



Detecting Code Smells

4 Tiny Tasks

Simon Hodel, 03.09.2020

Recap: Code Smells Overview

→ Code Smells

Bloaters

Object-Orientation Abusers

Change Preventers

Dispensables

Couplers

Task 1: What does smell here?

```
private static Cyborgs[] = (private static) new get();
private static void cyborgs() {
    // cyborgs() == null {
    // logger.info("...");
    }
    // {
    //     private static void cyborgs() {
    //         logger.info("...");
    //     }
    // }
    // {
    //     private static void cyborgs() {
    //         logger.info("...");
    //     }
    // }
}
```

- BSOAP_RAV.java:

```
private static void cyborgs() {
    // cyborgs() == null {
    // logger.info("...");
    }
    // {
    //     private static void cyborgs() {
    //         logger.info("...");
    //     }
    // }
    // {
    //     private static void cyborgs() {
    //         logger.info("...");
    //     }
    // }
}
```


Task 2: What does smell here?

```
Calendar dateType = typeOf(date.getTime());
if (zone.equals("UTC")) {
    Calendar dateType = Calendar.getInstance(TimeZone.getTimeZone("UTC")); // ["UTC", "UTC"] // :D
} else {
    Calendar dateType = Calendar.getInstance(TimeZone.getTimeZone("UTC")); // ["UTC", "UTC"]; // :D
}

Calendar currentDate = Calendar.getInstance();
Calendar dateType = dateType; // :D
// if you're using the Calendar class, you'll get the same date as the
Calendar dateType = dateType; // :D
Calendar dateType = dateType; // :D
```

Task 2: What does smell here?

```
const date = require('date-fns')(year, month, day, { locale });
if (zone === 'UTC') {
  const date = require('date-fns')(year, month, day, { locale, timeZone: 'UTC' });
} else {
  const date = require('date-fns')(year, month, day, { locale, timeZone: 'UTC' });
}

const currentDate = calendar.getCurrentDate();
const date = require('date-fns')(currentDate); // :D
// if you are using Dispensables Cars: you need to use the last 3 lines
const date = require('date-fns')(currentDate); // :D
const date = require('date-fns')(currentDate); // :D
```

Dispensables:
Comments

Task 3: What does smell here?

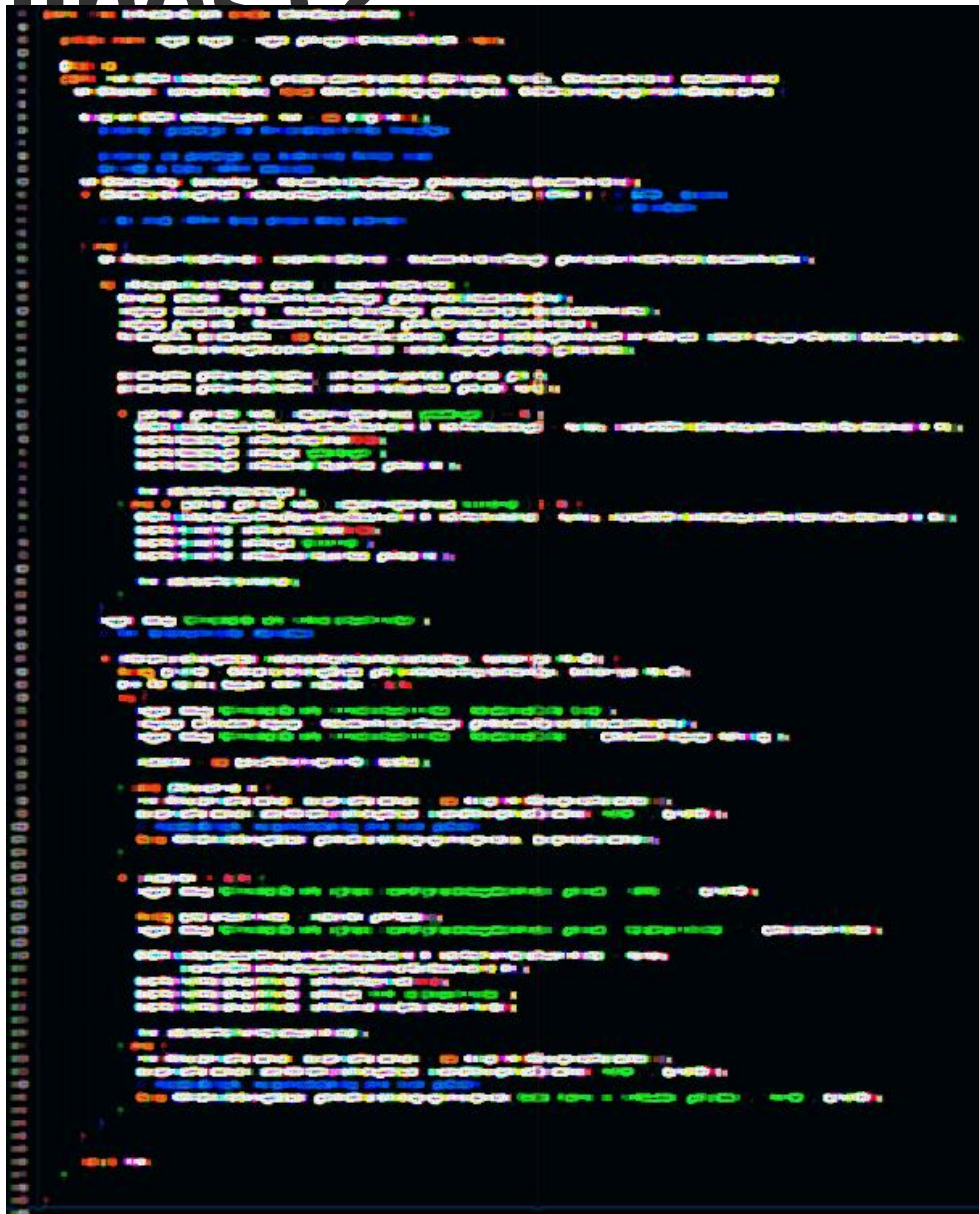
```
public class ... {  
    private long id = 1;  
    @Override  
    public void ... (long id, ... , ... ) {  
        ...  
        ...  
        ...  
        ...  
    }  
}
```

Task 3: What does smell here?

```
public class ... {  
    private long id = 0;  
    public ... (long id, ... , ... ) {  
        ...  
        ...  
        ...  
        ...  
    }  
}
```

Bloaters:
Long Parameter
List

Task 4: What does smell here (without reading the actual lines)?



Task 4: What does smell here (without reading the actual lines)?

The image shows a screenshot of a code editor with a dark background and colorful syntax highlighting. A vertical red bar is drawn along the left margin, extending from the top to the bottom of the page to highlight the extreme length of the code block. The code itself is dense and appears to be a single, very long method or class definition.

Bloaters:
Long Method



Credits

- Code Smell Catalogue:
<https://refactoring.guru/refactoring/smells>