/\*\*

\*

\* @author Steven Glasford

\*

\* The interface file for the program, methods can be found in the Scores class

\*/

public interface Bag {

public int getCurrentSize();

public boolean isEmpty();

public void add (int num);

public void remove (int num);

public void remove( );

public void clear( );

public int getFrequencyOf(int num);

public boolean contains(int num);

public String toString( );

public boolean equals(Object o);

}

/\*\*

\*

\* @author Steven Glasford

\*

\* A class implementing the methods described in the Bag interface

\*/

public class Scores implements Bag {

public int list[];

public int count;

/\*\*

\*

\* @param size - value used to initialize the size of the bag

\*

\* A constructor for the scores class

\*/

public Scores (int size){

if (size <= 0){

size = 50;

}

list = new int[size];

}

/\*\*

\*

\* @return The current size of the bag

\*

\* Used to get the size of the bag

\*/

@Override public int getCurrentSize(){

return count;

}

/\*\*

\*

\* @return whether or not the bag is empty

\*/

@Override public boolean isEmpty(){

return ((count == 0) ? true : false);

}

/\*\*

\* Used to clear the bag

\*/

@Override public void clear( ) {

count = 0;

}

/\*\*

\*

\* @param num - add a given number to the bag

\*

\* add to the list a number

\*/

@Override public void add (int num){

//add to the count of the bag first off

count++;

//double the size of the array if the array is full

if (count >= list.length){

//create a new array

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

//copy the old array into the new one

for (int i = 0; i < list.length; i++)

temp[i] = list[i];

//assign the reference of temp to list

list = temp;

//"delete" the temporary array

temp = null;

}

//add the number to the list

list[count- 1] = num;

}

/\*\*

\*

\* @param num - find the first instance of number num and remove it from the bag

\*

\* find a number in the bag and remove the first instance of it, and reduce

\* the size of the bag

\*/

@Override public void remove (int num){

boolean stop = false;

int i = 0;

//try to find the given number

while ((stop == false) && (i < count)){

//if the number is found we will remove the first found instance

//and reduce the size of the bag

if (list[i] == num){

//move the numbers down one from the point of the found number

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

list[j] = list[j+1];

//reduce the count of the numbers

count -= 1;

//stop the loop without using a break because that is habit

stop = true;

}

//go to the nuext number in the list

i++;

}

}

/\*\*

\* randomly remove a number in the bag

\*/

@Override public void remove( ){

//get a random number to remove between 0 and the count - 1

int random = (int)(Math.random() \* count + 1);

//remove the randomly generated number from the bag

for (int i = random; i < count; i++){

list[i-1] = list[i];

}

//reduce the size of the bag

count -= 1;

}

/\*\*

\*

\* @param num - find the frequency of a given number num in the bag

\* @return the frequency of the given number in the bag

\*

\* find the frequency of a given number in the list

\*/

@Override public int getFrequencyOf(int num){

int temp = 0;

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

if (list[i] == num)

temp++;

return temp;

}

/\*\*

\*

\* @param num - determine if the bag contains this number

\* @return true or false depending on if the bag contains a given number

\*

\* search through the bag to see if the given number is inside the bag

\*/

@Override public boolean contains(int num){

//a loop to go through every instance of the bag to find the given number

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

if (list[i] == num)

//return true if found

return true;

//return false if not found

return false;

}

/\*\*

\*

\* @return the bag as a string

\*/

@Override public String toString( ){

//create the string

String words;

words = "[ ";

//build the string

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

words = words + list[i] + ", ";

}

words += " ]";

return words;

}

/\*\*

\*

\* @param o - an object

\* @return true or false depending on if the object is identical to the bag

\*/

@Override public boolean equals(Object o){

//first test to see if the object is an int array

if (!(o instanceof int[])){

return false;

}

//check if the object is the same size as list[]

int[] b = (int[]) o;

if (!(b.length == list.length))

return false;

//go through the object and test for equality with respect to list[]

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

if (b[i] != list[i])

return false;

return true;

}

/\*\*

\*

\* @param i - a place in the bag array

\* @return The value stored in the array at the position, not in the bag

\*

\* get the number stored in i-th position in the bag

\*/

public int get(int i) {

int value = 0;

try {

value = list[i];

}

catch (ArrayIndexOutOfBoundsException exception){

exception.printStackTrace();

System.out.println(exception);

}

return value;

}

}

/\*\*

\*

\* @author Steven Glasford

\*/

public class Client {

/\*\*

\*

\* @param args

\*/

public static void main(String[] args) {

//create an object of type Scores

Scores scores = new Scores(100);

int range = 100;

//fill the bag with a bunch of random numbers between -100 and 100

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

int random = ((int)(Math.random() \* (range \* 2) + 1) - range);

scores.add(random);

}

//print out the bag

System.out.println(scores);

//add the number 86 to the bag

scores.add(86);

//print out the current size of the list

System.out.printf("The current size of the bag: ");

System.out.println(scores.getCurrentSize());

//randomly remove an entrant

scores.remove();

//get the 75th entrant

int at75 = scores.get(75);

System.out.printf("The 75th entrant: ");

System.out.println(at75);

//print out the number of given number in the 75th place

System.out.printf("The number of times the number in the 75th place is found: ");

System.out.println(scores.getFrequencyOf(at75));

//remove the first instance of the number in the 75th place

scores.remove(at75);

//print the frequency that the number that used to be in the 75th place now occurs

System.out.printf("The number of times the number at the 75th place now occurs, \nhaving taken the first instance of the number out of the bag: ");

System.out.println(scores.getFrequencyOf(at75));

//check the frequency of the number 86

System.out.println("The frequency of the number 86: " + scores.getFrequencyOf(86));

//check whether the array in Scores object contains the number 86

System.out.println("Does the array in Scores contain the number 86: " + scores.contains(86));

}

}

[ 42, -79, 66, 4, -79, -32, 68, -89, -80, 18, 44, 48, 49, -45, -26, 48, -11, -88, -42, -26, -56, -50, 63, -30, 57, -54, -88, 58, 84, 49, 60, -82, 39, -68, -1, -34, -29, 45, -26, -76, -91, -96, 64, -67, -75, -25, -30, -94, -9, -77, -99, 17, 99, 71, -13, 53, -97, 86, 71, -85, -38, -37, 19, -84, -38, 98, -13, 97, -36, 45, 26, 74, 52, 71, -41, 48, -64, -36, 97, -83, 41, 15, -64, 66, 16, 50, -83, 92, 95, -38, -69, -75, -26, -63, 71, 56, 32, -89, -6, -67, ]

The current size of the bag: 101

The 75th entrant: -64

The number of times the number in the 75th place is found: 2

The number of times the number at the 75th place now occurs,

having taken the first instance of the number out of the bag: 1

The frequency of the number 86: 2

Does the array in Scores contain the number 86: true

BUILD SUCCESSFUL (total time: 0 seconds)