Intro to Java

8 - ArrayLists / Solutions: Arrays

Solutions: Arrays

Write a method that calculates the sum of the items

```
public static int sumScores(ArrayList<Integer> scoreBoard) {
    int sum = 0;
    for (int x: scoreBoard) {
        sum += x;
    }
    return sum;
}
```

Write a method that finds the highest integer in the list

This works only if `scoreBoard` is not empty. Otherwise an error is generated.

```
public static int getHighScore(ArrayList<Integer> scoreBoard) {
    int max = scoreBoard.get(0);
    for (int x: scoreBoard) {
        if (x > max) {
            max = x;
        }
    }
    return max;
}
```

Write a method that finds the average value in a list

NOTE: using `getHighScore` from above

```
  public static double averageScore(ArrayList<Integer> scores) {
    double numberOfElements = scores.size();

    double average = sumScores(scores) / numberOfElements;

    return average;
  }
```

Write a method that returns a new array that contains the elements of the input array in the reverse order

```
public static ArrayList<Integer> reverseScores(ArrayList<Integer> scoreBoard) {
    ArrayList<Integer> newScoreBoard = new ArrayList<>();
    for (int i = scoreBoard.size() - 1; i >= 0; i++) {
        newScoreBoard.add(scoreBoard.get(i));
    }
    return newScoreBoard;
}
```

Remove words too short

We introduce a method `removeShortWords` that does the actual work, while `main` asks the user and prints the result of the program.

```
import java.util.ArrayList;
import java.util.Scanner;

class RemoveShortWords {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.println("How many words do you want to input?");
        int n = scanner.nextInt();
        System.out.println("Input " + n + " words:");
        ArrayList<String> words = new ArrayList<>();
        for (int i = 0; i < n; i++) {
            words.add(scanner.next());
        }

        System.out.println("What's the minimum length you want to have?");
        int minLength = scanner.nextInt();

        System.out.println("Words with minimum length " + minLength + ":");
        ArrayList<String> longWords = removeShortWords(words, minLength);
        for (String word: longWords) {
            System.out.println(word);
        }
    }

    public static ArrayList<String> removeShortWords(ArrayList<String> words, int minLength) {
        ArrayList<String> longWords = new ArrayList<>();
        for (String word: words) {
            if (word.length() >= minLength) {
                longWords.add(word);
            }
        }
        return longWords;
    }
}
```

Truncate

```
import java.util.ArrayList;
import java.util.Scanner;


class Scratch {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("How many words do you want to input?");
        int n = scanner.nextInt();

        System.out.println("Input " + n + " words:");
        ArrayList<String> words = new ArrayList<>();
        for (int i = 0; i < n; i++) {
            words.add(scanner.next());
        }

        System.out.println("From which word do you want to truncate the previous list?");
        String stopWord = scanner.next();

        System.out.println("Words up to " + stopWord + ":");
        ArrayList<String> truncatedWords = truncateAfter(words, stopWord);
        for (String word: truncatedWords) {
            System.out.println(word);
        }
    }

    public static ArrayList<String> truncateAfter(ArrayList<String> values, String x) {
        ArrayList<String> truncated = new ArrayList<>();
        for (String word: values) {
            if (word.equals(x)) {
                return truncated;
            }
            truncated.add(word);
        }
        return truncated;
    }
}
```

Finding elements

```java
public static int findValue(ArrayList<String> values, String x) {
    for (int i = 0; i < values.size(); i++) {
        if (values.get(i).equals(x)) {
            return i;
        }
    }
    return -1;
}
```

Remove duplicates

```java
public static ArrayList<String> removeDuplicates(ArrayList<String> values) {
    ArrayList<String> noDuplicates = new ArrayList<>();
    for (String s: values) {
        if (! noDuplicates.contains(s)) { // alternatively: if (findValue(noDuplicates, s) >= 0) { ... }
            noDuplicates.add(s);
        }
    }
    return noDuplicates;
}
```

Common values (intersection)

