Project Report:

Critical Thinking 2 – RPG Bag V2

Alejandro Ricciardi

Colorado State University Global

CSC400: Data Structures and Algorithms

Professor: Hubert Pensado

August 25, 2024

Project Report:

Critical Thinking 2 – RPG Bag V2

This documentation is part of the Critical Thinking 2 Assignment from CSC400: Data Structures and Algorithms at Colorado State University Global. This Project Report is an overview of the program's functionality and testing scenarios including console output screenshots. The program is coded in Java JDK-22 and named Critical Thinking 2 (RPG Bag V2).

The Assignment Direction:

Additional Bag Methods

In this second assignment, you will extend the functionality of the Java bag data structure by implementing additional methods. You will add methods for calculating the size of the bag, merging two bags together, and finding the distinct elements in the bag.

1. Modify the `Bag` class from the previous assignment to include the following additional methods:
    - `int size()`: This method should return the total number of elements in the bag, including duplicates.
    - `void merge(Bag<T> otherBag)`: This method should merge the elements of `otherBag` into the current bag.
    - `Bag<T> distinct()`: This method should return a new bag that contains only the distinct elements from the current bag.
2. Write a Java program that demonstrates the usage of the additional methods. Your program should perform the following operations:
    - Create two instances of the `Bag` class.
    - Add elements to each bag, including duplicates.
    - Print the size of each bag using the `size` method.
    - Merge the two bags together using the `merge` method.
    - Print the merged bag contents.
    - Create a new bag containing only the distinct elements using the `distinct` method.
    - Print the distinct bag contents.

Submit your completed assignment as a .java source code file.

⚠️ **My notes:**

- The program implements an Item class that acts as the base class for the classes Potion, Armor, and Weapon.

- The class Bag implements the Iterable interface and uses a linked list,
  [element | next] -> [element | next] -> [element | next] -> null, to store elements.

- A popular implementation of the Bag ADT is to use a HashMap. Although a HashMap does not allow duplicate entries, it can store a single entry for each element along with its current count. This would eliminate the need to iterate through the entire Bag to count specific elements. However, for this assignment, I chose to use a linked list structure to show a more traditional approach to implementing a Bag ADT.

- In addition to the required functionalities, I added an item ID system that ensures bag elements of the same type are not flagged as duplicates. For instance, two healing potions with different IDs will not be considered duplicates; however, two healing potions with the same ID will be considered duplicates.
  Furthermore, the Bag class contains, count, and remove methods treat item objects of the same type but with different IDs as the same type of element. For instance, two healing potions with different IDs will be treated as the same object by these methods.
  Both these two functionalities were accomplished by overring the equals() from the Java Object Class in the Item class, specifically in its subclasses: Potion, Armor, and Weapon.

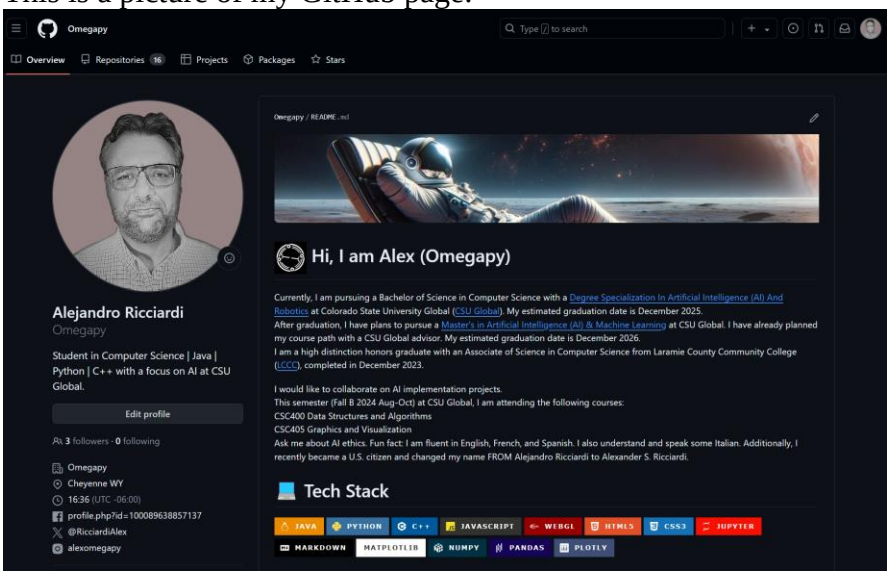
- The program source code can be found in the following files:
  - Item.java
  - Armor.java
  - Weapon.java
  - Potion.java
  - Bag.java
  - Main.java

