BCPR301 - Assignment 2

# Smell Detection

|  |  |
| --- | --- |
| Name | Duplicate Code (194 lines, 11 methods) |
| Location | PythonInterpreter\models - main.py - MainFlow - <multiple methods> - 38-232 |
| Reasons | 1. All methods have duplicate if-statements that check for flags. 2. Every if statement is similar in checking for either one or no flags. |
| Strategies / Approaches | Create a method for checking number and value of flags inputted that can be called by each method.  Method will take a required number or value of flag and return the given flag or error code.  For testing, each method will test a set of Booleans within the main.py file |

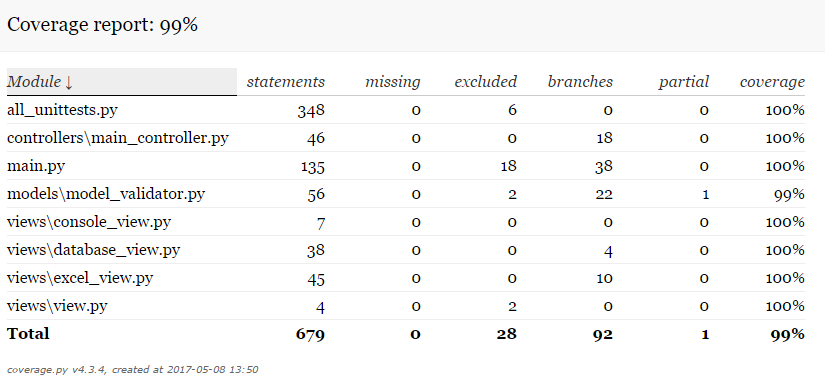
|  |  |
| --- | --- |
| Name | Long Method (33 lines, 1 method) |
| Location | PythonInterpreter\views - database\_view.py - DatabaseView - output() - 18-51 |
| Reasons | 1. 33 lines of code. 2. Code is fulfilling multiple functions such as:    1. Connection to database    2. Creating database    3. Adding data to database |
| Strategies / Approaches | Create separate private methods for handling database functions for connection, creation and adding data. This will also provide future-proofing so this class may be used again for different databases.  For testing, Booleans will be used to check each state of database at the end of each function. |

|  |  |
| --- | --- |
| Name | Switch Case (13 lines, 1 method) |
| Location | PythonInterpreter\models - model\_validator.py - Validate - validate\_all() - 168-181 |
| Reasons | 1. Multiple if/elif statements querying the same thing. 2. Object key being checked for similar results. |
| Strategies / Approaches | Create an Employee subclass to call the validation methods for each field.  Testing shouldn’t change as all validation methods are already being tested. |

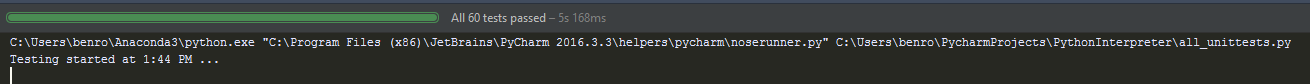
|  |  |
| --- | --- |
| Name | Speculative Generality (2 lines, 2 methods) |
| Location | PythonInterpreter\views - view.py - IView - output() - 15  PythonInterpreter\views - console\_view.py - ConsoleView - output() - 15 |
| Reasons | 1. Parameter created but not used in all implementations. 2. The ‘optional’ parameter is not used in the console view. |
| Strategies / Approaches | Create a second interface (abstract base class) for data handling, rather than the generic I/O interface.  Test only the implementations of IView (Console, Excel, Database) and ensure methods still work correctly. |

# Test Development

Created 60 tests across main, database\_view, model\_validator, console\_view, and excel\_view. These tests cover 99% of branch coverage for all code that will be changed in the refactoring process.



All tests pass as of version 1.00



# Refactoring

## Duplicate Code

### Identify

There is duplicate code that covers 11 methods throughout main.py in the MainFlow class.

The duplicate code exists to determine flags from CMD inputs and act accordingly throughout the method as necessary.

