**PR301 Assignment Two**

**Self-Marking Sheet – Sarah Ball**

**GitHub Repository:**

<https://github.com/sab0946/Plant-UML-to-Python-Convertor>

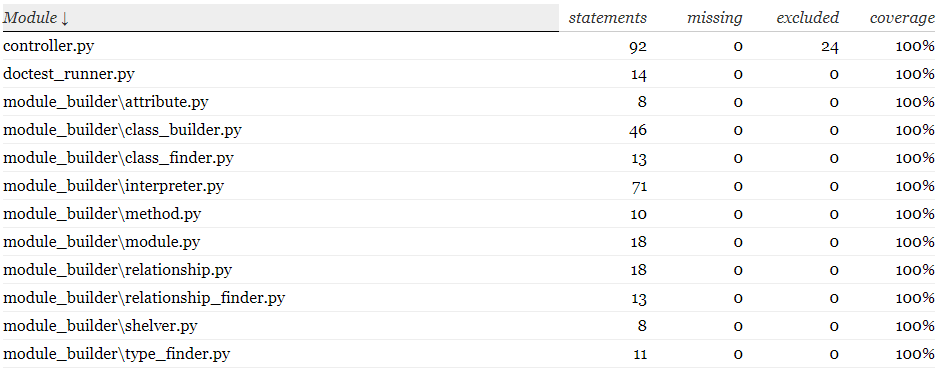
**Tests Developed:**

doc\_tests/attribute\_doctest  
doc\_tests/class\_builder\_doctest  
doc\_tests/interpreter\_doctest  
doc\_tests/method\_doctest'  
doc\_tests/module\_doctest'  
doc\_tests/relationship\_doctest  
doc\_tests/doctest\_controller  
doc\_tests/type\_finder\_doctest  
doc\_tests/class\_finder\_doctest  
doc\_tests/relationship\_finder\_doctest

All run from:

doctest\_runner.py

**Coverage Report:**



htmlcov/index.html

**Smell One:**

**Name:** Large Class

**Location:**

module\_builder/interpreter-Interpreter-between lines 8-113

**Reasons:**

|  |  |  |
| --- | --- | --- |
| **Identifying the worst Smell:** | |  |
| Smell | Appearances | # Lines Affected |
| Large Class | 3 | 358 |
| Duplicate Code | 6 | 36 |
| Long Method | 1 | 16 |

1. The interpreter class is performing more than one function, with static methods and long methods contained within the class.
2. This means that the class is harder to understand, de-bug and extend
3. The ClassBuilder class contains repeated and long methods, but ultimately is performing more than one task. This makes it harder to understand and extend.

**Strategies:**

Large class is the biggest issue - unsure if the Main (controller class) could/should be split. Starting with the second largest class - Interpreter

1. Decide what the class should actually do
2. Split the class into classes with single responsibilities
3. Use the Extract Sub-Class method to create sub-classes for the interpreter
4. Use the Move Method and Move Attribute methods to populate relevant methods and attributes into these new sub-classes
5. Move the new classes into separate files
6. Re-evaluate the code to see if Large Class is still the worst smell

**Version Control:**

**GitHub patch 2-15**

**Evaluation:**

Successfully split the large class into smaller classes, however there is still duplicated code across the methods, and now may be other bad smells created by the new classes. Before moving onto the next bad smell, a corrected class diagram will be drawn up to ensure that the worst bad smell remains the same.

**Smell One - Continued:**

**Name:** Large Class

**Location:**

module\_builder/class\_builder-ClassBuilder-between lines 6-76

**Reasons:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identifying the worst Smell:** | | |  | |
| Smell | Appearances | # Lines Affected | |
| Large Class | 2 | 253 | |
| Duplicate code | 2 | 22 | |
| long method | 1 | 16 | |
| refused bequest | 1 | 16 | |
| Temporary Field | 1 | 3 | |
| Switch Statements | 7 | 64 | |

* Large class still the biggest issue - although two new bad smells have been added (temporary field & refused bequest). Decided to split up the ClassBuilder class first, using the same method above.

