Jason Walker

Jason.walker@blackpool.ac.uk

App structure

Warehouse Application

Contents

[RegularItems.py 2](#_Toc182578848)

[Step 1: Import the Base Class 2](#_Toc182578849)

[Step 2: Define the RegularItem Class 2](#_Toc182578850)

[Step 3: Define the PerishableItem Class 3](#_Toc182578851)

[Key Points to Remember 3](#_Toc182578852)

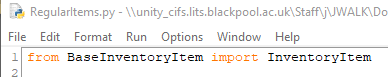
[Integration 3](#_Toc182578853)

# RegularItems.py

This document provides an in-depth explanation of the RegularItems script, detailing the functionality of the 'RegularItem' and 'PerishableItem' classes.

# Step 1: Import the Base Class

The script begins by importing the `InventoryItem` class from the BaseInventoryItem module. This ensures that both `RegularItem` and `PerishableItem` can inherit the structure and behaviors of the base class.

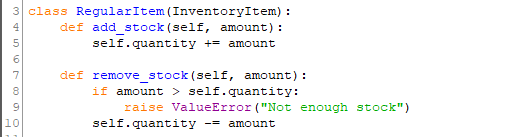


Explanation:

1. `from BaseInventoryItem import InventoryItem`: This imports the base class defined in the BaseInventoryItem.py file.

# Step 2: Define the RegularItem Class

The `RegularItem` class inherits from `InventoryItem` and provides specific implementations for adding and removing stock.



Explanation:

1. `class RegularItem(InventoryItem)`: Declares the `RegularItem` class as a subclass of `InventoryItem`.

2. `add\_stock(self, amount)`: Implements the method to increase the `quantity` attribute by the specified amount.

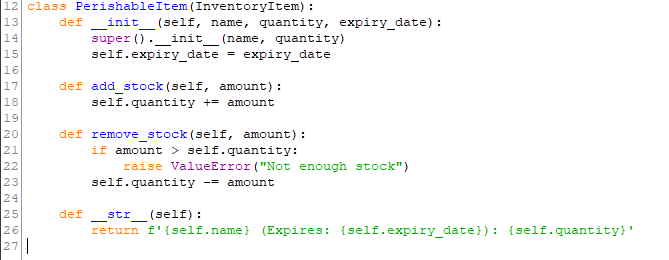
3. `remove\_stock(self, amount)`: Implements the method to reduce the `quantity` attribute:

- If the amount exceeds the current stock, raises a `ValueError` with the message 'Not enough stock'.

- Otherwise, reduces the `quantity` by the specified amount.

# Step 3: Define the PerishableItem Class

The `PerishableItem` class also inherits from `InventoryItem`. In addition to the attributes of the base class, it introduces an expiry date.



Explanation:

1. `def \_\_init\_\_(self, name, quantity, expiry\_date)`: Initializes the `PerishableItem` object with an additional `expiry\_date` attribute.

2. `super().\_\_init\_\_(name, quantity)`: Calls the constructor of the parent class to initialize the `name` and `quantity` attributes.

3. `add\_stock` and `remove\_stock`: These methods are implemented similarly to the `RegularItem` class.

4. `\_\_str\_\_(self)`: Overrides the string representation method to include the expiry date. The format is 'ItemName (Expires: Date): Quantity'.

# Key Points to Remember

1. `RegularItem` and `PerishableItem` both inherit the structure and behaviors of the `InventoryItem` base class.

2. The `PerishableItem` class introduces an additional attribute, `expiry\_date`, to handle perishable goods.

3. Both classes implement `add\_stock` and `remove\_stock` to manage inventory operations, with appropriate error handling.

# Integration

These classes can be used to create specific types of inventory items within the application. For example:

1. Use `RegularItem` for non-perishable goods like electronics or furniture.