Peer review for group 75 by group 92

Pricing Policy

Firstly we noticed that the way pricing policy was implemented which was that it is heavily integrated into the whole system, and cannot work as a submodule on it's own. So the first improvement that could be made is to separate out the pricing policy submodule from the rest of the system and make it work independently. This would massively increase readability as well as make testing much easier.

Next we thought that some of the variable names were quite vague(what is myBase, myIncrement) and naming was often inconsistent since sometimes they made the second word of a variable a capital letter and sometimes not(e.g. myBase and dailyPrice but then surname, forname, postcode) I and sometimes not. Also we noticed some inconsistency with their use of lists, sometimes they used arrayList other times collections.

We also thought that the use of the iterator to calculate the price was clever and efficient way to do it. There were also a methods that is never used(addNewBike).

Test

The most obvious thing we noticed in the test was that things that should have been done in the setup were done in actual test methods and the things they did do in the setup were also done in some methods leading to unnecessary redundancy. We also found two tests(testSetMyDailyRentalPrice, testSetMyDailyRentalPrice2) which were completely identical and tested the same things, we assume this was just a mistake. Despite these the test class was still very readable.

In the calculatePrice method default values are declared and initialized when needed, initializing them earlier in the class as defaults to be used when needed would reduce the complexity of the code.

Some of their data was quite limited like they had made 3 providers but only 1 bike. The tests did cover the main use of the pricing policy interface but due to their limited data they weren't able to test extreme data.