Java Programming

3-2: Collections – Part 1

Practice Activities

Lesson Objectives:

- Create a collection without using generics
- Create a collection using generics
- Implement an ArrayList
- Implement a Set

Vocabulary:

Identify the vocabulary word for each definition below.

Set	A set similar to an ArrayList without any specific ordering.
ArrayList	An ordered Collection that may contain duplicates.
Collection	An interface used to define a group of objects. This includes lists and sets.
ArrayList	A list that is very similar to an array.
Set	A Collection of elements that does not contain any duplicates.

Try It/Solve It:

1. This activity uses the **javabank** project, open it in the IDE. You are going to update the code to use an ArrayList instead of a static one dimensional array.

    a. Open the javabank.java file.

    b. Find the line that creates the static one-dimensional array named myAccounts

```
static AbstractBankAccount myAccounts[]  = new AbstractBankAccount[MAXACCOUNTS];
```

    c. Comment out that line and create an ArrayList of AbstractBankAccount named myAccounts directly underneath it.

```
//static AbstractBankAccount myAccounts[]  = new AbstractBankAccount[MAXACCOUNTS];
static ArrayList<AbstractBankAccount> myAccounts =
                                            new ArrayList<AbstractBankAccount>();
```

    d. Identify all of the errors that now appear in the code and update them using the add() or get() methods to work with the ArrayList.

```
myAccounts[noAccounts] = new CreditAccount(name,accountNum,balance);
```

becomes

```
myAccounts.add(new CreditAccount(name,accountNum,balance));
```

```
myAccounts[i].setBalance(myAccounts[i].getBalance()+deposit);
```

becomes

```
myAccounts.get(i).setBalance(myAccounts.get(i).getBalance()+deposit);
```

e. Run and test the java application by creating multiple accounts of various types and displaying them to the textArea.

2. The following activity uses the **bikeproject**, open it in the IDE.

   a. Create a new driver class named BikeList that contains a main method.

   b. You will be required to store both Mountain and Road bikes so create an ArrayList named bikes that is based on the Superclass.

   c. Add two integer local variables named mountainBikeSales and roadBikeSales that are both initialized to zero.

   d. Create a fillArray method that takes the bikes ArrayList as a parameter. The method should generate a random number between 0 and 1. If the random number is less than one then add a mountain bike to the list otherwise add a roadbike. The method should add 10 bikes to the ArrayList.

   e. Call the fillArray method from amin under the local variabl;e declarations.

   f. Create a displayStock method that takes the bikes ArrayList as a parameter. Use an enhanced for to display the bikes to the console.

   g. Call the displayStock() method from main under the fillArray() method call.

   h. Create a method named calculateStock that takes the bikes ArrayList as a parameter and returns an int value. Declare a local int variable named bikesSold that is initialized to zero and will hold the number of mountain bikes in stock. Use an enhanced for loop to go through the bikes list. Use an if statement to check if each bike in the array is an instance of a mountain bike, increment the bikesSold variable if the if statement returns true. Return the value held in bikesSold.

   i. Create a method named displayBikeNumbers that takes the bikes ArrayList as a parameter and returns an int value. Declare two local int variables named mb(mountain bikes) and rb (road bikes). Get the value for the mb variable by calling the calculateStock() method. Get the value of rb by subtracting the number of mountain bikes from the number of bikes in the ArrayList. Your ouptput should look like the following:

```
Stock Levels
We have 4 Mountain Bikes in stock
We have 6 Road Bikes in stock
```

   j. Call the displayBikeNumbers() method as the last method call in main.

3. What is the difference between a set and a list? A List represents an ordered sequence of objects.

   A Set is a collection that cannot contain duplicate elements.

4. You decide you want to roll 2 dice and see what the frequency is of each possible number combination. Would you use a Set collection to do this? State your reason(s).

   ArrayList. Set cannot contain duplicate elements

5. Using a collection create a variable that will store a list of countries (Strings). Your collection should not store duplicates, and order is not important. Test your code by adding 6 countries, one of which is a duplicate.

   The duplicate is overwritten

6. Would the following Collection.sort() statements both work? Explain your answer.

```
HashSet<String> countriesSet = new HashSet<String>();
Collections.sort(countriesSet);
ArrayList<String> countriesList = new ArrayList();
Collections.sort(countriesList);
```