

CSC 2331 Data Structures

Assignment 2: Bags (max 60 points)

Due: 02/10/2020, 11:30 am

1. Problem 1 (30 points): **Bag Client**. Write a client class `ConsonantsFilter` that filters out all consonants from a bag containing letters. To accomplish this:
 - a. Create three **bags** that hold strings. One bag is named `letters` and contains several one-letter strings. Another bag is named `vowels` and contain five strings, one for each vowel. The third bag is empty and is named `consonants`.
 - b. One at a time
 - i. Remove a string from the bag `letters`.
 - ii. Check whether the string is in the bag `vowels`. *Hint*: look for an appropriate method in `BagInterface` that you can use.
 - iii. If it is not, place it into the bag `consonants`.
 - c. After you have checked all of the strings in the bag `letters`, report the number of consonants in the bag.
2. Problem 2 (30 points): **Bag Implementation**: In mathematics, the concepts of bags and sets are closely related. The difference is that bags allow duplicate items, while a set does not. An operation that removes the duplicates from a bag would be helpful if one wished to implement a set. Write a method in the `ArrayBag` class that removes the duplicates of all items in a bag:

```
public void removeDuplicates() {...}
```

Write also a client class `TestArrayBag` to test the `removeDuplicates()` method.

Hints for possible solutions of Problem 2:

Variant 1:

In order to remove the duplicates from the bag you can use a **pair of nested loops**. Both loops scan the `bag` array (an instance variable in the `ArrayBag` class):

1. Write an **outer loop** to visit the position of each item in the array `bag`.
2. The **inner loop** shall remove the duplicates that come **after** the currently considered element. You can use a **while loop** to scan over the remaining items in the array: if the index of the current element is 'i', remove the duplicates from position 'i+1' to the last element saved in the array.
 - a. To remove an item, use the same technique we used for the modification of the `remove()` method:
 - i. Copy the last item over the item to be removed.
 - ii. Replace the last item with null.

Variant 2:

Use a second array (e.g. `bag2`), where you copy only the elements from the `bag` array that do not appear already there.

Here is a scheleton of the method:

```
public void removeAllDuplicates() {  
    @SuppressWarnings("unchecked")  
    T[] bag2 = (T[]) new Object[numberOfEntries]; // unchecked cast  
    int sizeBag2 = 0; // number of elements in bag2  
  
    // First loop: visit each element of the array bag  
  
        // Inner loop: check if the current item from bag is already in the second array, bag2,  
        // and if not, store it in bag2  
        .....  
  
    // Save back the elements of bag2 in the original array (instance variable) bag  
    ....  
}
```

What to turn in: The classes `ConsonantsFilter.java`, `ArrayBag.java`, and `TestArrayBag.java`.

Important Note: Differently from the labs, the assignments are **strictly individual**. You are not allowed to use somebody's code, or to present code found on the Web. The assignments which do not comply with these constraints will get zero points (both sides).