TWD Linked List

Steven On

CPT 307: Data Structures & Algorithms

Professor Joshua Reichard

October 1, 2019

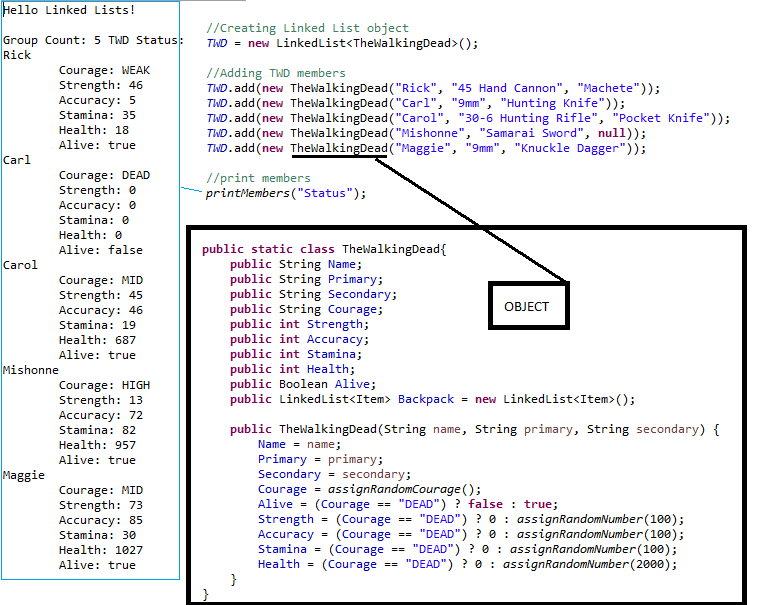
TWD Linked List

A Java LinkedList is an abstract data type (ADT) that can be used as a container of various objects. A LinkedList may contain but not limited to the following functions:

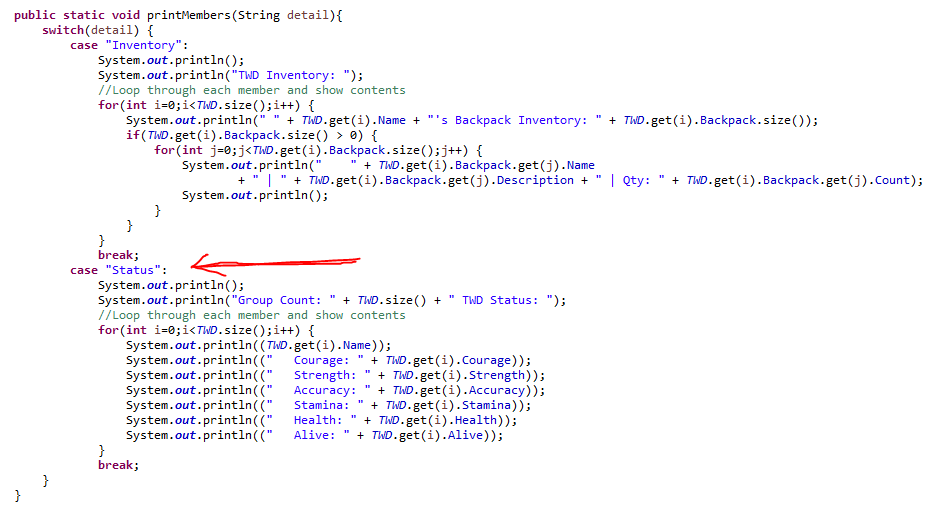
* add(): add an item to the list.
* remove(): remove an item from the list.
* get(): display an item from the list.
* size(): count of items in the list.

