

Exercise 6 (April 2)

- ▶ Solve the following problems by programming.
- ▶ Send your programs with testing results to program06@yeah.net before (including) April 14.
- ▶ Enjoy your seventh Lab.

PRACTICE EXERCISES in Stream-Processing.pdf

- ■ **E19.4** Write a program that reads all words from a file and, using a `Stream<String>`, prints all distinct words with at most four letters (in some order).
- ■ ■ **E19.5** Write a method

```
public static <T> String toString(Stream<T> stream, int n)
```

that turns a `Stream<T>` into a comma-separated list of its first `n` elements.
- ■ **E19.6** The static `getAvailableCurrencies` method of the `java.util.Currency` class yields a set of `Currency` objects. Turn it into a stream and transform it into a stream of the currency display names. Print them in sorted order.
- ■ **E19.7** Write a lambda expression for a function that turns a string into a string made of the first letter, three periods, and the last letter, such as "W...d". (Assume the string has at least two letters.) Then write a program that reads words into a stream, applies the lambda expression to each element, and prints the result. Filter out any words with fewer than two letters.



■ ■ **E19.10** Write a method

```
public static Optional<Integer> smallestProperDivisor(int n)
```

that returns the smallest proper divisor of n or, if n is a prime number, a value indicating that no result is present.

■ ■ **E19.11** Write a program that reads an integer n and then prints all squares of the integers from 1 to n that are palindromes (that is, their decimal representation equals its reverse). Use `IntStream.range`, `map`, and `filter`.

■ ■ **E19.12** Write a method

```
public static Stream<String> characters(String str)
```

that yields a stream of strings of length 1 that contains the characters of the string `str`. Use the `codePoints` method and skip code points greater than 65535. Extra credit if you don't skip them and instead produce strings of length 2.

■ ■ **E19.13** Read all words from a file and print the one with the maximum number of vowels. Use a `Stream<String>` and the `max` method. Extra credit if you define the comparator with the `Comparator.comparing` method described in Special Topic 19.4.

■ ■ **E19.14** Read all words from a file into an `ArrayList<String>`, then turn it into a parallel stream. Use the dictionary file `words.txt` provided with the book's companion code. Use filters and the `findAny` method to find any palindrome that has at least five letters, then print the word. What happens when you run the program multiple times?

- ■ **E19.16** Read all words in a file and group them by the first letter (in lowercase). Print the average word length for each initial letter. Use `collect` and `Collectors.groupingBy`.
- ■ **E19.17** Assume that a `BankAccount` class has methods for yielding the account owner's name and the current balance. Write a function that, given a list of bank accounts, produces a map that associates owner names with the total balance in all their accounts. Use `collect` and `Collectors.groupingBy`.
- ■ **E19.18** Write a program that reads a `Stream<Country>` from a file that contains country names and numbers for the population and area. Print the most densely populated country.
- ■ **E19.19** Write a function that returns a list of all positions of a given character in a string. Produce two versions—one with streams and one without. Which one is easier to implement?
- ■ ■ **E19.20** Find all adjacent duplicates of a `Stream<String>`, by using a predicate that compares each element against the previous one (stashed away in an array of length 1), updates the array, and returns the result of the comparison. You have to be careful with the first element.



PROGRAMMING PROJECTS in Stream-Processing.pdf

- ■ **P19.1** In a stream of random integers, filter out the even ones, call `limit(n)`, and count the result. Set `n` to 10, 100, 1000, and so on. Measure the amount of time these operations take with a regular stream and a parallel stream. How big does `n` have to be for parallel streams to be faster on your computer?
- ■ **P19.2** Write a program that generates an infinite stream of integers that are perfect squares and then displays the first `n` of them that are palindromes (that is, their decimal representation equals its reverse). Extra credit if you use `BigInteger` so that you can find solutions of arbitrary length.

- **P19.10** Write a program to determine how many actors there are in the data set in Worked Example 19.2. Note that many actors are in multiple movies. The challenge in this assignment is that each movie has a list of actors, not a single actor, and there is no ready-made collector to form the union of these lists. However, there is another `collect` method that has three parameters:

- A function to generate an instance of the target
- A function to add an element to a target
- A function to merge two targets into one

For example,

```
stream.collect(  
    () -> 0,  
    (t, e) -> t + e,  
    (t, u) -> t + u)
```

computes the sum of elements in a `Stream<Integer>`. Note that the last function is only needed for parallel streams.

Define methods for generating a set, adding a list of actors into one, and for combining two sets.

- **P19.11** Write a program to determine the 100 actors with the most movies, and the number of movies in which they appear. For each movie, produce a map whose keys are the actors, all with value 1. Merge those maps as in Exercise P19.10. Then extract the top 100 actors from a stream of actors.