**Strategies:**

Looked at the code, and decided that the Long Method and duplicate code within the class is what needs to be removed to reduce the size of the class - it still only has one function.

Used in-line method to reduce the size of the class

**Version Control:**

**GitHub patch 16-22**

**Evaluation:**

Class is still large, but has only one function and has been shrunk by 20 lines of code.

Now it has been established that the class has only one function it is time to move onto the next bad smell.

**Smell Two:**

**Name:** Switch Statements

**Location:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Location** | module name | /file name | class name | method name | start line | end line |
| ex1 | module\_builder | method | Method | find\_type | 10 | 21 |
| ex2 | module\_builder | attribute | Attribute | find\_type | 21 | 32 |
| ex3 | module\_builder | class\_builder | ClassBuilder | add\_relationship | 38 | 48 |
| ex4 | module\_builder | relationship\_finder | RelationshipFinder | find\_relationship | 7 | 22 |
| ex5 | module\_builder | attribute | Attribute | \_\_str\_\_ | 11 | 21 |

**Reasons:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identifying the worst Smell:** | | |  | |
| Smell | Appearances | # Lines Affected | |
| Large Class | 1 | 183 | |
| long method | 1 | 16 | |
| refused bequest | 1 | 16 | |
| Temporary field | 1 | 3 | |
| Switch Statements | 6 | 54 | |

* Have decided to still leave the Controller class alone for now, as it appears that it will remain a large class. Switch statements now affect the most lines of code within the rest of the program.

|  |  |
| --- | --- |
| 1 | complicated switch statements are hard to understand |
| 2 | embedding if/elif statements, and multiple elif statements make it hard to track what is happening to the data in a class |
| 3 | it also makes classes longer than they need to be |

**Strategies:**

|  |  |
| --- | --- |
| 1 | Same switch statement appears in two places - first remove this duplication 1. Move the method out of each class - create as a static method 2. Change the references inside each class 3. Adjust the method to include a default variable type of None |
| 2 | Replace switch statements with dictionaries containing the associated variables and methods |
| 3 | Look for any further duplication that arises from this refactoring |
| 4 | remove any duplication found and re-assess the code |

**Version Control:**

**GitHub patch 23-28**

**Evaluation:**

The duplication was easy to remove to a separate class, with static method. This single method can now be refactored to remove the Switch Statement bad smell, without needing to change the code in two places.

Add\_relationship - switch statement replaced by methods in the Relationship class. This keeps the responsibility of knowing the relationship type within the relationship class, without the need for a complex if/else statement.

find\_relationship - removed un-used if-statement to tidy up the complexity, no need to remove completely.

\_\_str\_\_ method - replaced the if/elif statements with a dictionary

**Smell One - Three:**

**Name:** Middle Man

**Location:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Location** | module name | /file name | class name | method name | start line | end line |
| ex1 | module\_builder | module\_adder | ModuleAdder | add\_module | 4 | 11 |
| ex2 | module\_builder | class\_adder | ClassAdder | add\_class | 4 | 11 |

**Reasons:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identifying the worst Smell:** | | |  | |
| Smell | Appearances | # Lines Affected | |
| Large Class | 1 | 183 | |
| long method | 1 | 16 | |
| refused bequest | 1 | 16 | |
| Temporary field | 1 | 3 | |
| Middle Man | 2 | 16 | |

|  |  |
| --- | --- |
| 1 | The classes are only there to delegate to another class |
| 2 | This adds unnecessary complexity to the class |
| 3 | this means the code is harder to read and understand |
| 4 | This issue was added when the class-size was reduced |

**Strategies:**

|  |  |
| --- | --- |
| 1 | Identify where the methods within these classes are being called |
| 2 | use the move method technique to move the methods into this class |
| 3 | delete the now unnecessary class |

**Version Control:**

**GitHub patch 28-32**

**Evaluation:**

This refactoring has made the FileReader class bigger again, but kept all of the functionality in one place (reading a file) with no middle mad classes.

This makes the read\_file method easier to understand, as you no longer have to look into a separate class to work it how it works.