Assignment Streams

Object Orientation

Spring 2025

1 Streams

A stream pipeline in Java always consists of three parts.

1. A *source* that creates a stream.
2. One or more *intermediate methods* that transform, filter, or sort the stream, or perform other operations on it.
3. A *terminal method* that returns a result or performs some actions.

2 Learning Goals

In this exercise you learn how to use streams. After doing this exercise you should be able to:

- Use streams to process arrays, lists and other collections.
- Write stream pipelines with map, filter, and reduce.

3 Your Tasks

On Brightspace you find a project template with lots of failing testcases. Fix all testcases by writing stream pipelines. Every testcase can be solved with a single line of code that looks similar to the following one.

```
int result = Stream.of(input)
      .map( /* transform the stream */ )
      .filter( /* filter elements */ )
      .reduce( /* calculate a result */ );
```

You should not change any other code in the project template.

3.1 Submit Your Project

To submit your project, follow these steps.

1. Find the folder that contains your assignment. In Visual Studio Code, you can find this by going to: File → Open Folder. Add the correct folder to a zip file. At the root level, your zip file should contain only one folder, e.g. `assignment-student-start`, which contain the entire project, i.e. the app folder and the Gradle files. *Do not submit only the `.java` files or the `src` folder!*

2. **Submit this zip file on Brightspace**. Do not submit individual Java files. Only one person in your group has to submit it. Submit your project before the deadline, which can be found on Brightspace.
3. **Do not submit any other format.** Do not submit .rar or .tar.gz or .7z files. Only zip files are allowed.