This is the worst smell because of the number of methods and lines that are covered.

### Version Control / Test

All tests are passing before refactoring, and version has been updated to v2.00

All tests continue passing throughout implementations v2.01, v2.02

Modification is required to continue through v2.03 - v2.06

Changes to methods load/save are not necessary, as they act differently.

All tests continue passing throughout implementations v2.07 - v2.09

### Modify to Remove Smell & PEP8

First modification will be to create a method that takes the ‘line’ parameter and an array of expected flags and return either a response (error/success) code or the flag depending if the line matches an expected flag.

On modification, it proved useful to also include the \_\_doc\_\_ string to save further code duplication.

### Evaluate

There is still a fair amount of switch case statements across the various methods, however the duplicated code has been dealt with using method extraction. Length of file has decreased by a small amount (approx. 20 lines) and flag capabilities can be extended using a single method for future implementation of do\_ commands.

## Long Method

### Identify

There is a long method in database\_view.py in the DatabaseView class.

The long method (output()) exists to output data to a database, if there is no existing database, it creates one.

This is the next worse smell because it covers 33 lines of code in only one method.

### Version Control / Test

All tests are passing before refactoring, and version has been updated to v3.00

All tests continue passing throughout implementations v3.01 - v3.04.2

### Modify to Remove Smell & PEP8

First modification will be to extract the connection code from output and place it in a method. This can also be used in the get\_data() method.

Second modification will be to extract the creation code from output and place it in a static method.

Third modification will be to extract insertion code from output and place it in a static method.

Final modification will be to extract closing code from output and place it in a static method. This can also be used in the get\_data() method.

### Evaluate

Output method is significantly reduced (8 lines total, down from 33) meaning it is no longer a Long Method smell. The code efficiency is vastly increased as the get\_data method can also use some of the extracted methods. Upon analysis, it is realised that the code could be made more efficient by removing the hard-coded SQL however that is not something that will be fixed during this refactor.

## Switch Statement

### Identify

There is a switch statement in mode\_validator.py in the Validate class.

The switch statement exists (in validate\_all()) to assign a specific method to each value within a dictionary.

This is the next worse smell because of the number of lines (12) and the excessive use of elif statements.

### Version Control / Test

All tests are passing before refactoring and version has been updated to v4.00

All tests continue passing throughout v4.01 and v4.02

### Modify to Remove Smell & PEP8

First modification will be to create an Employee class with data fields inside \_\_init\_\_ and a validate\_employee method for returning the required checklist. Validate\_employee will call the Validate class methods specific to each field.

Implementation will change validate\_all method to create new employee with valid data.

\*\* Upon creation of employee class, a more efficient fix was found. The switch statement will be replaced with a dictionary literal, with the checklist validating each piece of data appropriately.

### Evaluate

The switch statement is completely resolved, having been replaced with a dictionary literal. The original idea of using an employee class would be more efficient throughout the code, however would require more rigorous testing in place to ensure proper working of the class. To fix this specific smell and still pass all testing, a dictionary literal is the better option to fix the smell.

## Speculative Generality

### Identify

There is Speculative Generality in view.py and console\_view.py in the View and ConsoleView classes respectively.

The Speculative Generality exists to give ExcelView and DatabaseView a third parameter however ConsoleView only requires two.

This is the final smell to diversify the different types of smells being fixed. It is also the next worse because it covers 2 classes (4 including the classes using the parameter).

### Version Control / Test

All tests are passing before refactoring, and version has been updated to v5.00

### Modify to Remove Smell & PEP8

First modification will be to create a new FileView class as an ABC.

First implementation will be to implement FileView in DatabaseVeiw.

Second implementation will be to implement FileView in ExcelView.

Second fix will be to remove the third parameter from View. This will one require implementation changing parameters in ConsoleView.

### Evaluate

The speculative generality is completely resolved, and has been replaced by a secondary ABC for the Database and Excel Views. This leaves the primary ABC for manual I/O only which could later be extended to a logger for error management.