

Self review knowing TDD Habits and Object Calisthenics

Simple rules with great value

- No abbreviation (OC) , Naming convention (TDD)
 - Keep all entities small
 - Only one logical assertion per test
 - Don't mix state and collaboration assertions
 - Refactor aggressively and constantly
-

Abbreviations and Naming problems

```
private Integer[] arr = {1, 2, 3, 4, 5};  
  
public void printArray() {  
    for (int i = 0; i < arr.length; i++) {  
        System.out.print(arr[i] + " ");  
    }  
    System.out.println();  
}
```

```
private Integer[] arr = {1, 2, 3, 4, 5};  
  
public void printArray() {  
    for (int i = 0; i < arr.length; i++) {  
        System.out.print(arr[i] + " ");  
    }  
    System.out.println();  
}
```

```
private Integer[] arr = {1, 2, 3, 4, 5};  
  
public void printArray() {  
    for (int i = 0; i < arr.length; i++) {  
        System.out.print(arr[i] + " ");  
    }  
    System.out.println();  
}
```

Abbreviations and Naming problems

- Not having a convention at the very beginning
 - Short meaningless names
 - Does not express behaviour
 - One must read and understand the code to find it out
 - Hard to reuse
 - IDE suggestions. Does not help if you can't decide by the name
-

Small and focused objects

- Interface Endpoints
 - Mixing inner functions
 - Business logic gets hidden
 - Hard and time consuming to make a change
 - Hard to test
-

Unit Tests Smells

- Name of the test
 - Missing structure
 - Mixing state and collaboration check
 - Iteration inside the test
-

Refactor aggressively and constantly
