Refactor 1

# Identifying the Worst Smell

The switch statement smell in the Controller class is the worst smell because this makes the Controller look very confusing as it is full of if/else statements. Fixing this would improve the readability. I would fix the valid() method switch statement smell.

# Version Control Repository

<https://github.com/ism0080/Python_badsmell_PR301>

# Modification to Remove the Worst Smell

Switch Statement Smell – Controller = valid()

1. Find the smell
2. Created the dictionary of option flags
3. Created a method for the flag to run
4. Removed a flag as I found it was duplicate code
5. Smell was fixed

# Effectively Evaluations

This went well as I fixed the bad smell and didn’t introduce any new bad smells. I also found that there was a duplication smell that didn’t need to be in that part of the code. The controller is a little bit more readable now but the rest of the switch statements in the Controller would need to be refactored the exact same way as this for that to happen.

Refactor 2

# Identify the Worst Smell

The comment smell in the Reader class is the worst smell because, this smell could be also considered a long method, but it makes more sense to be a comment smell because of the comments. If I extracted the different methods then they could become useful to use again separately rather than calling it as one method.

# Version Control Repository

<https://github.com/ism0080/Python_badsmell_PR301>

# Modification to Remove the Worst Smell

Comment Smell – File Reader

1. Moved variables 'line' and 'list\_of\_dictionaries' to the initializer of FileReader
2. Renamed ‘group’ to ‘new\_line\_split’
3. Extract a method called ‘splits\_on\_comma\_separation’
4. Extract a method called ‘create\_list\_of\_dictionaries’
5. Removed Comments
6. Smell Fixed

## Effectively Evaluations

This went well as I fixed the bad smell and didn’t introduce any new bad smells. The Reader class is now more readable and flexible, it will be easier to add new functionality if needed. The reader is higher in software quality now.

Refactor 3

# Identify the Worst Smell

The duplicated code smell in the display module, PyGal class is the worst smell. This smell is worse than the other ones but I hadn’t realised this until now. It is bad because it also affects the Controller as the implementation code for it is in the Controller so I will need to refactor in both files to fix this smell. Fixing this smell will make the chart maker a lot more flexible.

# Version Control Repository

<https://github.com/ism0080/Python_badsmell_PR301>

# Modification to Remove the Worst Smell

Duplicated Code Smell – Controller: pygal / py\_view & display: Pygal = bar\_char

1. Identified the Smell
2. Added a for loop that will add the data values from the database to a dictionary instead of inputting all the data as separate variables. The user is asked for input. The flag will now be the number of fields that the user wants to graph
3. Added the new for loop to the try statement. Updated the py\_view method so that it now accepts the new input
4. Fixed the display bar\_char() method so that it now gets the data to add to the chart from the dictionary, this removes the duplication
5. Fixed Documentation, Tests and Titles to match the smell fix

# Effectively Evaluations

This went well as I fixed the bad smell and didn’t introduce any new bad smells. The PyGal class is now flexible and it is a lot easier to choose more fields to view in the chart, it will be easier to add new functionality if needed. It is also a lot easier to understand what is happening in the controller and display when creating the PyGal chart. Display is more readable and Controller is slowly becoming more readable.

Refactor 4

# Identify the Worst Smell

The duplicated code smell in the Controller class, pickled method is the worst smell. This smell is bad because I am duplicating code, by doing this I’m also creating a switch statement so if I wanted to add more functionality then I would be extending the switch statement which would make it even harder to maintain. Fixing this smell will improve readability and flexibility.

# Version Control Repository

<https://github.com/ism0080/Python_badsmell_PR301>

# Modification to Remove the Worst Smell

Duplication code smell – Controller: pickled()

1. Located the smell
2. Created an ‘options’ dictionary for the flags
3. Created a method to check if the name is empty or not for the pickle file
4. Added try / except to new method
5. Removed old switch statement from the pickled method
6. Didn't need the extra method so merged the two
7. Smell fixed

# Effectively Evaluations

This went well as I fixed the bad smell. I almost introduced a bad smell but then realised that I had created a method that I didn’t need to use, so no new bad smells were added. The duplication has now been removed from the pickled() method and is now easier to read. This also increased the readability of the Controller class. The software quality is slowly increasing with each refactor.