This checks for duplicate while also checking if each value of `values1` is present also in `values2`:

```java
public static ArrayList<String> commonValues(ArrayList<String> values1, ArrayList<String> values2) {
    ArrayList<String> common = new ArrayList<>();
    for (String s: values1) {
        if (values2.contains(s) && !common.contains(s)) {
            common.add(s);
        }
    }
    return common;
}
```

Alternatively, we can reuse `removeDuplicates`:

```
public static ArrayList<String> commonValues(ArrayList<String> values1, ArrayList<String> values2) {
    ArrayList<String> common = new ArrayList<>();
    for (String s: values1) {
        if (values2.contains(s)) {
            common.add(s);
        }
    }
    return removeDuplicates(common);
}
```

Union

```
public static ArrayList<String> union(ArrayList<String> values1, ArrayList<String> values2) {
    ArrayList<String> values = new ArrayList<>();
    values.addAll(values1);
    values.addAll(values2);
    return removeDuplicates(values);
}
```

Difference

```
public static ArrayList<String> difference(ArrayList<String> values1, ArrayList<String> values2) {
    ArrayList<String> values = new ArrayList<>();
    for(String v : values1) {
        if(!values2.contains(v)) {
            values.add(v);
        }
    }
    return values;
}
```

Unique Values

```
public static ArrayList<String> uniqueValues(ArrayList<String> values1, ArrayList<String> values2) {
    ArrayList<String> values = new ArrayList<>();
    for(String v : values1) {
        if(!values2.contains(v)) {
            values.add(v);
        }
    }
    for(String z : values2) {
        if(!values1.contains(z)) {
            values.add(z);
        }
    }
    return values;
}
```

Or by reusing the difference method from previous exercise:

```
public static ArrayList<String> uniqueValues(ArrayList<String> values1, ArrayList<String> values2) {
    ArrayList<String> values = difference(values1, values2);
    values.addAll(difference(values2, values1));
    return values;
}
```

Word count

```
import java.util.ArrayList;
import java.util.Scanner;

class WordCount {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.println("How many words do you want to input?");
        int n = scanner.nextInt();
        System.out.println("Input " + n + " words:");
        ArrayList<String> words = new ArrayList<>();
        for(int i = 0; i < n; i++) {
            words.add(scanner.next());
        }

        System.out.println("Word count:");

        ArrayList<String> wordsNoDuplicates = removeDuplicates(words);
        for (String word: wordsNoDuplicates) {
            int c = count(words, word);
            System.out.println(word + ": " + c);
        }
    }

    public static int count(ArrayList<String> values, String x) {
        int counter = 0;
        for (String s: values) {
            if (s.equals(x)) {
                counter++;
            }
        }
        return counter;
    }

    public static ArrayList<String> removeDuplicates(ArrayList<String> values) {
        ArrayList<String> noDuplicates = new ArrayList<>();
        for(String s: values) {
            if (! noDuplicates.contains(s)) {
                noDuplicates.add(s);
            }
        }
        return noDuplicates;
    }
}
```

Swap

```java
import java.util.Scanner;

import java.util.ArrayList;


class Main {

  public static void main(String[] args) {

    Scanner scanner = new Scanner(System.in);

    System.out.println("Insert numbers for list, stop by entering -1");

    ArrayList<Integer> numbers = getListFromUser(scanner);

    System.out.println(numbers);

    System.out.println("Enter two indices");

    int index1 = scanner.nextInt();

    int index2 = scanner.nextInt();

    swap(numbers, index1, index2);

    System.out.println(numbers);

  }


  public static ArrayList<Integer> getListFromUser(Scanner scanner) {

    int nextNumber = 0;

    ArrayList<Integer> list = new ArrayList<Integer>();

    while(nextNumber != -1) {

      nextNumber = scanner.nextInt();

      if(nextNumber != - 1) {

        list.add(nextNumber);

      }

    }


    return list;

  }


  public static void swap(ArrayList<Integer> list, int index1, int index2) {

    if(index1 < 0 || index1 >= list.size() || index2 >= list.size() || index2 < 0) {

      // one of the indices is not a valid position in the list

      System.out.println("Invalid indices");

    } else if(index1 != index2){

      Integer value = list.get(index1);
      list.set(index1, list.get(index2));
      list.set(index2, value);
    }
  }
}
```