My Program Description:

The program is an implementation of a Bag Abstract Data Structure (Bag ADT) using a Linked list structure.
[element | next] -> [element | next] -> [element | next] -> null.
The Bag class represents the inventory of an RPG video game player.
The Bag allows for the storage and management of game items such as Potions, Armor, and Weapons.
The Bag ADT is implemented as a generic class that can store any element object type.

Git Repository

This is a picture of my GitHub page:

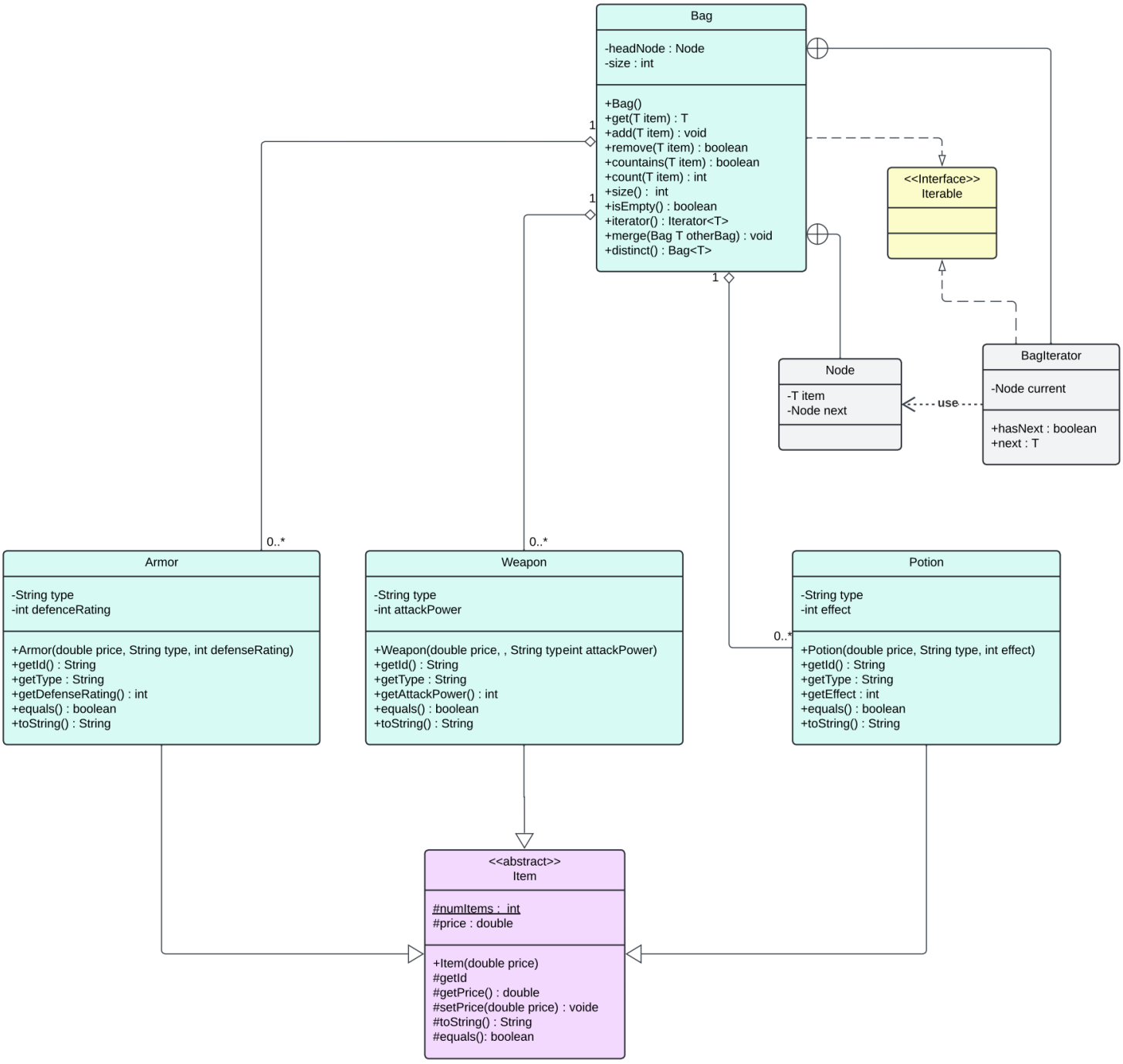


I use GitHub as my Distributed Version Control System (DVCS), the following is a link to my GitHub, Omegapy.
My GitHub repository that is used to store this assignment is named My-Academics-Portfolio and the link to this specific assignment is: https://github.com/Omegapy/My-Academics-Portfolio/tree/main/Data-Structures-and-Algorithms-CSC400/Critical-Thinking-2

Classes Description:

- **Armor Class**
  It extends the Item Class. It represents an armor item in the RPG game.
- **Weapon Class**
  It extends the Item Class. It represents a weapon item in the RPG game.
- **Potion Class**
  It extends the Item Class. It represents a potion item in the RPG game.
- **Weapon Class**
  This Class is the base class for all item types, such as Potion, Armor, and Weapon. It represents a generic item in the RPG game.
- **Item Class**
  This abstract Class is the base class for all item types, such as Potion, Armor, and Weapon. It represents a generic item in the RPG game.
  Bag Class
  A Bag ADT class. This class implements a bag using a list structure,
  [element | next] -> [element | next] -> [element | next] -> null, to store elements.
- **Main Class**
  Runs tests on the functionality of the Bag data structure.

Figure 1
UML Class Diagram



Screenshots

Figure 2

