Kieran Abelen

Sidder19@hotmail.com

Assignment 2

Coder and Tester refactoring

Contents

[Repository link: 2](#_Toc19946827)

[Refactoring 2](#_Toc19946828)

[Introduction 2](#_Toc19946829)

[Smell detection 2](#_Toc19946830)

[References 3](#_Toc19946831)

Coder – Kieran Abelen

# Repository link:

Original: (Marcus)  
Branch repository: (Kieran)

# TIGr Solution

## Brief

There are lot of duplicate classes present, 3 parser subclasses and 3 drawer subclasses. The will account for multiple bad smells  
Based off Mikes restriction of not changing the template parent class, there will be a few bad smells from this like Incomplete library class.

Smell detection

### Shotgun Surgery

Lots of classes have identical functions so to change one mean to change them all.

#### Location

1. DrawerKieran.py, DrawerJack.py: Of the three drawers both Tkinter drawers with the exact same functions.
2. ParserTK.py, ParserJonathonV2.py, ParserJerry.py: Three parsers are present, and all have the same functions.

#### *Reason*

1. Changing any Tkinter draw\_line or other functions would require repeated changes in each drawer.
2. Any changes to the commands or accepted data would result in changes in all of them.

#### Refactoring Strategies

* Move Method: This would create a single class of all the methods
* Move field: Would just move the data but would keep the classes this won’t remove the shotgun.
* Inline Class: The parent class is abstract so we shouldn’t add functionality to it.

### Duplication Code

The same with the shotgun lots of classes have duplicate code

#### Location

1. DrawerKieran.py, DrawerJack.py: These two classes have duplicate functions
2. ParserTK.py, ParserJonathonV2.py, ParserJerry.py: These three class have duplicate functions and even have similar but different features like regular expressions filter but no one way of doing it.

#### *Reason*

1. Tkinter drawers have almost the exact same code with DrawerKieran.py having a smaller draw\_line
2. There are differences inside the functions, but they all do the same thing and ParserJonathonV2 has the best functionality.

#### Refactoring Strategies

* Extract Method: Simplest and the most universal option
* Extract Class: Normally you extract part of the class but could be used for whole thing.
* Pull Up Method: Template pattern is the abstract parent class so adding functionality is not advised.
* Form Template Method: this could work with the drawer, but it technically already exists with the template pattern.

### Incomplete Library Class

#### Location

#### *Reason*

1. To many Ifs to achieve relatively little.
2. Searches the same file for different things in the if very ineffective.
3. Better but still sorts each command against a single condition.

#### Refactoring Strategies

1. Replace Temp with Query: Use simple maths to eliminate the Ifs
2. Replace Method with Method Object: You only need to search once and compare it against a

### Refused Bequest

#### Location

1. DrawerJack.py. line 44: has more than 10 line if statements
2. ParserKC.py, line 11: has a multiline if search.
3. ParserJerry.py, line 16: if statements sorting a find all from an input.

#### *Reason*

1. To many Ifs to achieve relatively little.
2. Searches the same file for different things in the if very ineffective.
3. Better but still sorts each command against a single condition.

#### Refactoring Strategies

1. Replace Temp with Query: Use simple maths to eliminate the Ifs
2. Replace Method with Method Object: You only need to search once and compare it against a

### Long Method

#### Location

1. DrawerJack.py. line 44: has more than 10 line if statements
2. ParserKC.py, line 11: has a multiline if search.
3. ParserJerry.py, line 16: if statements sorting a find all from an input.

#### *Reason*

1. To many Ifs to achieve relatively little.
2. Searches the same file for different things in the if very ineffective.
3. Better but still sorts each command against a single condition.

#### Refactoring Strategies

1. Replace Temp with Query: Use simple maths to eliminate the Ifs
2. Replace Method with Method Object: You only need to search once and compare it against a lookup table

Extract Class: This method would solve both since DrawerKieran.py already solved this issue so did ParserJonothanV2

### Long Class

#### Location

ParserJonathonV2.py: The whole class except the parser part on line 31

Reason

This class is doing many things.

* Lookup table stored inside the class.
* The regularly expression sorting is also here.

The real issue is if there two things are added to, they would really make the class to big.

#### Refactoring Strategies

1. Extract Class: This will be a great solution
2. Extract Subclass: could work but there are two extractions
3. Extract Interface: This class doesn’t have any overly complex dependencies so this wont the solution.

The simplest solution is Extract class since it would just create two subclasses.  
Two static subclass, one with the lookup table and the other with regular expression filter.

Refactoring

### Priority

#### Shotgun Surgery

This is the worst bad smells since any other refactors might need to take place I’m multiple classes so by moving all the duplicate methods into one class will make future refactoring and testing easier.

#### Duplicate code

Same as shotgun basically but does not necessarily result in the removal of classes just extracted duplicate methods.

#### Incomplete Library Class

#### Refused Bequest

#### Long Class

This will reduce functionality and dependency on the parser class. Not as important to fix.

#### Long Method

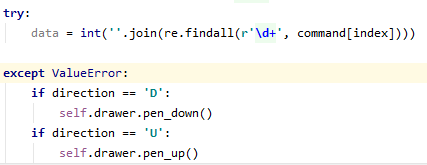
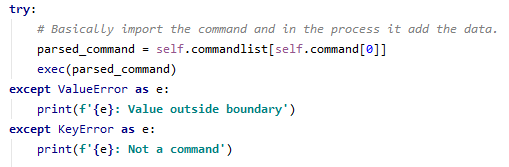
Low priority since it will be not only fixed in duplications code bad smell but wont impact the code much if not changed.

Version control via an online repository (Kieran)

### Refactor 1: Shotgun Surgery

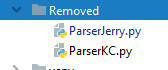
Fixes both Duplicate code and Long Method too.

1. Move Method into ParserJonathonV2.py
   1. The only thing moved was the Try and Except from ParserJerry.py

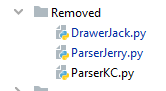
 

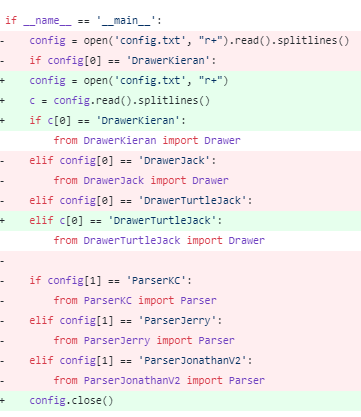
ParserJerry.py Moved -> ParserJonathonV2.py

1. Remove the other two parser classes: No longer needed



1. Remove duplicate DrawerJack.py: Since it is a worse DrawerKieran.py Tkinter drawer.  
   No Move method required.



1. Remove the classes being imported  
   
2. Testing and effectively evaluations on your refactored code in a concise fashion (N marks)

# References

Kieran. (n.d.). *mellyonz/Assignment\_2\_G2-\_Kieran-Tester.* Retrieved from github.com: https://github.com/mellyonz/Assignment\_2\_G2-\_Kieran-Tester

Marcus. (n.d.). *forestraindrip/PR301\_Assignment2: The source code for PR301 Assignment2.* Retrieved from github.com: https://github.com/forestraindrip/PR301\_Assignment2