

# Training materials

- The Java Tutorials. [Lambda Expressions](#).
- The Java Tutorials. [Method References](#).
- The Java Tutorials. [The Collection Interface](#). Subsection Aggregate Operations.
- The Java Tutorials. [Lesson: Aggregate Operations](#).
- The Java Tutorials. [Reduction](#).

## Code Exercise

Create a superclass for a trainee trial (see 01-classes-\*). Use the only field account to present a trainee. Any trial consists of two tests to be estimated by integer values (mark1 and mark2) from 0 to 100 inclusively. A trial is considered to be passed if the sum of marks is not less than some constant.

Create subclasses for following kinds of trials:

- Simplified (or light) trial. It also includes two tests. But a trial is passed if both marks are not less than some constants for every test.
- Complicated (or strong) trial with two tests too. A trial is passed if the sum of a half mark1 and a whole mark2 is not less than the same constant as in the base trial.
- Extraordinary (or extra) trial. It contains an additional test. This trial is passed if the base trial is passed and a mark3 for an additional test is not less than some constant.

Define a Runner where:

1. Create an ArrayList implementation for 9 entities (3 – for a superclass and 2 – for every subclass).
2. Print the collection content (one element per line).
3. Print the number of passed trials.
4. Sort the collection by the sum of first and second marks.
5. Print sums of first and second marks from the collection (one sum per line).
6. Create a new collection from unpassed trials, clear all marks and print this collection. Check whether all trials are failed (the result type is boolean).
7. Create a numeric array from sums of first and second marks of sorted collection (see item 4) and print it in the format:  
sum[0], sum[1], ... , sum[sum.length - 1]

### Замечания и ограничения

- Использовать возможности java 8 максимально.
- Операторы цикла запрещены.
- Имена информационных классов: Trial, LightTrial, StrongTrial, ExtraTrial.