Initial Bag Inventory

```
        ********************
        *     RPG BAG V2     *
        ********************

Size of Bag 1: 6
Size of Bag 2: 4

Bag 1
Bag contents:
Armor [ID: A4, type: Iron, defenseRating = 50, price = 75.0]
Potion [ID: P2, type: Strength, effect = 50, price = 10.5]
Potion [ID: P8, type: Strength, effect = 50, price = 10.5]
Potion [ID: P1, type: Healing, effect = 50, price = 10.5]
Potion [ID: P1, type: Healing, effect = 50, price = 10.5]
Potion [ID: P7, type: Healing, effect = 50, price = 10.5]

Bag 2
Bag contents:
Weapon [ID: W5, type: Sword, attackPower = 40, price = 150.0]
Potion [ID: P6, type: Protection, effect = 50, price = 10.5]
Potion [ID: P2, type: Strength, effect = 50, price = 10.5]
Potion [ID: P1, type: Healing, effect = 50, price = 10.5]
```

Figure 3

Merging Bags

```
Merged Bag 1 and Bag 2 into Bag 1

Size of the merged Bag 1: 10

Bag contents:
Potion [ID: P1, type: Healing, effect = 50, price = 10.5]
Potion [ID: P2, type: Strength, effect = 50, price = 10.5]
Potion [ID: P6, type: Protection, effect = 50, price = 10.5]
Weapon [ID: W5, type: Sword, attackPower = 40, price = 150.0]
Armor [ID: A4, type: Iron, defenseRating = 50, price = 75.0]
Potion [ID: P2, type: Strength, effect = 50, price = 10.5]
Potion [ID: P8, type: Strength, effect = 50, price = 10.5]
Potion [ID: P1, type: Healing, effect = 50, price = 10.5]
Potion [ID: P1, type: Healing, effect = 50, price = 10.5]
Potion [ID: P7, type: Healing, effect = 50, price = 10.5]

Testing contains method:
Bag contains Healing Potions: true
Bag contains Strength Potions: true
Bag contains Potection Potions: true
Bag contains Iron Armors: true
Bag contains Swords: true

Testing count method:
Count of Healing Potions: 4
Count of Strength Potions: 3
Count of Potection Potions: 1
Count Swors: 1
Count of Iron Armors: 1
```

