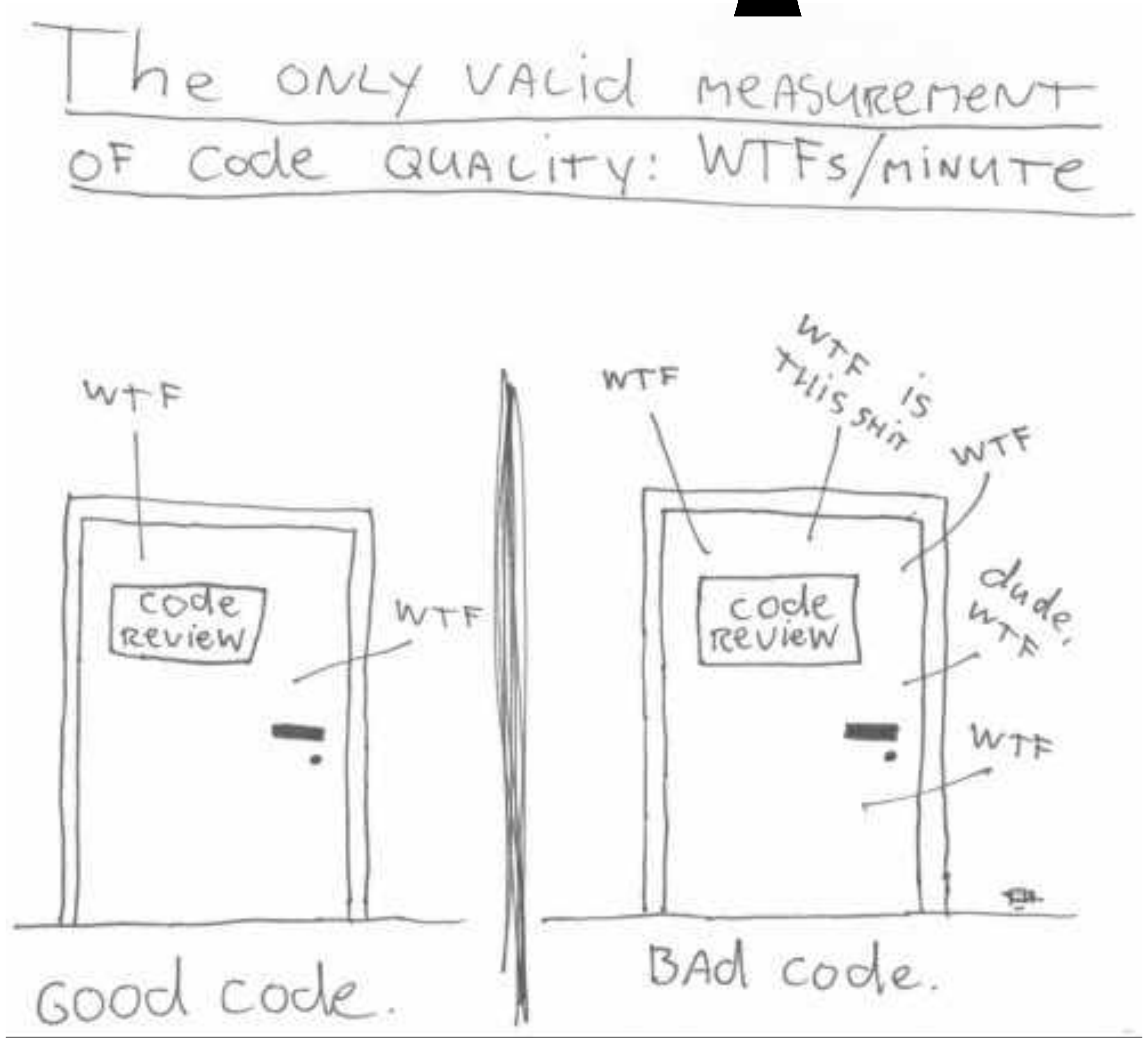
My Way to



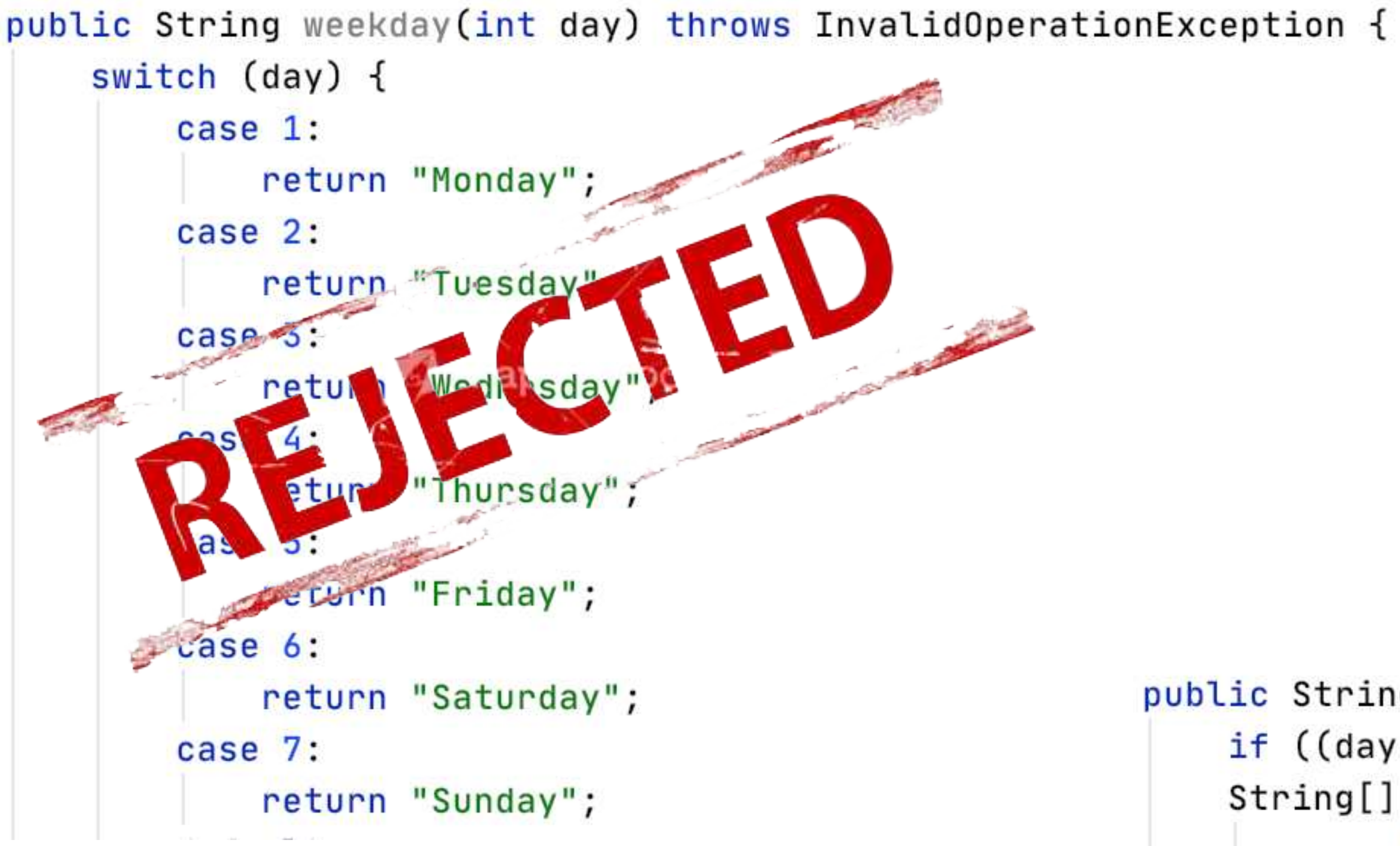
reduce the WTFpm

WTFpm


The ONLY VALid MEASUREMENT
OF Code QUALITY: WTFs/MINUTE
WTF
code review
WTF
Good code.
WTF
WTF is THIS SHIT
WTF
code review
dude, WTF
WTF
BAd code.

