Assignment Prefix: lab03

Due Date: Friday, September. 16th @ 11:59pm

Points: 100

This is an individual assignment.

Restrictions: you cannot use any methods from the Java Array(s) class to copy an array, check for equality, or otherwise manipulate an array. You must write the Java code to perform these functions.

Create a NetBeans project named Lab03 and ensure it is saved to a location like desktop or your flash drive. In the project you will do the following:

In this assignment you will be creating a generic version of ***Bag*** which would be a collection of items that the client would define. The ***Bag*** interface will be a ***Generic interface*** with a ***Type Parameter***. Just like in Lab02, ***Bag*** will not order the items in any particular order nor does it prevent any duplicates.

Develop a Generic interface named ***Bag*** that can store certain number of items (type will be specified by the client). Provide the following methods in the interface.

- A method that returns the current count of items in the bag
- A method that checks if the bag is empty
- A method that checks if the bag is full
- A method that adds an item to the bag and returns true if the item was added successfully or false if the item was not added.
- A method that removes a random item from the bag as long as the bag is not empty. This method would return the removed item from the bag.
- A method that removes a specific item from bag. This method will take as parameter the item to be removed, find the first occurrence of the item. Finally, the method returns true if the removal was successful else false.
- A method that removes all the items from the bag
- A method that returns the count of occurrences of a specific item in the bag.
- A method that checks if an item exists in the bag.

Design a Generic class called ***ArrayBag*** which will implement the Generic ***Bag*** Interface created earlier. This class will include a Type Parameter as part of class declaration. The client of ***ArrayBag*** will be able to specify the actual type.

- Declare an instance variable ***bag*** – an array of ***Generic*** type: This structure will hold the items in the bag.
- Declare another instance variable ***count***: This will provide the count of items currently stored in the bag. This count will increment as a new item is added to the bag and decrement as an item is removed from the bag.
- Provide a default constructor that will initialize the instance variable bag to a new array of length 1.
- Provide an overloaded constructor that allows the client to specify the initial capacity of the bag.
- Implement the methods received from the interface.
- The method that adds an item to the bag should check if the bag is full. If the bag is full then it should automatically double the capacity of the bag and add the item.
- The method that removes a specific item which is passed as a parameter should use the ***equals( )*** method to compare the contents of objects in the bag with the contents of the parameter. If there is an object in the bag with the same contents then, it removes that item from the bag and returns true or else false if there is no item with the same contents.
- The methods that remove items (randomly or specified item) should automatically allow the items to shift to replace the removed item.
- Implement the following additional methods
    - A method that returns an item at a specific index position in the bag.
    - A method that returns the current capacity of the bag
    - A method that returns the copy of the bag array instance variable

Create a user-defined class called ***Player***. Each Player object will have the following attributes (instance variables): ***name, position played***, and ***jersey number***. Use appropriate data types to define these instance variables and use recommended naming conventions for the variable names. Provide a constructor, implement the accessor and mutator methods for each of the instance variables, and include the toString( ) and equals( ) methods.

Finally, create a Client Program ***NDSU-BasketBall*** with the ***main( )*** method. Inside the main method do the following:

- Create an object of ***ArrayBag*** called ***team*** to store all players' information of NDSU Men's or Women's Basket Ball team.
- Run a ***for loop*** to prompt the user for each Player's information, create the Player object and finally add the player to the team.
- Remove a random player from the team.
- Add a new Player with some made up information.
- Display the current count of players in the team.
- Remove the Player that you just added earlier with made up information from the team using appropriate method.
- Display the current count of players in the team.
- Print the Players in the team.
- To demonstrate that your generic class can support objects of different types:
    - Create an object of ***ArrayBag*** called courses to store the course ids of the courses that you are taking this semester (CSci 161, …..) as Strings.
    - Populate the bag with the course ids.
    - Remove a random course id from the Bag.
    - Print the contents of the bag.

Comment your ***Bag*** interface, ***ArrayBag*** and ***Player*** classes with Java Doc commenting style. Use single line or block style comments for the ***client*** program.

Using Visio draw a ***UML class diagram*** displaying all the classes and the relationships between them.

For this diagram **include the client class**. The contents of the client class can just be the ***main( )*** method. This class will have association relationships between ***ArrayBag*** and ***Player*** class.

An association relationship simply means that a class uses another class – In this project, Client class is using ***ArrayBag*** and ***Player*** classes. This is depicted by an undirected solid line connecting the two classes.

Things to turn in:

- Open a Microsoft Word document name using the following file naming convention
    - i.e. *lab03-LnameFM*
        - lab03 = assignment prefix
        - Lname = your last name
        - F = your first initial
        - M = your second initial
- Copy and Paste the source code of the ***Bag*** Interface (make sure to use *Ctrl + A* to select all the source code of the program and *Ctrl + C* to copy).
- Copy and Paste the source code of the ***ArrayBag and Player*** class.
- Copy and Paste the source code of the client program – ***NDSU-BasketBall***
- Copy and paste the output of the client program
- Create a screen capture of your NetBeans IDE that includes the contents of the Output Window and paste it into your Word document below your source code.
    - To create a screen capture of your NetBeans IDE
        - Select, left-click in the NetBeans IDE
        - Use Alt-PrintScreen to place an screen capture image on the clipboard
        - Use Ctrl-V to paste the contents of the clipboard into your Word document
- Copy and paste the ***UML class diagram***
- Next, zip the Project folder.
- Finally on blackboard, submit both your Word document and project zipped file.