Part 2 Additional Features

This document’s purpose is to give an outline/insight into the additional features implemented in my solution to Assessment 2 of CSC1029 Object Oriented Programming. These additions include:

* An engineer menu where a VendingMachine instance can be maintained and updated.
* The tracking of coins contained in the VendingMachine instance, coins input by the user along with what coins their change consists of.
  + MoneyBox class to manage and maintain this.
* Functionality to allow saving/loading of machine state.

Q2.1: Hidden Engineer Menu

I implemented the EngineerMenu class which can be accessed through the main VendingMachineApp menu (details as to how found in the readme). Through the EngineerMenu, the engineer can view all items, view a single item’s details, add an item, restock an item, view the data regarding a vending machine, set the status of the machine and reset the machine to default.

Q2.2: Coin Denominations

This feature keeps track of coins contained in the machine, along with coins inserted by the user. Coins will be inserted by the user and stored in the machine; when the user purchases an item, if the machine contains enough coins to return change, it will break the user’s change down into the coins that make it up and output the returned coins to the user. If the machine doesn’t contain enough/can’t give certain coins, it will inform the user which coins it couldn’t return. This feature was done using the MoneyBox class I created.

Q2.2 EXTRA: The MoneyBox Class

To track user/machine coins, I implemented the MoneyBox class which essentially stores the count of £2, £1, 50p, 20p, 10p and 5p coins and provides functionality like checking equality of two MoneyBoxes, adding of MoneyBoxes, checking if a MoneyBox contains a coin, just to name a few. I originally implemented Q2.2’s functionality using ArrayLists, but I ended up having 4 ArrayLists in the VendingMachine class, which I thought was quite inefficient.

Q2.3: State Storage and Retrieval

This functionality allows for the saving and loading of a vending machine’s state. This includes the vending machine’s data, MoneyBox values/inserted/total coins in the machine, along with all stock contained in the machine. When the system is launched, it should try read the state from VendingState.csv and if the data is found, it will load the state of the machine; if not, it will create a new vending machine with several items contained. When the system is closed, the state will be saved to the csv file.