

ASSIGNMENT 6

Concurrency

Learning Objectives:

The purpose of this assignment is to practice the concepts associated with concurrency and object oriented design.

Assignment Description:

Note that all your work must be your own work. You are not permitted to use code you find on the web. You are not permitted to share code with other students.

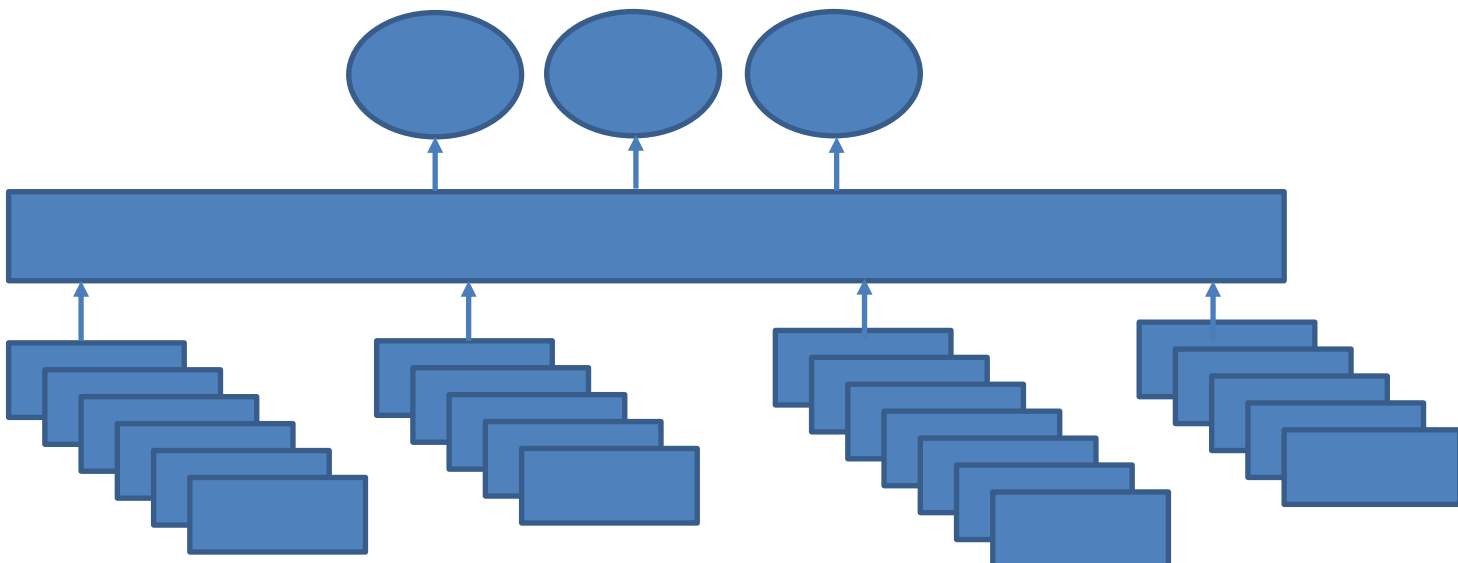
You will design and implement a factory simulator using the Java language. Your application will simulate the operation of a factory that produces four different types of products.

Problem Description

You will build an application that simulates the operations of a factory that produces four different products: Product A, Product B, Product C, Product D. Your application will take orders, and then simulate to production process. Orders will be submitted to your application using the JobSim.class application. The JobSim application will submit requests for products to your application. A product request is a string that represents a product type that the factory should produce. Valid product strings are: "ProductA", "ProductB", "ProductC", and "ProductD". Requests from JobSim will be sent continually between 1 and 6 seconds. Once a job arrives at your system, it will be queued for production. The factory has three machine tools to produce the products as follows:

| | Tool 1 | Tool 2 | Tool 3 |
|-----------|-----------------|-----------------|-----------------|
| Product A | 10 seconds | 8 seconds | Not Used |
| Product B | Not Used | 12 seconds | 6 seconds |
| Product C | 11 seconds | Not Used | 9 seconds |
| Product D | 7 seconds | 6 seconds | 5 seconds |

Each product must use the tools in order. Product A uses Tool 1, then Tool 2. Product B uses Tool 2, then Tool 3. Product C uses Tool 1, then Tool 3. Product D uses Tool 1, Tool 2, then Tool 3. Submission of requests for products and production of products will be concurrent.



Tools may be utilized simultaneously, but not on the same product. Another way to think of this is that a product cannot be in two places at the same instant. For example, Product A may be in manufacture with Tool 1 while another Product is in manufacture with Tool 2. Your application will schedule products for production and manage the utilization of the tools. You may simulate the production times using `Thread.sleep(time_in_ms)` for the allocated amount of time shown above.

An example server application has been provided that demonstrates how to take orders from the JobSim application. You will not have access to JobSim source code. Please do not deviate from the provided protocol. We will use JobSim to test your code. JobSim and the example server application are a client-server architecture utilizing TCP/IP and socket level communications. You may incorporate a server into your system anyway you like. Technically, this is a distributed system, but assume that both JobSim and your application will be on the same system (localhost).

Provide the following information to the user...

- The current products in production and the status (e.g. Product D – on Tool 3...)
- Number of jobs and product types queued for production
- Number of each type of products produced
- Total number of requests
- Average job arrival rate

You may provide a GUI if you like to provide this information, but it is not mandatory and it will not affect your grade either way.

You must use the console java compiler, *javac* to compile your code. We will test your code by putting all of your source code (*.java files) into a test directory and compiling it as follows:

```
javac *.java
```

We will then execute your program and test your application using JobSim. You must follow these instructions precisely – if we can't figure out how to build your app and run it, we won't. Note that we will not install any additional tools. You may use any tools you like to write/edit your source code, build, and test your application. However, what you submit to us must adhere to the guidelines outlined here.

If any of these requirements are unclear, please communicate with the course instructor(s).

Deliverables

You will provide all of your source code and a design document in a single archive. Describe the design of your application. Show the static and dynamic structure of your application as well as the dynamic behavior. Use UML for all of the diagrams you create. Provide a prose explanation of the design. Describe any concurrency management mechanisms you used. If you fail to include a prose description of the design to accompany your design diagrams, you will receive a zero for the design.

Packaging and Submitting Your Work

Post on blackboard a single archive file of your report and code. Please use Windows or Mac OS compressed folder to create your archive – other archive formats (rar, pkzip, 7zip, etc.) are not acceptable. Please include your name on all of your work including your source code. Name your archive as follows:

A6+<your family name>

For example, my archive name for assignment 6 would be A6+Lattanze. You will lose points for not following these directions. Please check with the instructor(s) if you are unclear.

Grading:

This assignment is worth 200 points as follows:

- **Application:** 100 points: Correct functionality. The quality of comments in the source code.
- **Quality Design Document:** 100 Points: Quality of design and quality of report (grammar, format, content, and thoroughness).