Prof. Dr. Christoph Bockisch
MSc Steffen Dick
Fachbereich Mathematik und Informatik
AG Programmiersprachen und -werkzeuge

Philipps Universität Marburg

Companion exercises
Objektorientierte Programmierung: Wintersemester 2021/2022

No. 12, due until 14.02.2022

Task 12.1: Alyx — 6 Points

With the use of Lambda-expressions, solve the following tasks within the class `Lambda`:

a) Implement a method `public static void removeIf(ArrayList<T> list, Predicate<T> filter)` that uses `public boolean removeIf(Predicate<? super E> filter)` of `ArrayList` to remove elements from a given `list`. 1

b) Write a method `public static <T> void sortBy(ArrayList<T> list, Comparator<T> comparator)` that uses `public void sort(Comparator<? super E> c)` of `ArrayList` to sort a given `list`. 1

c) Write a method `public static <T> String listToString(ArrayList<T> list)` that converts a given `list` into `String` by using `public void forEach(Consumer<? super E> action)` of `ArrayList`. Make sure that every entry has its own line. 2

   Attention: Use Stringbuilder to create your `String`.

d) Write for every of the above methods a useful JUnit-Test. Use `ArrayList` of any type of your choosing. Make sure that your list contains at least 10 different elements. 2

Task 12.2: Azarath Metrion Zinthos! 6 Points

In this task, we want to use the Streams-API to read and analyse text-files. The class `BufferedReader` has a method `Stream<String> lines()` that returns the input-stream as a per-line-stream. Create the class `IOStreams` with the following methods (if possible, pass `Exceptions` on to the calling method):

a) `public static Stream<String> words(String filePath)` that reads a file by using `BufferedReader` and converts it to a `Stream` of words. Only use methods from `String`, `Arrays`, `Stream` and (of course) `BufferedReader` in your implementation. 1

b) `public static Map<String, Integer> countWords(Stream<String> words)` that counts every occurrence of a word in a `Stream` and returns the result as a `Map`. Only use methods of the `Collectors` interface in your implementation. 2

c) `public static void writeCountedWords(Map<String, Integer> countedWords, String filepath)` that creates a new file in `filepath` (the name of the new file is included in `filepath`) and writes the content of the `Map` as content of the new file. Make sure that every entry within the map is written as its own line. 2

   Attention: You can find a file *countedWords.txt* within Ilias that can be used as a template.

d) Test your methods using JUnit. You may use *theRaven.txt* located in Ilias for your tests. The file *countedWords.txt* contains the wanted solution for counting words in *theRaven.txt*. 1