Zip

```
import java.util.Scanner;
import java.util.ArrayList;

class Main {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.println("Enter first list of Strings, type quit to stop");
    ArrayList<String> list1 = getListFromUser(scanner);
    System.out.println("Enter second list of Strings, type quit to stop");
    ArrayList<String> list2 = getListFromUser(scanner);
    System.out.println("Your lists:");
    System.out.println(list1);
    System.out.println(list2);

    // now check if the lists have the same size
    if(list1.size() == list2.size()) {
      ArrayList<String> result = createNewList(list1, list2, list1.size());
      System.out.println(result);
    } else {
      ArrayList<String> shorterList;
      ArrayList<String> longerList;
      // find out which list is shorter and which one is longerList
      if(list1.size() > list2.size()) {
        longerList = list1;
        shorterList = list2;
      } else {
        longerList = list2;
        shorterList = list1;
      }
      // now create new list up to size of shorter list
      ArrayList<String> result = createNewList(list1, list2, shorterList.size());
      // now add remaining elements from longer list
      for(int i = shorterList.size(); i < longerList.size(); i++) {
        result.add(longerList.get(i));
      }
      System.out.println(result);
    }
  }

  public static ArrayList<String> getListFromUser(Scanner scanner) {
    String nextString = "";
    ArrayList<String> list = new ArrayList<String>();
    while(!nextString.equals("quit")) {
      nextString = scanner.next();
      if(!nextString.equals("quit")) {
        list.add(nextString);
      }
    }

    return list;
  }

  public static ArrayList<String> createNewList(ArrayList<String> l1, ArrayList<String> l2, int untilIdx)
    ArrayList<String> result = new ArrayList<String>();
    for(int i = 0; i < untilIdx; i++) {
      result.add(l1.get(i));
      result.add(l2.get(i));
    }
    return result;
  }
}
```

Guess character

```java
import java.util.Scanner;
import java.util.ArrayList;

class Main {
  public static void main(String[] args) {
    ArrayList<String> charactersToGuess = new ArrayList<String>();
    charactersToGuess.add("a");
    charactersToGuess.add("n");
    charactersToGuess.add("o");
    charactersToGuess.add("w");

    int guesses = doTheGuessing(charactersToGuess);
    System.out.println("You guessed all characters in " + guesses + " tries");
  }

  public static int doTheGuessing(ArrayList<String> charactersToGuess) {
    int guesses = 0;
    Scanner scanner = new Scanner(System.in);

    // let the user guess until all characters have been found
    while(charactersToGuess.size() > 0) {
      System.out.println("Guess a character!");
      String nextGuess = scanner.next();
      guesses++;
      // remember: remove returns true if it found and removed the string in the list!
      if(charactersToGuess.remove(nextGuess)) {
        System.out.println("You guessed correctly!");
      } else {
        System.out.println("That was wrong!");
      }
    }

    return guesses;
  }
}
```

Sorting

```
import java.util.ArrayList;

class Main {
  public static void main(String[] args) {
    ArrayList<Integer> l = new ArrayList<>();
    l.add(4);
    l.add(7);
    l.add(5);
    l.add(2);
    l.add(3);
    l.add(1);
    l.add(6);
    System.out.println(sort(l));
  }

  // this is a BubbleSort
  public static ArrayList<Integer> sort(ArrayList<Integer> values) {
    ArrayList<Integer> sorted = new ArrayList<Integer>(values);
    boolean swapped = false;
    int n = values.size();
    do {
      swapped = false;
      n = n -1;
      for(int i = 0; i < n; i++) {
        if(sorted.get(i) > sorted.get(i + 1)) {
          swap(sorted, i, i + 1);
          swapped = true;
        }
      }
    } while(swapped);
    return sorted;
  }

  public static void swap(ArrayList<Integer> l, int idx1, int idx2) {
    Integer v1 = l.get(idx1);
    l.set(idx1, l.get(idx2));
    l.set(idx2, v1);
  }
}
```

Made with ❤ by teachers at ReDI School.