COMP3522 Assignment 2: The Supply Chain

Rahul Kukreja, Jeffrey Yim
jyim3@bcit.ca
BCIT CST - Computer Systems Technology Diploma

Introduction

The second COMP 3522 assignment for the semester is here!

In this assignment you will be moving beyond the Python Fundamentals and start writing systems that include Design Patterns! You will be working in groups of two (2) for this assignment. So find a partner, and create a **NEW** Github repo for this assignment. (I don't want anyone else having access to your personal COMP3522 repo except yourself). Name this GithubRepo COMP3522_Assignment2_A########_A######## , where A######## is the student ID number of the group members.

For this assignment we will be simulating a store that keeps festive seasonal items all year long. That is, it keeps a stock of toys, stuffed animals, and candy that vary by holidays (Easter, Christmas and Halloween). So if you ever wanted easter eggs in december, this would be the place to go!

Your code will be taking in bulk orders that come in from the store's website in the form of excel files. The system should implement the abstract factory pattern to achieve this.

When going through the assignment brief start drawing out a preliminary UML class diagram to identify all the classes, attributes and any behaviors that you may need. A draft diagram should be made before you start writing any code. Be creative and enjoy the process! I encourage you to come discuss your designs with me if you want any feedback.

Submission Requirements

1. The penalty for late submissions will be 25% of the total grade per day that it is late.
2. Submit your .py files and UML diagrams in a separate GitHub repository following the naming convention *COMP3522_Assignment2_A########_A########* , where *A########* is the student ID number of the group members..
3. This is a group assignment. All code must be written by the group members only. I encourage you to discuss and share ideas with your friends but remember to write your own code!
4. Include a Readme file that describes how your application works and if there are any errors or use cases that have limitations or if it doesn't meet any of the requirements. This is also the spot where you want to document any features (the ones specified in this assignment and any

extra features) that you may or may not have implemented. This will help me keep an eye out for them as I grade your work.

Grading

The assignment is marked out of **24**. For full marks, you must:

1. Correctly implement the requirements described in this document - **10 Marks**
2. Correctly format and comment your code. Eliminate all warnings offered by PyCharm, follow PEP 8 and PEP 257 guidelines, use good function and variable names, write code that is easy to understand, use whitespace wisely, write good and appropriate docstrings, etc. - **2 Marks**
3. Structure/design your classes to maximize code re-use, make it readable and maintainable. Follow the SOLID principles and the Law of Demeter. **- 4 Marks**
4. Submit a UML Class Diagram depicting appropriate use of OOP principles, inheritance and design in your code- **2 Marks**
5. Submit a UML Sequence Diagram depicting the flow of control when an order is ringed in. **- 2 Marks**
6. Handle errors and unexpected player behavior (read: input) gracefully using the Easier to Ask for Forgiveness philosophy. - **3 Mark**
7. Format your output. Make sure your messages display the correct information in a pleasant, readable manner. You could even use ASCII art if you dare! - **1 Mark**

Please remember that this is a group assignment. Any code/ascii art taken from the web must be referenced/cited!

Good luck, and have fun!

Implementation Requirements

Your storefront (give your store a name!) should implement the following features.

The User Menu

When the program runs, it should provide a terminal menu that the store owner would have access to. The menu should let the cashier.

- **Process Web Orders**
  At the end of each day the store owner downloads an excel file of all the online orders placed that day and process them through the system.

- **Check Inventory**

This allows the cashier to check what is currently in stock and will also provide a status indicator for items if the stock for this item is **LOW**, **VERY LOW**, **IN STOCK, or** OUT OF STOCK.
    - In Stock - 10 or more items in stock
    - Low - Less than 10 items
    - Very Low - Less than 3 items
    - Out of Stock - 0 items

- **Exit**
  Exits the program and prints out the daily transaction report.

The Inventory

The store maintains the following items:

Toys

For each festive season, the store stocks a unique toy. Despite that, there are some properties of each toy that all toys have in common. These are:

- Whether the toy is battery operated or not.
- The minimum recommended age of the child that the toy is safe for.
- A name
- A description
- Product ID (A unique combination of letters and numbers)

The holiday specific toys are:

1. **Santa's Workshop**
   The premium Christmas present, this is not a battery operated toy. The doll house comes in different varieties. They can vary in:
    - dimensions (width and height)
    - The number of rooms

2. **RC (Remote Controlled) Spider**
   The RC Spider is the toy to get during Halloween. This toy is battery operated. The different varieties of spiders that are sold have the following properties:
    - Speed
    - Jump height
    - Some spiders glow in the dark, while others do not.
    - The type of spider - This can either be a Tarantula or a Wolf Spider and nothing else.

3. **Robot Bunny**
   The Robot Bunny is the toy for toddlers and infants out there. The toy is battery operated. These come in different varieties as well! Their properties are:
    - The number of sound effects
    - The colour - This can be either Orange, Blue, or Pink and nothing else

Stuffed Animals

All stuffed animals have the following attributes:

- Stuffing - This can either be Polyester Fiberfill or Wool
- Size - This can either be Small, Medium or Large
- Fabric - This can either be Linen, Cotton or Acrylic
- Name
- Description
- Product ID

The holiday specific stuffed animals are:

1. **Dancing Skeleton**

The dancing skeleton is made out of Acrylic yarn and stuffed with Polyester Fiberfill. This skeleton is sure to add to your Halloween decorations.