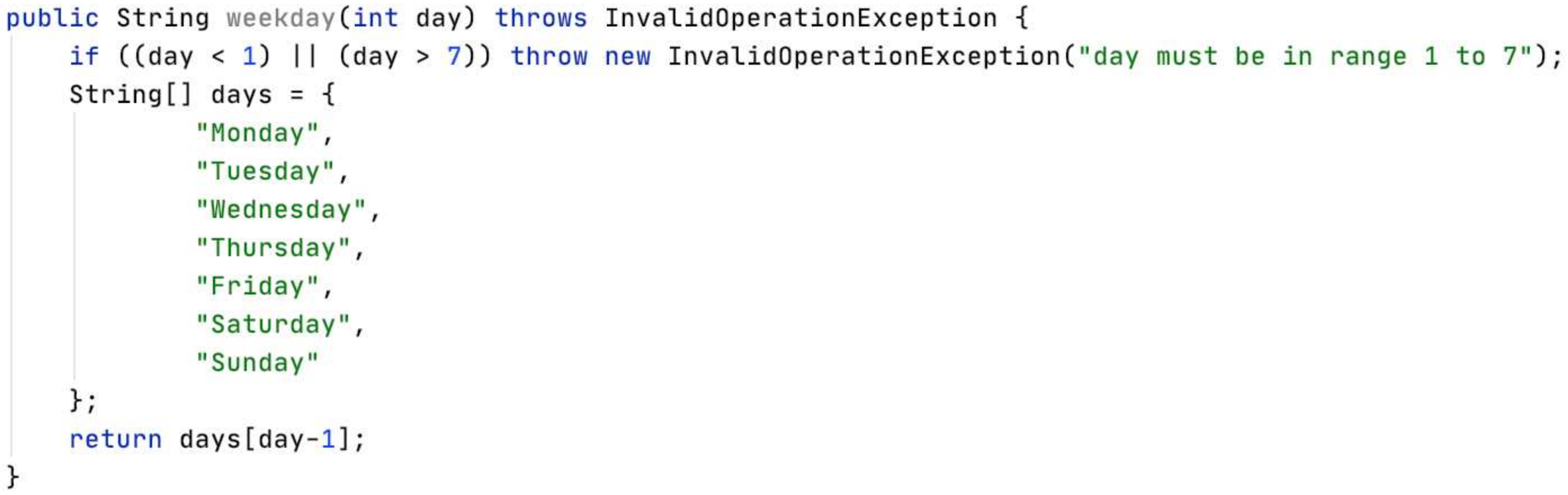
KISS

Keep it simple stupid



```
public String weekday(int day) throws InvalidOperationException {
    switch (day) {
        case 1:
            return "Monday";
        case 2:
            return "Tuesday";
        case 3:
            return "Wednesday";
        case 4:
            return "Thursday";
        case 5:
            return "Friday";
        case 6:
            return "Saturday";
        case 7:
            return "Sunday";
```