I used the functions above to create a program that’s “The Walking Dead” themed. Below, you will find the execution steps on how I used the LinkedList’s “add” function:

1. Instantiate a new LinkedList object referencing the data type as my custom object “The WalkingDead”

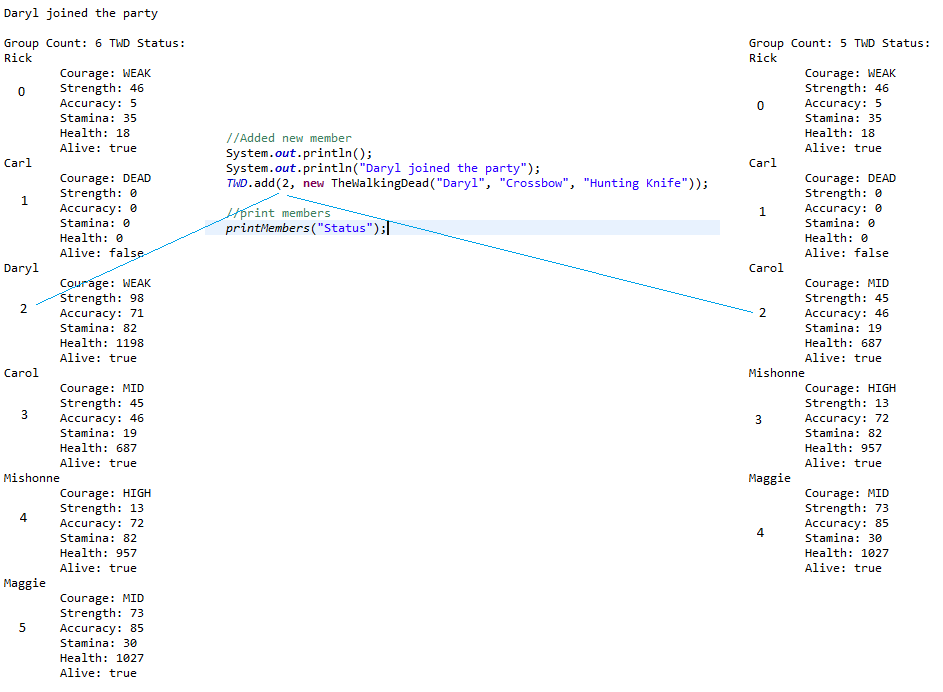


1. I added 5 “TheWalkingDead” objects to the LinkedList. By using the add method. Since my object contained a constructor, I was able to add to the linked list by passing in a new “TheWalkingDead” object with the following three properties: “Name”, “Primary”, and “Secondary”. The object’s constructor filled in the remaining details for the character using the Random Java library and conditions based on the “Courage” property. I learned how to use a one line if statement from Harold, E.R. (2017).
2. Here’s the screenshot for the “printMembers(“Status”)” method. I’m starting to notice a lot of similarities between C# and Java such as the switch statement. The syntax for a switch is the same. I used a switch to perform two different types of prints to the console. Passing the “Status” to the “printMembers” function will print all members stats which include the following properties: Name, Courage, Strength, Accuracy, Stamina, Health, and Alive.

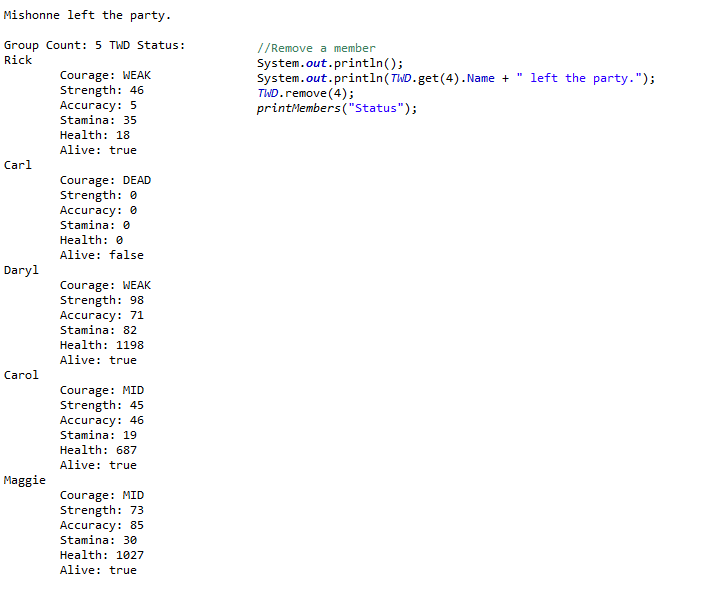


Below, you will find the execution steps on how I used the LinkedList’s “add” function to insert “Daryl” at index 2 and the “remove” function to remove “Mishonne” from index 4:

