Vending machine app

Console-based application

Giuseppe Pecorella (c146 – JAVA)

M3 - London

Table of Contents

[1. Introduction 2](#_Toc86747610)

[2. Class diagram 2](#_Toc86747611)

[3. Project structure and VMC pattern 3](#_Toc86747612)

[4. Design and final consideration 4](#_Toc86747613)

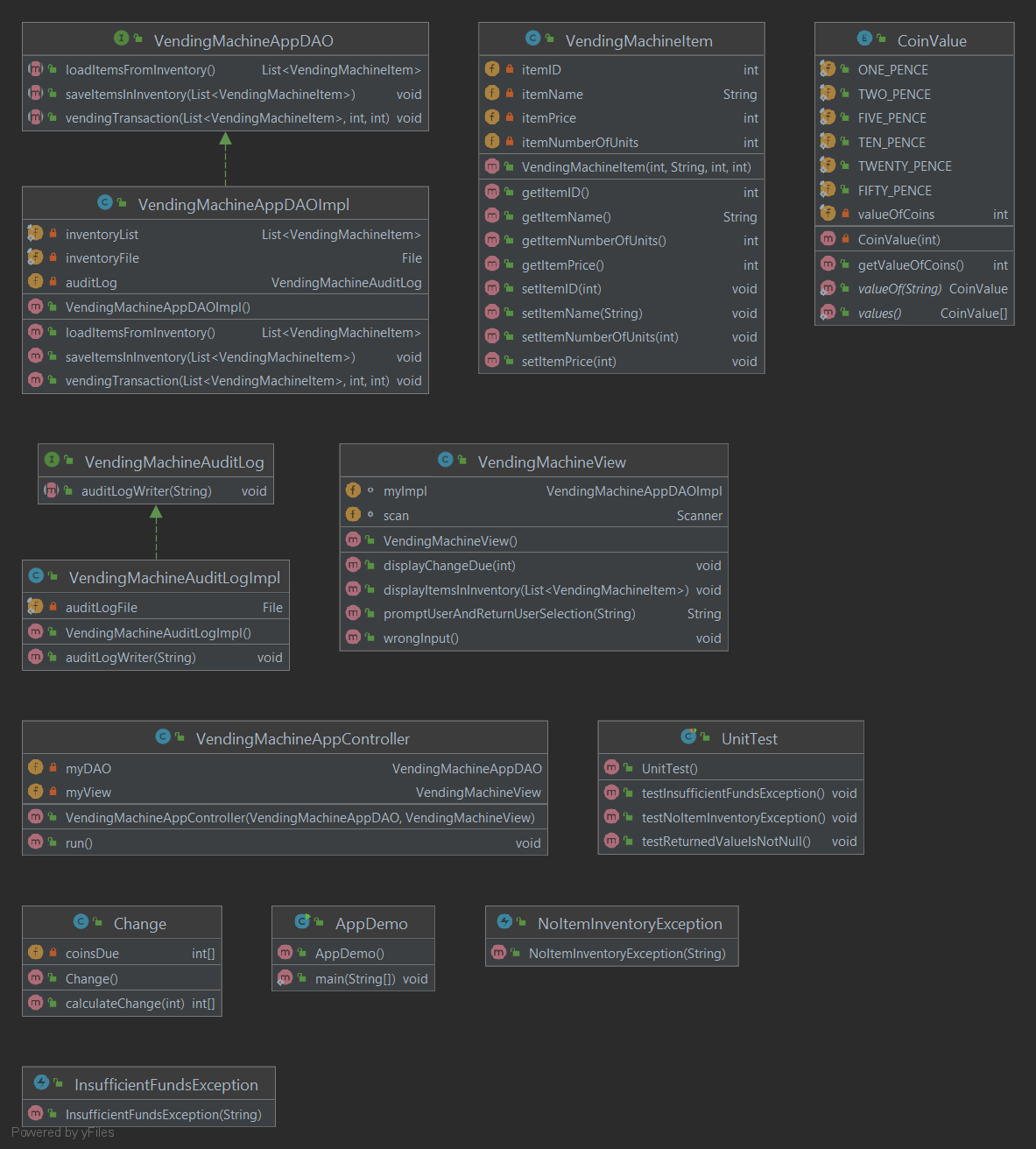
# 1. Introduction

This document is intended to accompany the release of Vending Machine App, a Java console-based application.

