**Guided Learning Plan for ArrayList**

**Class\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Lesson 3 ——Traverse ArrayList**

|  |  |  |
| --- | --- | --- |
| **Topic** | **details** | **Essential Knowledge** |
| Traverse ArrayList | While loops, for loops, and enhanced for each loops can all be used to traverse an ArrayList just like an array.  Here is the possible Loop conditions to traverse the ArrayList named arr      While Loop：    Enhanced For Loop：      for (int i=0;i<grades.size();i++)  {  if (grades.get(i)<70.0){  Grades.remove(i);  }  }  The above code segment does not work,try to explain why, and how to fix it?  (**Hint**:suppose the content of grades is 100,100,60,50,80,70, what’s the result after execute this code segment? | |
| Enhanced for loop | Warning: | |
| Common mistakes | **X**  ArrayList<int> myList = new ArrayList<int>();  **X**          Can you find error(s)? | |
| Summary  (ESSENTIAL KNOWLEDGE  ) | * Iteration statements can be used to access all the elements in an ArrayList . This is called * traversing the ArrayList * Deleting elements during a traversal of an ArrayList requires using special techniques to avoid skipping elements. * Since the indices for an ArrayList start at 0 and end at the number of elements − 1, accessing an index value * outside of this range will result in an IndexOutOfBoundsException being thrown. * Changing the size of an ArrayList while traversing it using an enhanced for loop can result in a * ConcurrentModificationException being thrown. Therefore, when using an enhanced for loop to traverse an ArrayList, you should not add or remove elements. | |

**Lesson 3 ——Tranverse ArrayList**

1.In the following code segment, assume that the ArrayList wordList has been initialized to contain the String values ["apple", "banana", "coconut", "lemon", "orange", "pear"].

int count = 0;

for (String word : wordList)

{

if (word.indexOf("a") >= 0)

{

count++;

}

}

System.out.println(count);

What is printed as a result of executing the code segment?

A 1 B 2 C 3 D 4 E 5

2、In the following code segment, assume that the ArrayList data has been initialized to contain the Integer values [4, 3, 4, 5, 3, 4].

int j = 0;

while (j < data.size() - 1)

{

if (data.get(j) > data.get(j + 1))

{

System.out.print(data.get(j + 1) + " ");

}

j++;

}

What, if anything, is printed as a result of executing the code segment?

A 3 3 B 4 5 C 4 5 4

D Nothing is printed because the code segment does not compile.

E Nothing is printed because an IndexOutOfBoundsException occurs.

5.In the code segment below, assume that the ArrayList object numbers has been properly declared and initialized to contain [0, 2, 4, 5].

for (int k = numbers.size() - 1; k >= 0; k--)

{

if (numbers.get(k) > k)

{

System.out.print(k + " ");

}

}

What, if anything, is printed as a result of executing the code segment?

A 1 2 3 B 2 4 5 C 3 2 1 D 5 4 2 E Nothing will be printed because an IndexOutOfBoundsException will occur.

You are given a class called WordPair that can store pairs of words.

class WordPair {

private String word1;

private String word2;

public WordPair(String w1, String w2) {

word1 = w1;

word2 = w2;

}

public String getFirst() {

return word1;

}

public String getSecond() {

return word2;

}

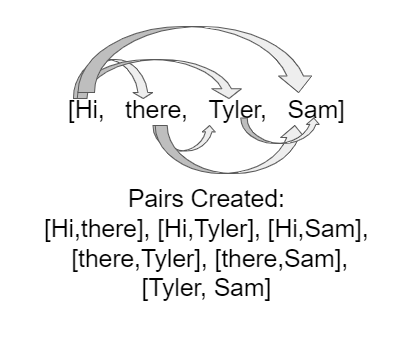
public String toString() {

return "(" + word1 + ", " + word2 + ")";

}

}

In this FRQ, you are given an array of words and you will create pairs of them by taking the first word and pairing it with all the other words, then taking the second word and pairing it with all but the first one, and so on. For example, if the word array is [“Hi”, “there”, “Tyler”, “Sam”], this figure shows how the word pairs are formed.