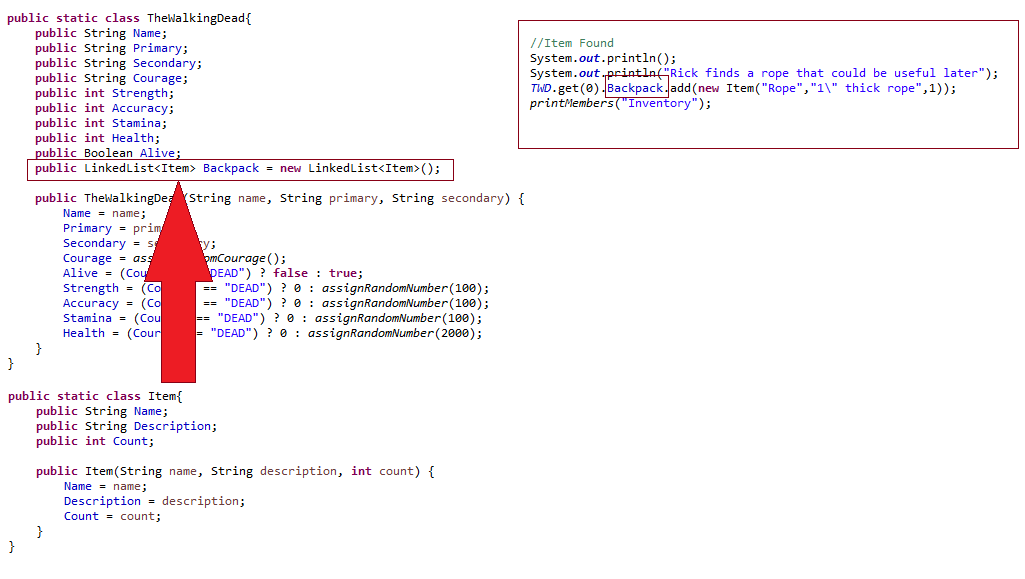
Carol, currently sits on index 2, using the LinkedList add method and passing in the index, along with the object allows for the new object to be inserted in a specific location of the list. Daryl is now sitting at index 2 and Carol has been bumped along with the rest of the group that precedes Carol by 1 index.



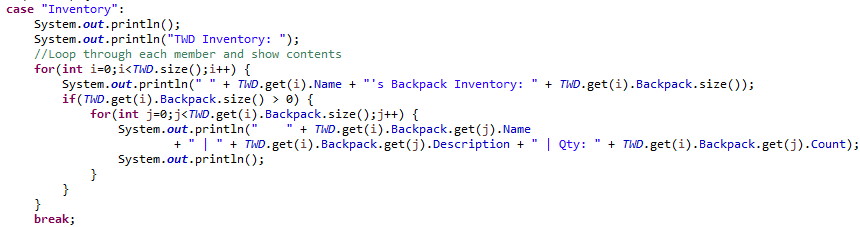
Using the LinkedList “remove” function and passing the index of 4, Mishonne is removed from the list.



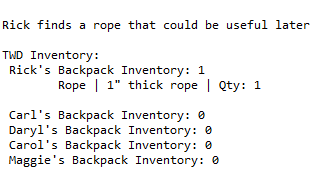
That was all that was asked for this assignment, but I got curious to find out how deep of a level I can go with an object and a LinkedList. I created a Backpack LinkedList property to allow the characters to store items they have found.



In this example above, the right shows the Java code and tells the console that our character Rick has found a rope and added it to his inventory. The snapshot below shows the code on how the inventory is displayed:



All ADTs contain a “size” method which allows the developer to loop through the list and display items from the list. For this example, I have two loops one to loop through the first LinkedList to retrieve the name of each character and another loop to loop through items in the character’s backpack.



This was a pretty fun exercise of the use of the LinkedList ADTs. One day. I’ll comeback to this program and create a command-line based game to walk our heroes through an adventure.

References

Harold, E.R. (Dec. 14, 1997) The ? : operator in Java. *Cafeaulait.org.* Retrieved from <http://www.cafeaulait.org/course/week2/43.html>

tutorialspoint. (n.d.). Java – The LinkedList Class. *tutorialspoint*. Retrieved from <http://www.tutorialspoint.com/java/java_linkedlist_class.htm>