

# Identifying code smells

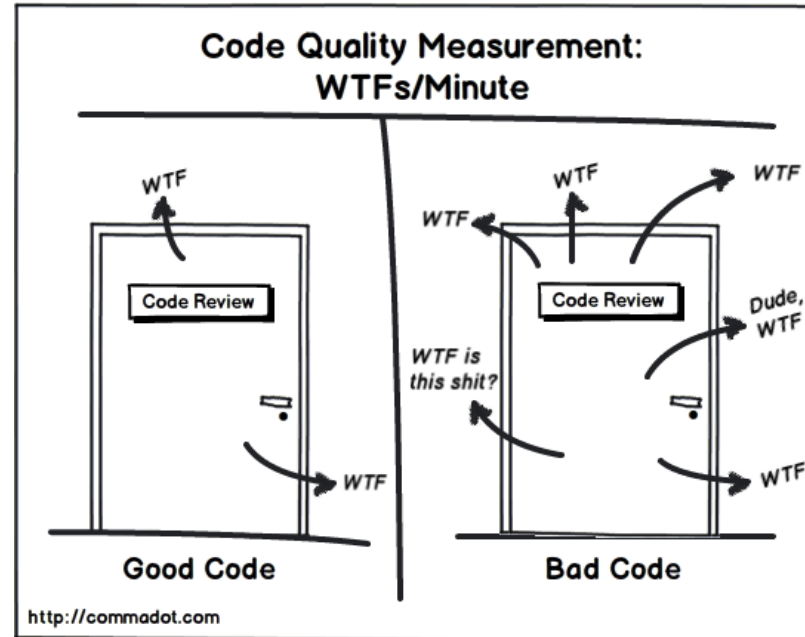


# Agenda

- ▶ What are code smells?
- ▶ Types of code smells
- ▶ Identifying code smells
- ▶ Conclusion

# What are code smells?

- ▶ Characteristic in code that possibly indicates a deeper problem
- ▶ Usually not bugs
- ▶ Can make the code harder to maintain



# Types of code smells

Bloaters

Object-  
Orientation  
Abusers

Change  
Preventers

Dispensables

Couplers

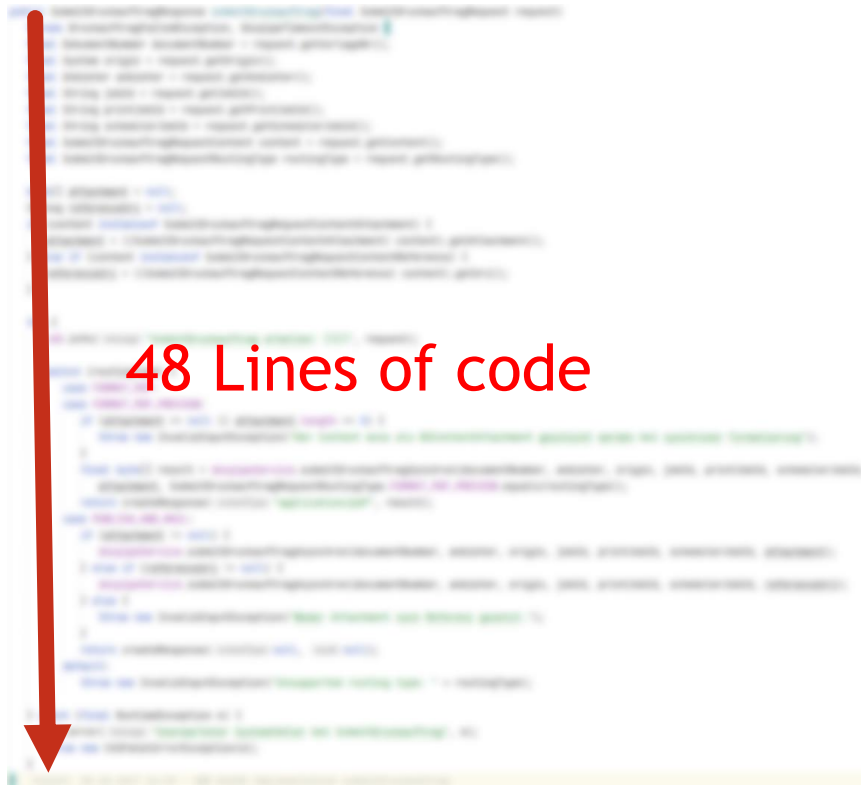
# Identifiying code smells

`public void foo(int a, int b, int c, int d, int e, int f, int g, int h, int i, int j, int k, int l, int m, int n, int o, int p, int q, int r, int s, int t, int u, int v, int w, int x, int y, int z) {  
 // ...  
}`