In the class WordPairsList below,

1. you will write the constructor which takes the array of words and pairs them up as shown in the figure.
2. you are asked to write a method called numMatches() that counts and returns the number of pairs where the first word is the same as the second word. For example, if the word array is [“hi”,”bye”,”hi”], the pairs generated would be [“hi”,”bye”], [“hi”,”hi”], and [“bye”,”hi”]. In the second pair [“hi”,”hi”], the first word is the same as the second word, so numMatches() would return 1.

import java.util.\*;

public class WordPairsList

{

private ArrayList<WordPair> allPairs;

public WordPairsList(String[] words)

{

// WRITE YOUR CODE HERE

// initialize allPairs to an empty ArrayList of WordPair objects

// nested loops through the words array to add each pair to allPairs

}

public int numMatches()

{

//Write the code for the second part described below

return 0;

}

public String toString() {

return allPairs.toString();

}

public static void main(String[] args)

{

String[] words = {"Hi", "there", "Tyler", "Sam"};

WordPairsList list = new WordPairsList(words);

System.out.println(list);

// For second part below, uncomment this test:

//System.out.println("The number of matched pairs is: " + list.numMatches());

}

}

class WordPair {

private String word1;

private String word2;

public WordPair(String w1, String w2) {

word1 = w1;

word2 = w2;

}

public String getFirst() {

return word1;

}

public String getSecond() {

return word2;

}

public String toString() {

return "(" + word1 + ", " + word2 + ")";

}

}

**Lesson 4 ——Developing algorithm with ArrayList**

1、Consider the following method, remDups, which is intended to remove duplicate consecutive elements from nums, an ArrayList of integers. For example, if nums contains {1, 2, 2, 3, 4, 3, 5, 5, 6}, then after executing remDups(nums), nums should contain {1, 2, 3, 4, 3, 5, 6}.

public static void remDups(ArrayList<Integer> nums)

{

for (int j = 0; j < nums.size() - 1; j++)

{

if (nums.get(j).equals(nums.get(j + 1)))

{

nums.remove(j);

j++;

}

}

}

The code does not always work as intended. Which of the following lists can be passed to remDups to show that the method does NOT work as intended?

A {1, 1, 2, 3, 3, 4, 5}

B {1, 2, 2, 3, 3, 4, 5}

C {1, 2, 2, 3, 4, 4, 5}

D {1, 2, 2, 3, 4, 5, 5}

E {1, 2, 3, 3, 4, 5, 5}

2、The removeElement method is intended to remove all instances of target from the ArrayList object data passed as a parameter. The method does not work as intended for all inputs.

public void removeElement(ArrayList<Integer> data, int target)

{

for (int j = 0; j < data.size(); j++)

{

if (data.get(j).equals(target))

{

data.remove(j);

}

}

}

Assume that the ArrayList object scores and the int variable low\_score have been properly declared and initialized. In which of the following cases will the method call removeElement(scores, low\_score) fail to produce the intended result?

A When scores is [0, 2, 0, 2, 0, 6] and low\_score is 0

B When scores is [2, 4, 0, 5, 7, 0] and low\_score is 0

C When scores is [3, 4, 5, 7, 7, 2] and low\_score is 1

D When scores is [8, 8, 4, 3, 3, 6] and low\_score is 3

E When scores is [9, 9, 5, 9, 7, 7] and low\_score is 5

3、In the following code segment, assume that the ArrayList numList has been properly declared and initialized to contain the Integer values [1, 2, 2, 3]. The code segment is intended to insert the Integer value val in numList so that numList will remain in ascending order. The code segment does not work as intended in all cases.

int index = 0;

while (val > numList.get(index))

{

index++;

}

numList.add(index, val);

For which of the following values of val will the code segment not work as intended?

A 0 B 1 C 2 D 3 E 4