

Assignment 1 - Additional exercises

2. List mutations

- e. Implement list mutation using thread pools from `java.util.concurrent`.
 - 1. Implement parallel list mutation and piece-wise parallel list mutation using thread pools.
 - 2. Check your implementations writing a JUnit test that uses the `testListMutation` method.

3. Aggregations

- d. Implement a piece-wise parallel aggregation:
 - 1. **AggregationChunked**: Implement aggregation as a parallel aggregation, but having roughly 3 active threads at any given time (besides the main thread). Each thread should do approximately the same amount of work.
You are allowed to use your implementations of **AggregationSequential** and **AggregationParallel**.
 - 2. Check **AggregationChunked** by writing a JUnit test that uses the `testAggregation` method.
- e. Implement aggregation using thread pools.
 - 1. Implement parallel aggregation and piece-wise-parallel aggregation using thread pools.
 - 2. Check your implementations writing a JUnit test that uses the `testAggregation` method.