- The dancing skeleton also **glows in the dark**.

2. **Reindeer**

The reindeer comes with its very own personal mini sleigh and is the stuffed animal for Christmas. It is made out of Cotton and and stuffed with Wool.

- Has a glow in the dark nose.

3. **Easter Bunny**

The Easter Bunny is made out of Linen and stuffed with Polyester Fiberfill.

- It comes in different **colours** - White, Grey, Pink and Blue and nothing else.

Candy

All candies have the following properties:

- A flag to check if it contains any nuts
- A flag to check if it is lactose free.
- Name
- Description
- Product ID

The holiday specific candies are:

1. **Pumpkin Caramel Toffee**

The Pumpkin Caramel Toffee is Halloween themed and is not lactose free and may contain traces of nuts.

- It comes in two varieties — **Sea Salt** and **Regular**.

2. **Candy Canes**

Candy Canes are Christmas themed. It is lactose free and does not contain nuts.

- The stripes on the candy cane can either be **Red** or **Green**

3. **Creme Eggs**

Creme Eggs are Easter themed and are not lactose free and may contain traces of nuts.

- Pack Size - The creme eggs come in different packets, each containing a different number of creme eggs.

Process Web Orders

The store owner can go to their online storefront and download the orders received during the day in the form of an excel sheet.

Your code must prompt the user for the name of this file and use the `pandas` package to read them in and process the order.

pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of Python. Check out how to use pandas here: https://www.lynda.com/Data-Science-tutorials/pandas-Essential-Training/636129-2.html. You should be able to log into Lynda.com via your BCIT student accounts.

The store owner can do this multiple times a day if need be. I recommend you create variations of the excel file that I have provided and use it to test your program.

Your code must contain an `OrderProcessor` class that is responsible for reading each row of these files and creating and yielding an `Order` object. The OrderProcessor class contains a `FactoryMapping` which maps the holiday to the appropriate factory class.

Each `Order` contains the following:

- Order number
- Product ID
- Item - The type of item (Toy, StuffedAnimal and Candy).
- Name of the item
- A dictionary of product details. These details are the rest of the attributes of the item as specified in the excel sheet **EXCEPT** the name of the holiday — Easter, Christmas or Halloween.
- The order should also contain a reference to the appropriate `Factory object` that can create this item.

Now, on to the important bits, what happens with these `Order` objects?

They are sent to the store (more on the store below). The store should find the item in its inventory and reduce the quantity accordingly. In the event an item is not found, the store can use the corresponding factory sent as part of the order to order more items. (More on this below).

The Storefront

Your system should contain a `Store` class that should be responsible for:

- Receiving orders and maintaining its inventory
- Getting items from a factory class if the store does not have enough stock
- Creating the Daily Transaction Report

The first time the store receives an order for an item, it is likely that it won't have the item in it's inventory since the store should be initialized with an empty inventory.

In the event the store receives an order for an item that it does not have inventory for, then it should go ahead and get a 100 of those items made by the corresponding factory class.

Daily Transaction Report

When the user chooses to exit the program, the program should write the Daily Transaction Report to a text file. The file specifies the list of orders processed that day.

The text file should follow the naming convention `DTR_DDMMYY_HHMM.txt` where `DDMMYYY` refers to the date, month and year (for example, 19th Feb 2020 would be 1902020) and0 `HHMM` refers to the hour and minute (for example 1:30pm would be 1330).

Be sure to follow the formatting in the example below:

```
HOLIDAY STORE - DAILY TRANSACTION REPORT (DRT)
05-03-2020 17:58
Order 102, Item StuffedAnimal, Product ID H9405S, Name "Skelly the Tap
Dancer", Quantity 3
Order 103, Item Toy, Product ID T2134C, Name "Santas Workshop Deluxe
Edition", Quanity 20
Order 104, Item Toy, Product ID T3243H, Name "Terrifying Tarantula", Quantity
17
Order 105, Could not process order data was corrupted, InvalidDataError -
Stuffing can only by "Polyester Fiberfill" or "Wool"
```

Concluding Thoughts

Remember to approach your code in an object-oriented fashion. Don't just start bashing out code and hoping that it will all work out. Take a planned approach and consider these steps.

1. Identify your Product Families and Variations. This is key to getting the Abstract Factory Pattern down.
2. Draw the Product Hierarchies and the Factory Hierarchies.
3. Identify any other classes and objects you will need. Figure out what each class is responsible for.
4. Draw a UML class diagram showing how the classes/objects relate to each other. Write down the attributes and methods. o At this stage you can do a simple sketch on paper. Don't worry about syntax and only mention the important attributes/methods.
5. Write some code
6. Repeat from step 3. (Take an iterative approach!)

When creating your **final** UML diagram, be sure to check syntax, and mention all the attributes and methods.

After you submit your Lab to GitHub, send the marker a private message on Discord telling them you finished, along with your student number, gitName, and collaboration url.

• ie: "I uploaded my work to gitHub. My name is *your name* (*your student*) and my partner's name is *partner name*(*partner student number*) and my gitName is YYYYY, my git collaboration url is: ZZZZZZZZZZZZZZZ"