Practice 5

In this practice, we'll dig deeper into the question left in practice 3. **You'll be using Lambda and Streams API for this practice.**

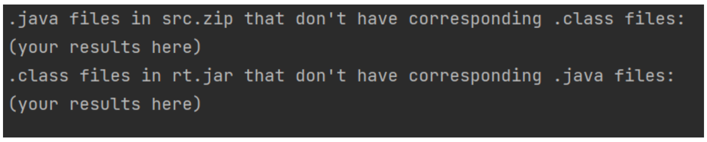
You've probably noticed from practice 3 that the number of `.java` files in `java.io` and `java.nio` packages from `src.zip` are different from the number of `.class` files in these two packages from `rt.jar` . But why? Shouldn't one source file be compiled to one bytecode file (e.g., compiling `HelloWorld.java` produced `HelloWorld.class`)?

Not necessarily. If an *inner classes* is present in a class, it will be compiled to `ClassName$InnerClassName` . If the inner class is anonymous, it will be compiled using numbers, such as `ClassName$1` . Because of this reason, one source `.java` file, when compiled, may produce multiple `.class` files. For example, compiling `java/io/Console.java` results in `java/io/Console.class` , `java/io/Console$LineReader.class` , `java/io/Console$1.class` , `java/io/Console$2.class` , and `java/io/Console$3.class` .

In this practice, let's remove all the `$` parts from the `.class` paths and re-compare them with the `.java` files. For example, `java/io/Console.class` , `java/io/Console$LineReader.class` , `java/io/Console$1.class` , `java/io/Console$2.class` , `java/io/Console$3.class` should all map to `java/io/Console.java` . After this processing, let's print out

- All the `.java` files in `java.io` and `java.nio` in `src.zip` that **still** don't have corresponding `.class` files
- All the `.class` files in `java.io` and `java.nio` in `rt.jar` that **still** don't have corresponding `.java` files

Sample Output:



```
.java files in src.zip that don't have corresponding .class files:
(your results here)
.class files in rt.jar that don't have corresponding .java files:
(your results here)
```

Evaluation Criteria

The practice will be checked on this to the next next lab class (Apr. 6) by teachers or SAs. What will be tested:

1. That you understand every line of your own code, not just copy from somewhere
2. That your program compiles correctly (javac)

3. Correctness of the program logic
4. That the result is obtained in a reasonable time

This practice will contribute 1 mark to your overall grade. Late submissions within 2 weeks after the deadline (Apr. 6) will incur a 20% penalty, meaning that you can only get 80% of the score.