

Restaurant 2.0

This task is an extension of the previous restaurant task. Before you start, make a TAG of your first restaurant using the appropriate subversion command. Use the TAG [.../tags/RestaurantV1](#).

You should start with some analysis of your initial solution.

1. Describe your policy for making the Restaurant thread safe, that is the policy that you used to program Restaurant 1.0. Analyse all possible situations where things could go wrong and describe your solutions!
2. Test your code against the chosen policy. Don't alter your code in your Restaurant 1.0, except for possible testing code lines. Also look at the CPU-usage.

Document everything in a report and hand the report in on paper. The report should contain the group number and names and numbers of the students of the group as well as the names of the authors of the report.

Then improve on you implementation. Now you are allowed to use classes from the concurrency API in the `java.util.concurrent.*` packages, but for this version avoid the executor framework. Consider that forbidden for this version.

TIP: First try to make the given queue thread safe and add a `T take()` method that blocks on an empty queue. Then take advantage of this improved queue in the application and then then use the classes in `java.util.concurrent.*`.

1. Make a new design for a threadsafe restaurant: Restaurant 2.0. Now you are asked to use parts of the new concurrency package, that should make your implementation a lot easier.
2. Document you improvements as well. Hand in your results in print, before the due time as mentioned on the website.