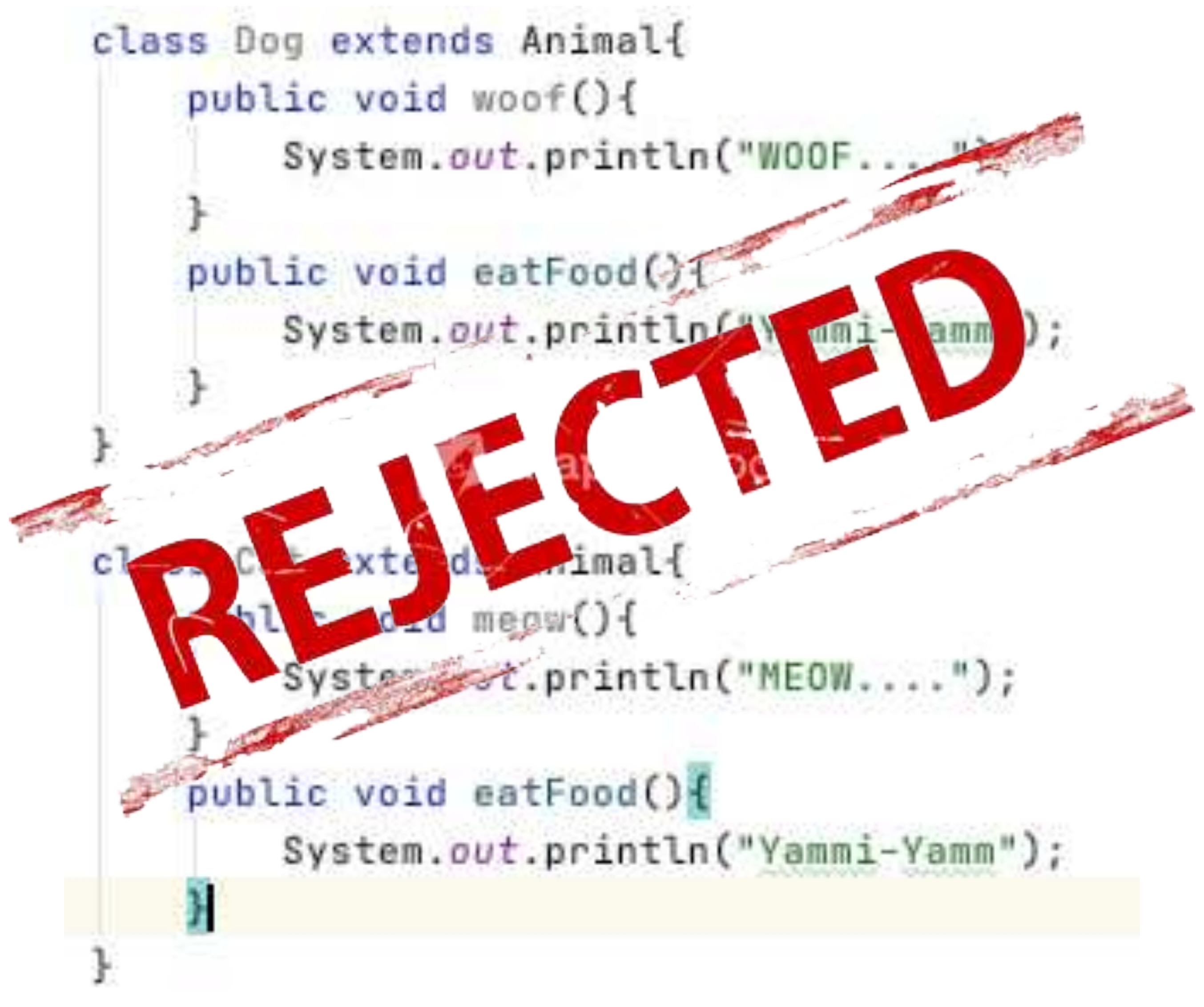
REJECTED



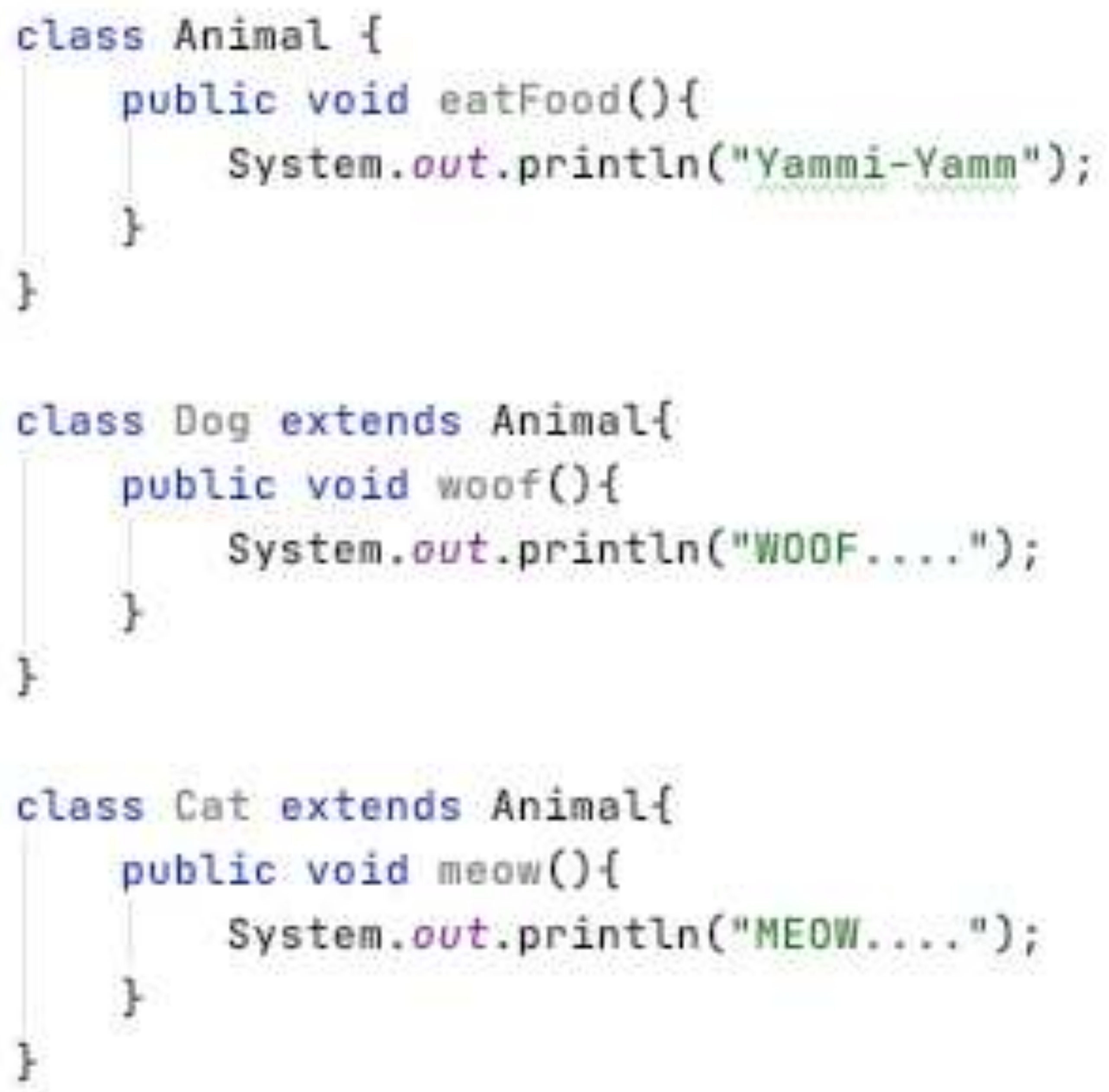
```
public String weekday(int day) throws InvalidOperationException {
    if ((day < 1) || (day > 7)) throw new InvalidOperationException("day must be in range 1 to 7");
    String[] days = {
            "Monday",
            "Tuesday",
            "Wednesday",
            "Thursday",
            "Friday",
            "Saturday",
            "Sunday"
    };
    return days[day-1];
}
```

