Refactoring Tasks

Qi Liu

April 2019

1 Motivation

It is very significant to refactor your code to make it more readable and reusable. From my Event Simulation program, i realized there are several problems which could potentially confuse people who read it and also potentially cause some errors.

2 Tasks

2.1 Remove Duplicate Code

There are some parts of my BankSimulator program containing duplicate code. For example, i had a method addNewCustomer(Customer customer) which adds a new customer to the queue for waiting to be served. And I also had a method updateCustomerQueue which also includes a part which does the same thing as addNewCustomer method. So i should delete some duplicate methods or integrate those methods together to make the program cleaner.

2.2 Change Poor Names

When i write BankSimulator, i gave name like customerQueue and minTime which i thought is clear enough to understand. However, later i found i need more queue objects related to customer object which means i need to give clearer names to distinguish different customer queues.

2.3 Make Programs More Object Oriented

Since there are many shared method signatures between BankSimulator and SupermarketSimulator, it is probably better to make an interface containing common method signatures for both simulator to override. And it is also useful if we want to add more other types of simulators.

2.4 Change Magic Numbers to Constant

In BankSimulator and SupermarketSimulator, the number of clerks/checkers is used a lot in some methods, and also i simply used a number instead of a variable with a name indicating what it is. First of all, I need to create a variable which has a name like numOfClerksOrCheckrs instead of a number. And i also need to declare it as a constant variable out of classes since it is reusable and its value is never changed.