Bloater

Long parameter list

# Identifying code smells



Bloater

Long Method

Large Class

# Identifying code smells

```
public void add(String name, int age) {
    if (name == null || name.length() == 0) {
        throw new IllegalArgumentException("Name cannot be null or empty");
    }
    if (age < 0 || age > 120) {
        throw new IllegalArgumentException("Age must be between 0 and 120");
    }
    // ... more code ...
}

// ... more code ...
}
```

Couplers

Feature envy

# Identifying code smells

```
1 // This method calculates the perimeter of a rectangle. It takes two double values as arguments: width and height.
2 // The perimeter is calculated as 2 * (width + height).
3
4 public double calculatePerimeter(double width, double height) {
5     return 2 * (width + height);
6 }
7
8 // This method calculates the area of a rectangle. It takes two double values as arguments: width and height.
9 // The area is calculated as width * height.
10
11 public double calculateArea(double width, double height) {
12     return width * height;
13 }
```

Dispensables

Comment



# Identifying code smells

```
public class HelloWorldApplication {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

```
public class HelloWorldApplication {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

```
public class HelloWorldApplication {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

```
public class HelloWorldApplication {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

```
public class HelloWorldApplication {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

Change  
Preventers

Shotgun Surgery

# Identifying code smells

```
enum class E { ... };
void foo() {
    switch (E::A) {
        case E::A:
            // ...
            break;
        case E::B:
            // ...
            break;
        case E::C:
            // ...
            break;
    }
}
```

```
enum class E { ... };
void foo() {
    switch (E::A) {
        case E::A:
            // ...
            break;
        case E::B:
            // ...
            break;
        case E::C:
            // ...
            break;
    }
}
```

Object-  
Orientation  
Abuser

Switch Statement

# Identifying code smells



Dispensables

Lazy Class

# Identifying code smells

```
class User {
  String name;
  String email;
  String phone;

  User(String name, String email, String phone) {
    this.name = name;
    this.email = email;
    this.phone = phone;
  }

  void setName(String name) {
    this.name = name;
  }

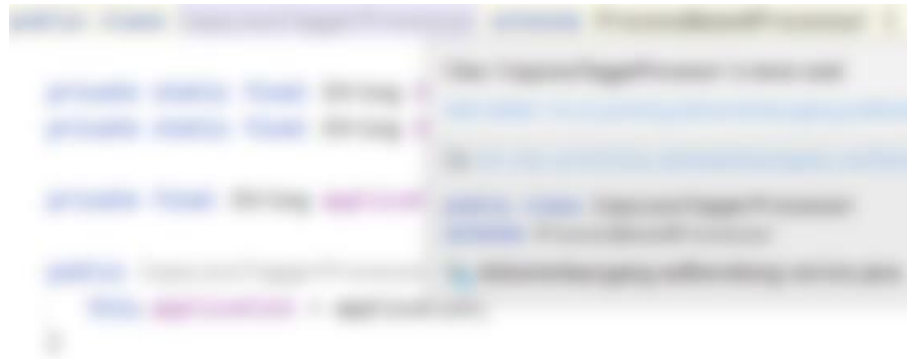
  void setEmail(String email) {
    this.email = email;
  }

  void setPhone(String phone) {
    this.phone = phone;
  }
}
```

Dispensables

Data Class

# Identifying code smells



Dispensables

Speculative Generality

Dead Code

# Identifying code smells

Additional mention

A blurred image of a code snippet, likely representing a code smell. The text is illegible due to blurring, but it appears to be a list of items or a table with multiple rows and columns.

Uncommunicative Name

# Conclusion

- ▶ Knowing code smells helps finding them
- ▶ Not all code smells are easy to spot
- ▶ Not all signs of code smells are necessarily signs of bad code
- ▶ Refactor code smells when you identify them

# Thank you for your attention!

## Any questions?





# Sources

- ▶ <https://pragmaticways.com/31-code-smells-you-must-know/>
- ▶ <https://refactoring.guru/refactoring/smells>
- ▶ Alcor Academy Lesson Code Smells