DRY

Do not repeat yourself



```
class Dog extends Animal{
    public void woof(){
        System.out.println("WOOF....");
    }
    public void eatFood(){
        System.out.println("Yammi-Yamm");
    }
}

class Cat extends Animal{
    public void meow(){
        System.out.println("MEOW....");
    }
    public void eatFood(){
        System.out.println("Yammi-Yamm");
    }
}
```



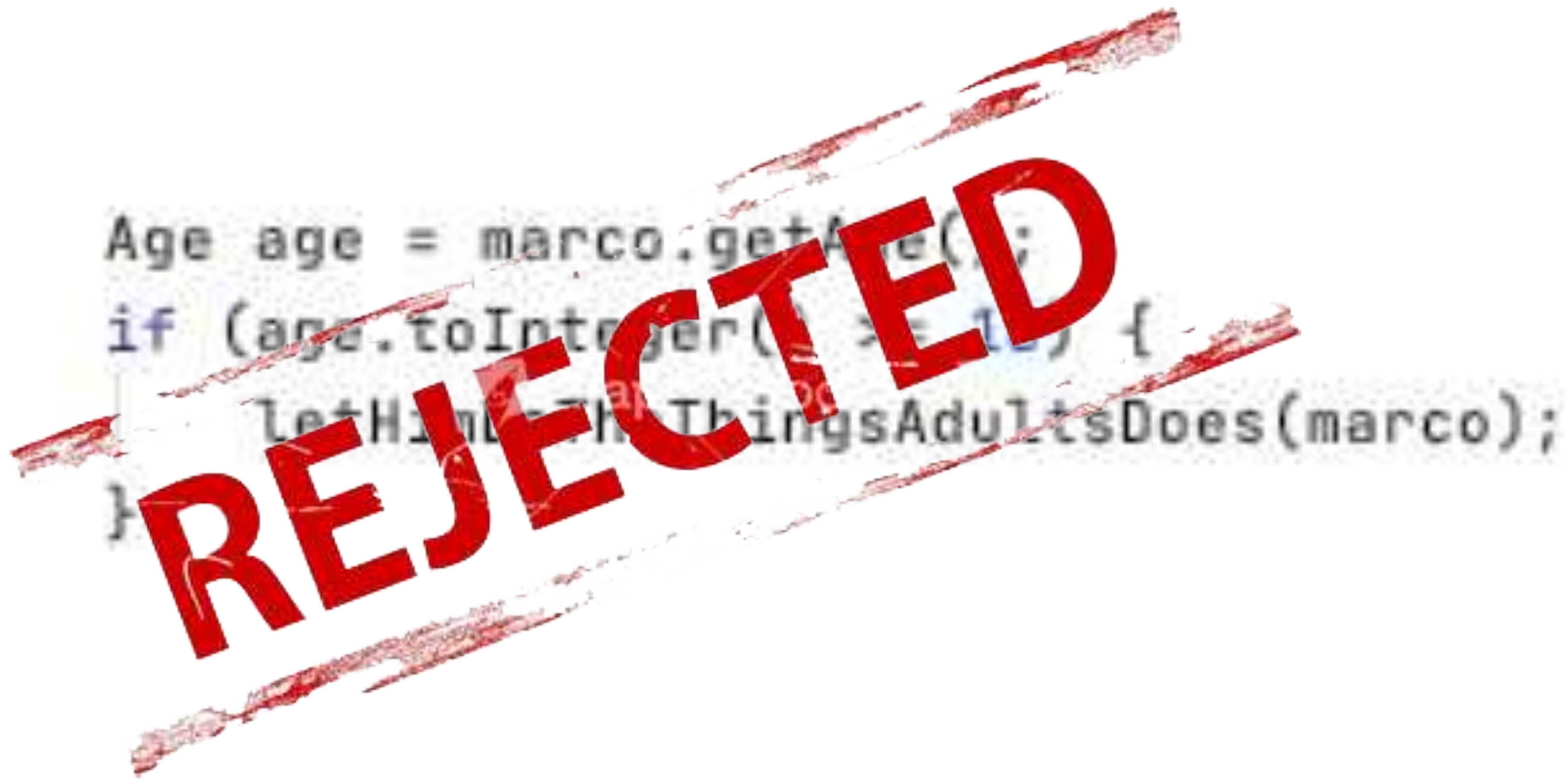
```
class Animal {
    public void eatFood(){
        System.out.println("Yammi-Yamm");
    }
}

class Dog extends Animal{
    public void woof(){
        System.out.println("WOOF....");
    }
}

class Cat extends Animal{
    public void meow(){
        System.out.println("MEOW....");
    }
}
```

