How it's made?

Introduction

Steel cables, woven thread by thread.
Before iron ore becomes a steel cable, it has to be mined, processed at multiple facilities and eventually be sold to the consumer.

Goal

The goal of the exercise is to make use of networking and threads.

The assignment will require you to manage two separate production chains consisting of mines, factories and stores. The first chain should consist of one mine, two factories and one store, all chained in a row. This production chain goes from iron ore, to purified steel, to steel cable and finally to the store.

The second chain should consist of at least two mines, five factories where at least two factories perform the same production step, and should result in two different final products which are both sold in two stores.
Make sure that the user knows at all times what each factory in the chain is doing.

Because multiple threads printing to one console is difficult to track, make a **simple** UI: A frame displaying print statements, used for each building. *Do not spend any more time than necessary on your UI. Do not make a fancy editor. Do not add actions. Only a simple view is sufficient.*

Requirements

- A **Mine** should have one or multiple miners that mine the resources, every miner should be a separate Thread. After a miner is finished extracting the product, it should send it to the next stage.
- A **Factory** can receive one type of products, has one or more production lines that process the product into another product. This product is then sent to a subsequent stage. Every production line should be a separate thread and should only be active when there is something to do.
- A **Store** can receive multiple types of products and sell them for a certain price.
- Create a GUI to display each link and its' current status/work. The GUI should consist of a frame for every mine/factory/store with a list of items and/or the work that it is currently doing.

Demands that hold for the whole system:

- The assignment involves only local networking. Therefore your program only needs to run on one machine only.
- Every link in the chain should know from the beginning what links it is connected to. For example, a mine should know what factories it is connected to and the other way around.
- If a sender has multiple addresses where the product can be sent to, it will decide between them randomly.
- An address consists of an IP-address and a port number of the receiving party.
- A receiver should be able to receive multiple transmissions of products without blocking.
- A transmission only sends one product.
- All the operations(mining, producing, selling) take one second to perform, This can be implemented using Thread.sleep().

The different components will share some features. Make sure to avoid code duplication and to keep a good coding style.

Some useful resources: Server-Client pair , Server Socket , Socket , ObjectOutputStream , ObjectInputStream , Thread , Synchronized Methods , Random .

Now you know how it's made.