# Bad Smell Analysis

The first bad smell is the worst one. The impact of the next bad smell is light than the precious one.

## Bad Smell 1:

Name: Large Class

Location: Folder: code/File: controller.py/Class: Controller/line: 187 to 289

Reasons:

1, A huge amount of code

2, Lack of readability

3, It is the worst bad smell because of it holding most of functions, complicated and hard to maintain

Strategies/approaches: Use Extract Class to split some functions into classes and Composite Pattern collects all the objects of the classes inherited from the parent class.

## Bad Smell 2:

Name: Duplication

Location: Folder: code/ File: validator.py/Class: Validator/ Line: 23 to 100

Reasons:

1, A set of Methods are similar code

2, Huge lines of code

Strategies/approaches: Abstracted Class

## Bad Smell 3:

Name: Data Clumps

Location: Folder: code/ File: employee.py/Class: Employee/ Line: 7

Reasons:

1, Hard to read

2, A long line of code when the method is called

Strategies/approaches: Passing Object instead of the parameters

## Bad Smell 4:

Name: If statement

Location: Folder: code/ File: employee.py/Class: Employee/ Line: 102 to 123

Reasons:

1, Long method

2, inflexible code

Strategies/approaches: Abstracted Class

# Evaluations

Bad smell large class:

100 lines of code become 28 lines

Bad smell duplication:

Divide Validator class into several small class using Abstract factory. In addition, Singleton pattern is also applied then the Validator object will not be spread in the memory.

Bad smell data clumps:

Passing the object instead of a group of parameters make the code easier to read.

Bad smell if-statement:

Apply Abstract factory to discipline the code then I can call the same method from an amount of objects to instead of calling different methods.

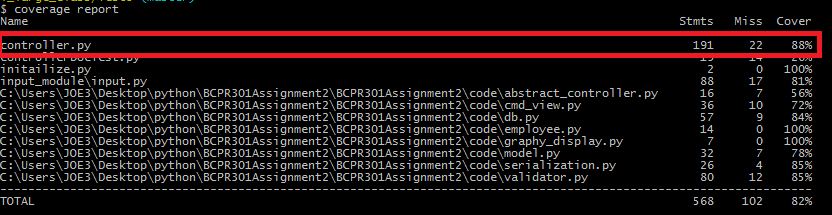
# Self-Marking

1, Smell detection

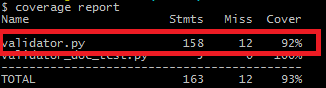
Four bad smells have been detected

2, Testing

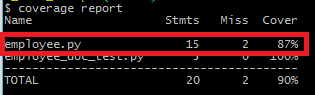
Controller Testing Coverage Report



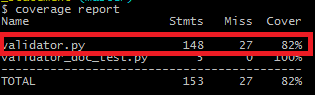
Validator Testing Coverage Report



Employee Testing Coverage Report



If-statement Testing Report



3, Refactoring

Identifying the worst smell and the reasons why it is the worst one

Version control via a remote repository and testing ---- yes

Modification to remove the worst smell and PEP8 validation ----- yes

Effectively evaluations

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Bad Smell 1 | Bad Smell 2 | Bad Smell 3 | Bad Smell 4 | total |
| Smell detection | 4 | 4 | 4 | 4 |  |
| Testing | 4 | 4 | 4 | 4 |  |
| Refactoring | 4 | 5 | 3 | 3 |  |
|  | 12 | 13 | 11 | 11 | 47 |

# Version control URL

https://github.com/Joe3Huang/BCPR301Assignment2

# Refactoring process