WET solution

- We enjoy typing
- Waste everyone’s time
- Write every time
- Write everything twice

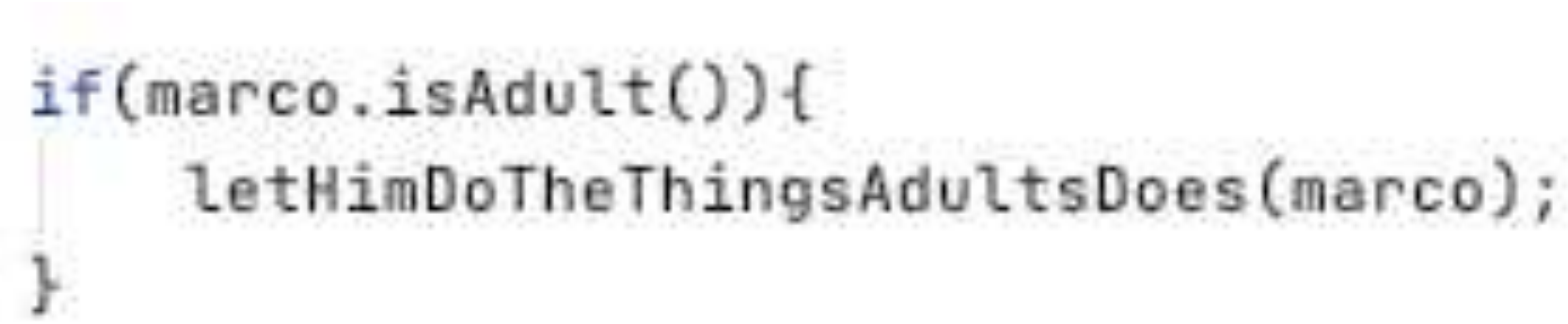
TDA

Tell don’t ask



```
Age age = marco.getAge();
if (age.toInteger() >= 18) {
    letHimDoTheThingsAdultsDoes(marco);
}
```

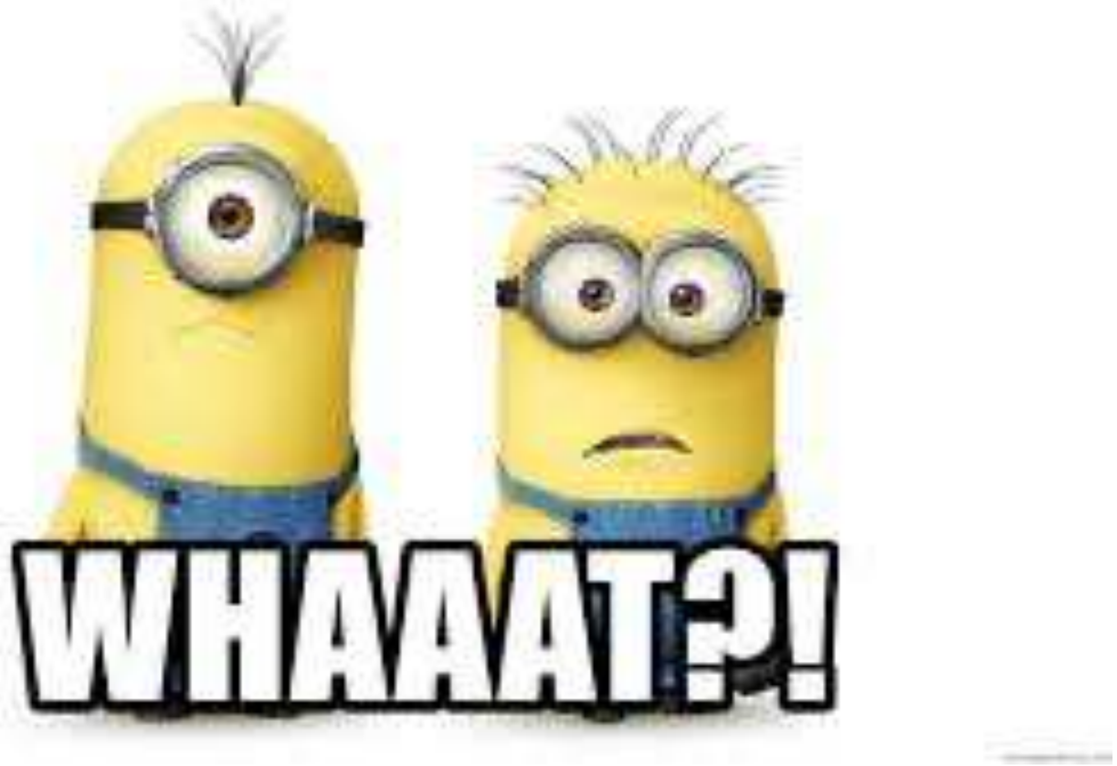
REJECTED



```
if(marco.isAdult()){
    letHimDoTheThingsAdultsDoes(marco);
}
```

