Richelin Metellus

CSCI 161\_CompSci II

Dr. Latimer

Jan 27, 2017

**Bag Interface.**

/\*\*

\* @author Richelin Metellus.

\* @version 01/26/2017

\* Bag consist of all the methods to append, clear, get, remove

\* item from a list.

\*/

public interface Bag {

/\*\*

\* returns a count of numbers in the bag

\* @return size of the bag

\*/

public int getCurrentSize();

/\*\*

\* check if bag is empty

\* @return true when it's empty.

\*/

public boolean isEmpty();

/\*\*

\* add num to the bag

\* @param num

\*/

public void add(int num);

/\*\*

\* remove a randomly selected num from the bag.

\*/

public void remove();

/\*\*

\*remove the first occurrence of the number num

\* @param num

\*/

public void remove (int num);

/\*\*

\*clear the entire content of the list

\*/

public void clear();

/\*\*

\*

\* @param num

\* @return the frequence of occurrent of the argument

\*/

public int getFrequencyOf(int num);

/\*\*

\*

\* @param num

\* @return true when the argument is present in the list.

\*/

public boolean contains(int num);

/\*\*

\*

\* @return the contents of the list

\*/

@Override

public String toString();

/\*\*

\*

\* @param o Object

\* @return true if equals to the class defined.

\*/

@Override

public boolean equals(Object o);

}

**Score Class**

/\*\*

\* @author Richelin Metellus

\* @version 01/26/2017

\* Class Scores provide implementation for the methods

\* inherited for the Bag interface.

\*/

import java.util.Random;

public class Scores implements Bag {

int[] list;

int count;

public Scores(){ // default constructor

list = new int[50];

}

public Scores(int size)

{

list = new int[size];

}

@Override

public int getCurrentSize()

{

return count;

}

@Override

public boolean isEmpty()

{

if( count == 0)

return true;

else

return false;

}

@Override

public void clear()

{

this.count = 0;

}

@Override

public void add(int num)

{

if (count < list.length)

{

list[count]= num;

count++;

}

else if(count == list.length)

{

int temp[] = new int[list.length \* 2];

int i;

for(i =0; i < count; i++)

temp[i]= list[i];

temp[i] = num; // i is going to next open slot since it is incremented.

count++;

list = temp;

temp = null;

}

}

@Override

public int getFrequencyOf(int num)

{

int numOccurrence = 0;

for(int i = 0; i < count ;i++)

{

if(list[i]== num)

numOccurrence++;

}

return numOccurrence;

}

@Override

public boolean contains(int num)

{

boolean flag = false;

for (int i= 0; i < count; ++i)

{

if(list[i]== num)

{

flag = true;

break;

}

}

return flag;

}

@Override

public void remove()

{

Random rand = new Random();

int randomIndex = rand.nextInt(count);

remove(list[randomIndex]);

}

@Override

public void remove(int num){

for (int i = 0; i < count; i++)

{

if (list[i] == num) // once num is found, value of i stay fix.

{

for( int j = i; j < count; j++ ) // j now takes over to shift left the elements on the right 0f index i.

list[j]= list[j+1]; //thus replacing/removing the value at index i.

count--;

return;

}

}

}

public int get(int index) throws ArrayIndexOutOfBoundsException{

int numAtIndex;

if (index < 0 || index >= count )

throw new ArrayIndexOutOfBoundsException(" Invalid:The Index not in between 0<=" + index +" <=" + count);

numAtIndex = list[index];

return numAtIndex;

}

@Override

public String toString()

{

String listContent = "Bag: [";

for (int i=0; i < count; ++i)

{

listContent = listContent + list[i] + " ";

}

return listContent;

}

@Override

public boolean equals(Object o)

{

if(!(o instanceof Scores))

return false;

Scores a = (Scores)o;

if (a.getCurrentSize() != count)

return false;

for (int i = 0; i< count; i++)

{

if(a.get(i) !=list[i])

return false;

}

return true;

}

}

**Client Class**

/\*\*

\*

\* @author Richelin Metellus

\* @version 01/27/2017

\* Client test the methods defined in the Bag interface.

\*/

import java.util.Random;

public class Client {

public static void main(String[] args) {

System.out.println("Entering Main");

Random rand= new Random();

Scores cart = new Scores(100);

int num;

for(int i = 0; i< 100; i++)

{

cart.list[i]= rand.nextInt(201)-100;

cart.count++;

}

System.out.println("Printing numbers in The " + cart +"]" +

"\nAdding 86 to the list");

cart.add(86);

System.out.println("86 is successfully appended to the end of list\n" + cart

+ "]\nList Current size: " + cart.getCurrentSize());

cart.remove();

System.out.println("Random number in the list successfully removed\n" + cart+

"] \nCurrent size: " + cart.getCurrentSize());

num = cart.get(75);

System.out.println("Frequency of " + num + " is: "+ cart.getFrequencyOf(num));

cart.remove(num);

System.out.println("Frequency of " + num + " after removal of its 1st occurrence: "

+ cart.getFrequencyOf(num));

System.out.println("Frequency of the number 86 in the bag: " + cart.getFrequencyOf(86));

System.out.println("Checking if the list contain 86: " + cart.contains(86));

}

}

**Program Outpout**

run:

Entering Main

Printing numbers in The Bag: [29 -17 84 -32 -70 -17 14 65 -71 -11 -40 -12 -85 44 -72 -53 -4 78 -76 -55 -89 8 80 -23 -91 25 12 -7 25 18 36 100 86 91 6 -41 51 -99 61 64 61 49 1 -59 28 -35 34 -29 73 57 -12 74 87 71 85 -8 59 94 -67 -64 -62 78 5 -42 -43 23 97 48 87 73 -26 -62 10 35 76 25 57 22 1 65 -19 -47 55 -18 -20 72 -96 -99 -1 -95 12 -82 43 -65 70 37 -42 -92 -32 -57 ]

Adding 86 to the list

86 is successfully appended to the end of list

Bag: [29 -17 84 -32 -70 -17 14 65 -71 -11 -40 -12 -85 44 -72 -53 -4 78 -76 -55 -89 8 80 -23 -91 25 12 -7 25 18 36 100 86 91 6 -41 51 -99 61 64 61 49 1 -59 28 -35 34 -29 73 57 -12 74 87 71 85 -8 59 94 -67 -64 -62 78 5 -42 -43 23 97 48 87 73 -26 -62 10 35 76 25 57 22 1 65 -19 -47 55 -18 -20 72 -96 -99 -1 -95 12 -82 43 -65 70 37 -42 -92 -32 -57 86 ]

List Current size: 101

Random number in the list successfully removed

Bag: [29 -17 84 -32 -70 -17 14 65 -71 -11 -40 -12 -85 44 -72 -53 -4 78 -76 -55 -89 80 -23 -91 25 12 -7 25 18 36 100 86 91 6 -41 51 -99 61 64 61 49 1 -59 28 -35 34 -29 73 57 -12 74 87 71 85 -8 59 94 -67 -64 -62 78 5 -42 -43 23 97 48 87 73 -26 -62 10 35 76 25 57 22 1 65 -19 -47 55 -18 -20 72 -96 -99 -1 -95 12 -82 43 -65 70 37 -42 -92 -32 -57 86 ]



Current size: 100



Frequency of 57 is: 2

Frequency of 57 after removal of its 1st occurrence: 1

Frequency of the number 86 in the bag: 2

Checking if the list contain 86: true

BUILD SUCCESSFUL (total time: 0 seconds)