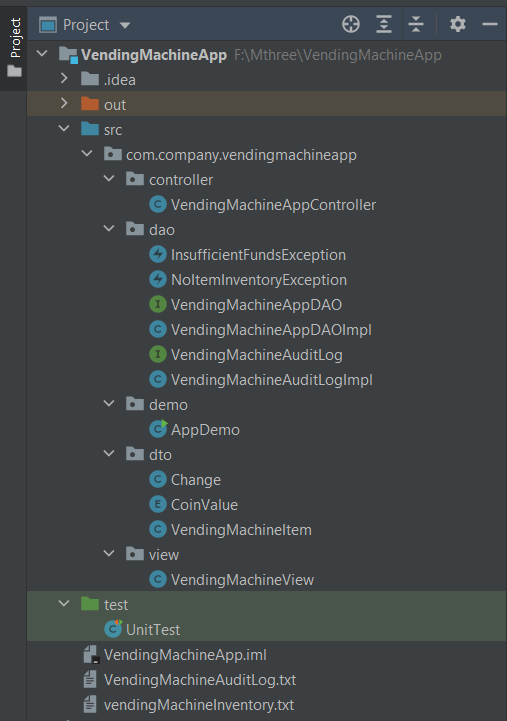
The application is written for a vending machine of snacks and caters for typical issues related to this type of devices.

The application data, both vending machine items and audit log, is stored in a .txt file.

# 2. Class diagram



# 3. Project structure and VMC pattern



As visible in the project diagram snapshot above, the application was design and developed following the MVC (Model, View, Controller) pattern. Each component of MVC is developed inside its own package to allow for separation of functions.

Package – *com.company.vendingmachineapp.controller*

* Class – *VendingMachineAppController*. Responsible for coordinating the other MVC components by calling their methods when appropriate during the application execution

Package - *com.company.vendingmachineapp.dao*

* Exception – *InsufficientFundsException*. Thrown when the money entered by the user is less than the selected item price
* Exception – *NoItemInventoryException*. Thrown when the user tries to buy an item with zero level in inventory
* Interface – *VendingMachineAppDAO*. Contains abstract methods for DAO
* Class - *VendingMachineAppDAOImpl*. Implements the relative interface; deals with CRUD operations
* Interface – *VendingMachineAppAuditLog*. Contains abstract method to write an audit log
* Class – *VendingMachineAppAuditLogImpl*. Implements the relative interface

Package - *com.company.vendingmachineapp.demo*

* Class – *AppDemo*. Contains the main method

Package - *com.company.vendingmachineapp.dto*

* Class – *Change*. Responsible for calculating the change due to the user after vending transaction
* Enumeration – *CoinValue*. Defines the value of coins available in the vending machine
* Class – *VendingMachineItem*. Represents the items to be vended

Package - *com.company.vendingmachineapp.view*

* Class – *VendingMachineView*. Responsible for the user interface; displays messages and prompts the user

Package – *default*

* Class – *UnitTest*. A set of unit test on DAO methods

# 4. Design and final consideration

Even though suggested in the project guidelines, I did not use BigDecimal as the data type for change calculation; reason for this being that since the requirement is to display the change in units of fifties, twenties, tens, fives, twos and tens coins I think that the use of integer numbers would be more suitable with this requirements instead of decimal numbers.

In developing the application, I deliberately considered not all the corner cases but at least those which are more pertinent to the purpose of the application, for instance the user entering the ID number for an item not in the list or not entering a number at all.