SoC / SRP

Separation of concerns

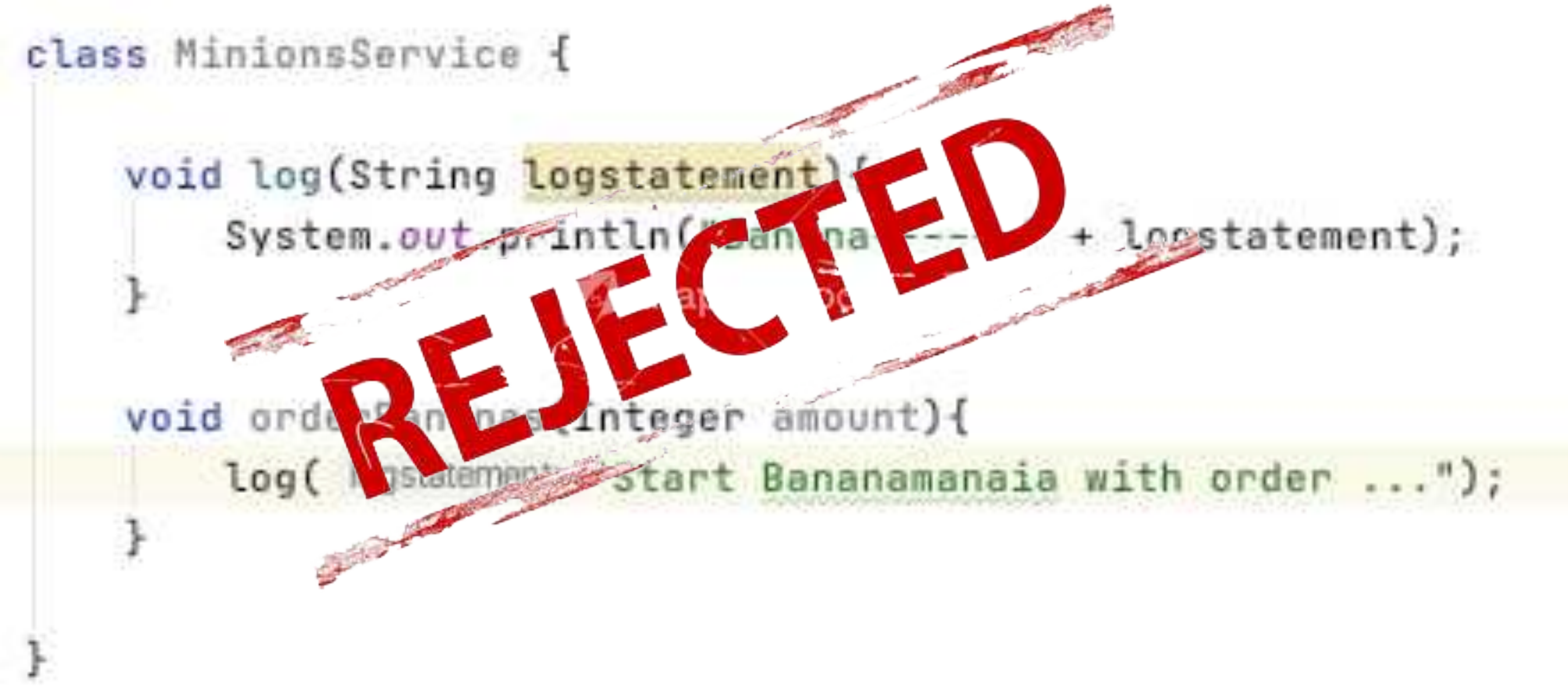


SRP

each object

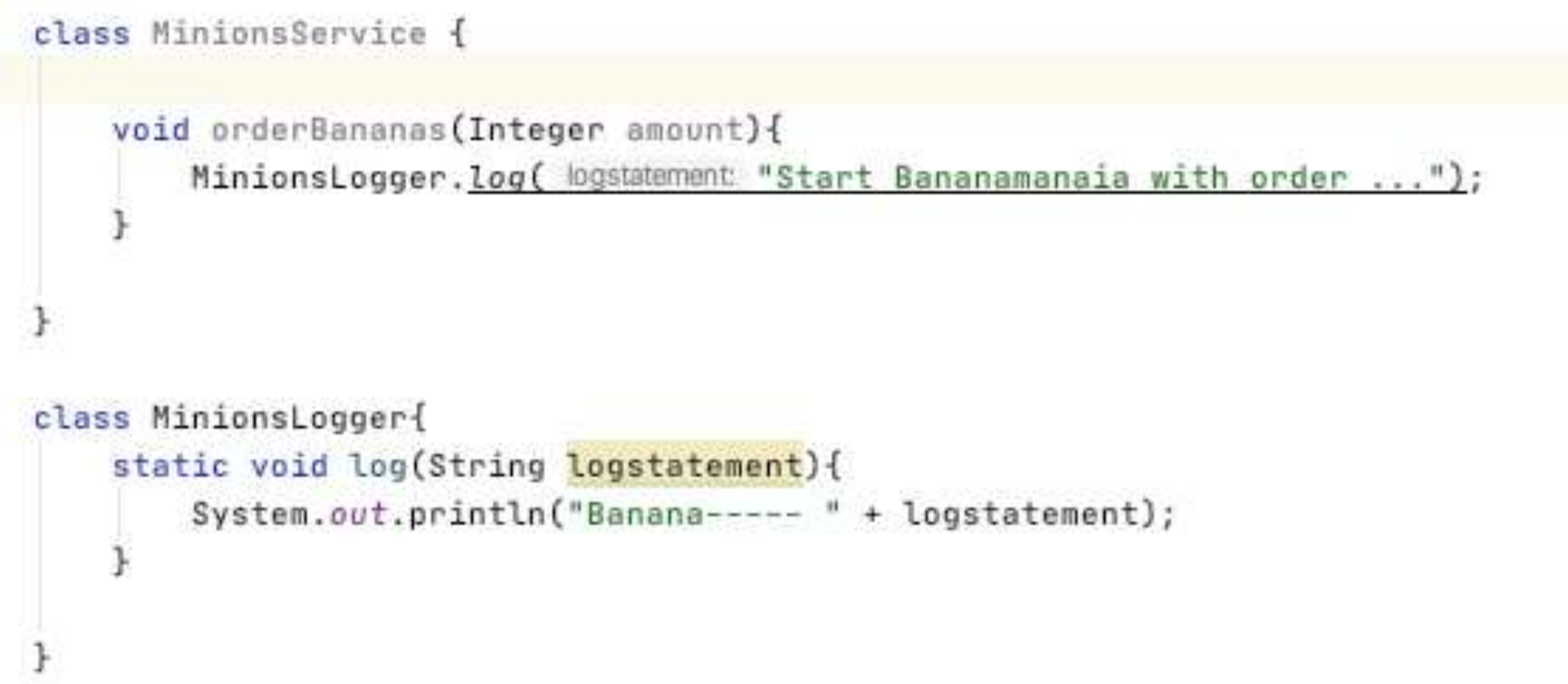
SoC

application level



REJECTED

```
class MinionsService {

    void log(String logstatement){
        System.out.println("Banana----- " + logstatement);
    }

    void orderBananas(Integer amount){
        log( logstatement: "Start Bananamanaia with order ...");
    }

}
```



```
class MinionsService {

    void orderBananas(Integer amount){
        MinionsLogger.log( logstatement: "Start Bananamanaia with order ...");
    }

}

class MinionsLogger{
    static void log(String logstatement){
        System.out.println("Banana----- " + logstatement);
    }

}
```

YAGNI

you ain't gonna need it

I bought a ...

just in case we need it

or we will eventually need it

Wrong! We didn't need it, we don't need it ...

in most of the cases

CONCLUSION

Try to follow the rules ....

less unicorn features

better readability

focus on most important things

maintainability

save nerves

better code quality

Don’t believe

measure by yourself

visit: https://github.com/AleCie/WTFpm