Tutorial problems – Wed 23 Oct and Fri 25 Oct 2019

5min Ask whether the students have any questions. Collect them by writing them on the board and decide with the whole group which of these to address.

15min The students should discuss in small groups the following method which supposedly is to remove all even numbers in an ArrayList. However, there is a problem with it. It works fine for input such as [0, 1, 2, 3], where there are no consecutive even numbers, but not for e.g., [0, 0, 1, 1, 2, 2, 3, 3], where it would result in [0, 1, 1, 2, 3, 3]. You may give these two example inputs so that they can discuss the corresponding results.

```
public static void removeEvenNumbers(ArrayList<Integer> list) {
    for (int i = 0; i < list.size(); i++) {
        if (list.get(i) % 2 == 0){
            list.remove(i);
        }
    }
}
```

The code can be patched by adding `i--;` immediately after `list.remove(i);`.

20min The students should write a method that can count the characters and lines in a file (cf. the `wc` command in Unix). Discuss problems with counting the words as well. They are supposed to use a `BufferedReader` in a `try-catch` method. Note that the method provided has problems, e.g., if there are multiple empty lines between words the word counter is increased although it should not.

10min A more complicated but better approach would be to use a method `read.split(" ");` which splits each line into an array of words and then adds up the lengths of these arrays.

```java
import java.util.ArrayList;
import java.io.*;

/**
 * This class contains the tutorial handout exercises of week 4.
 *
 * @authors Alexandros Evangelidis and Manfred Kerber
 * @version 30-10-2018
 */
public class Wk4 {

    /**
     * This method initialises an ArrayList of integers, by adding one element at a
     * time.
     *
     * @param n the length of the ArrayList.
     * @return an ArrayList of integers.
     */
    public static ArrayList<Integer> initialise1(int n) {
        ArrayList<Integer> list = new ArrayList<Integer>(10);
        for (int i = 0; i < n; i++) {
            list.add(i);
        }
        return list;
    }

    /**
     * This method initialises an ArrayList of integers by adding two elements at a
     * time.
     *
     * @param n the length of the ArrayList.
     * @return an ArrayList of integers.
     */
    public static ArrayList<Integer> initialise2(int n) {
        ArrayList<Integer> list = new ArrayList<Integer>(10);
        for (int i = 0; i < n; i++) {
            list.add(i);
            list.add(i);
        }
        return list;
    }

    /**
     * This method removes even numbers from an ArrayList of integers.
     *
     * @param list an ArrayList of integers.
     */
    public static void removeEvenNumbers(ArrayList<Integer> list) {
        for (int i = 0; i < list.size(); i++) {
            if (list.get(i) % 2 == 0) {
                list.remove(i);
                i--;
            }
        }
    }
```

```java
/**This method counts the size of file by counting
 * the number of lines, words and characters, respectively.
 *
 * @param filename the name of a file.
 * @return the number of lines, words and characters as part of a string.
 */
public static String countFileSize(String filename) {
    int charCounter = 0;
    int wordCounter = 0;
    int lineCounter = 0;
    int readChar;
    try {
        BufferedReader in = new BufferedReader(new FileReader(filename));
        while ((readChar = in.read()) != -1) {
            if ((char) readChar == ’\n’) {
                lineCounter++;
            }
            if ((char) readChar == ’ ’) {
                wordCounter++;
            }
            charCounter++;
        }
        in.close();
        return "Number of lines: " + lineCounter + "\nNumber of words: "
            + wordCounter + "\nNumber of characters: "
            + charCounter + "\n";
    } catch (IOException e) {
        return "File not found.";
    }
}


/** This method counts the words of a given file.
 *
 * @param filename the name of a file in a string.
 * @return the number of words as part of a string.
 */
public static String countFileWords(String filename) {
    int wordCounter = 0;
    String readString;
    String[] line;

    // try-with-resources
    try (BufferedReader in = new BufferedReader(new FileReader(filename))) {
        while ((readString = in.readLine()) != null) {
            line = readString.split("\\s+");
            wordCounter += line.length;
        }
        return "\nNumber of words: " + wordCounter + "\n";
    } catch (IOException e) {
        return "File not found.";
    }
}
```

```
    /**
     * main method to test the class.
     *
     */
    public static void main(String[] args) {
        ArrayList<Integer> list1 = initialise1(4);
        System.out.println(list1);
        removeEvenNumbers(list1);
        System.out.println(list1);

        ArrayList<Integer> list2 = initialise2(4);
        System.out.println(list2);
        removeEvenNumbers(list2);
        System.out.println(list2);

        System.out.println(countFileSize("DonQuixote.txt"));
        System.out.println(countFileWords("DonQuixote.txt"));
    }
}
```