Figure 4

Deleting Items

```
-------------------------------------------------------

Deleting Healing Potion!

A Healing Potion was removed successfully!
Count of Healing Potions after removal: 3
Bag contents:
Potion [ID: P2, type: Strength, effect = 50, price = 10.5]
Potion [ID: P6, type: Protection, effect = 50, price = 10.5]
Weapon [ID: W5, type: Sword, attackPower = 40, price = 150.0]
Armor [ID: A4, type: Iron, defenseRating = 50, price = 75.0]
Potion [ID: P2, type: Strength, effect = 50, price = 10.5]
Potion [ID: P8, type: Strength, effect = 50, price = 10.5]
Potion [ID: P1, type: Healing, effect = 50, price = 10.5]
Potion [ID: P1, type: Healing, effect = 50, price = 10.5]
Potion [ID: P7, type: Healing, effect = 50, price = 10.5]

Size of Bag 1 after deletion: 9

-------------------------------------------------------

Deleting second Healing Potion!

Another Healing Potion was removed successfully!
Count of Healing Potions after removal: 2
Bag contents:
Potion [ID: P2, type: Strength, effect = 50, price = 10.5]
Potion [ID: P6, type: Protection, effect = 50, price = 10.5]
Weapon [ID: W5, type: Sword, attackPower = 40, price = 150.0]
Armor [ID: A4, type: Iron, defenseRating = 50, price = 75.0]
Potion [ID: P2, type: Strength, effect = 50, price = 10.5]
Potion [ID: P8, type: Strength, effect = 50, price = 10.5]
Potion [ID: P1, type: Healing, effect = 50, price = 10.5]
Potion [ID: P7, type: Healing, effect = 50, price = 10.5]
Size of Bag 1 after deletion: 8

Testing contains method after Healing Potions removal:

Testing contains method:
Bag contains Healing Potions: true
Bag contains Strength Potions: true
Bag contains Potection Potions: true
Bag contains Iron Armors: true
Bag contains Swords: true

Testing count method:
Count of Healing Potions: 2
Count of Strength Potions: 3
Count of Potection Potions: 1
Count Swors: 1
Count of Iron Armors: 1

-------------------------------------------------------
```

Figure 5

Final Count Check

```
Distinct Bag Contents

Size of distinct Bag: 7
Bag contents:
Potion [ID: P7, type: Healing, effect = 50, price = 10.5]
Potion [ID: P1, type: Healing, effect = 50, price = 10.5]
Potion [ID: P8, type: Strength, effect = 50, price = 10.5]
Armor [ID: A4, type: Iron, defenseRating = 50, price = 75.0]
Weapon [ID: W5, type: Sword, attackPower = 40, price = 150.0]
Potion [ID: P6, type: Protection, effect = 50, price = 10.5]
Potion [ID: P2, type: Strength, effect = 50, price = 10.5]

Testing contains method after duplicates renoval removal:

Testing contains method:
Bag contains Healing Potions: true
Bag contains Strength Potions: true
Bag contains Potection Potions: true
Bag contains Iron Armors: true
Bag contains Swords: true

Testing count method:
Count of Healing Potions: 2
Count of Strength Potions: 2
Count of Potection Potions: 1
Count Swors: 1
Count of Iron Armors: 1
```

Note that potion items of the same type but with different IDs are not considered duplicates; however, those of the same type and with the same IDs are.

As shown in Figure 2 through Figure 5 the program runs without any issues displaying the correct outputs as expected.