Lab 1 - JuiceBottler - Documentation

Jace Claassen

Description: In this lab, I explored the concept of data and task paralyzation using threads. There are three classes: Orange.java which defines the different states of an orange (Fetched, Peeled, Squeezed, Bottled, Processed), BlockingMailbox.java which ensures proper coordination between my threads of who is responsible for parts of the orange juice-making process, and Plant.java which is the main controller of the whole simulation. Plant.java is a program that creates two threads of itself and within each plant thread, it has 5 worker threads, each responsible for a part of the orange. Once the program's 5 seconds are up the summary of information is printed to the console. This includes the number of oranges fetched, processed, wasted, and bottled.

Challenges: My first and biggest challenge was race conditions with the worker threads trying to take the orange. However, with the use of a provided BlockingMailbox class, I separated the workers by 1 process at a time, ensuring that race conditions weren't possible. In addition, it created a steady pipeline of work so that when one worker finished it would leave the current state of orange in the next worker's "mailbox" and allow itself to get another orange for their given task.

To build upon this challenge I found the implementation of BlockingMailbox to be a bit confusing. However, admittedly with the help of ChatGPT, I believe I found a solution that I understand and works.