Practical Assignment I

1 Introduction

The practical assignment I is focused on an architecture where concurrency (processes and threads) is key aspect.

Mandatory aspects:

- Programming language: Java = 8;
- IDE is NetBeans >= 8.2;
- Create one project only of Category “Java” – Java Application;
- It is not allowed to use Maven or any other equivalent tool;
- Project and project folder names: PA1_PXGYY (X–class number [1, 2], YY–group number [01, 99]).

2 Description

The project is about an harvest in an agricultural farm.

There are two main entities: the Control Center (CC) and the Farm Infrastructure (FI). The CC is responsible for supervising the harvest. The FI is the infrastructure for the agricultural harvest.

The CC and the FI are implemented as two different processes and the communication between them is through sockets.

2.1 Control Center

The CC must implement the following requirements:

- Relies on one process with a GUI;
- Has a visual interface to select the number of farmers for each harvest, NF: [2, 5];
- Has a visual interface to select the maximum number of steps a farmer can give in each advance in the Path: NS: [1-2];
- Has a visual interface to select the timeout (ms) for each farmer movement in the Path: TO: {0, 100, 250, 500, 1000};
- Has a visual interface to control the harvest state:
    - Prepare new harvest;
    - Start the harvest;
    - Collect the corn cobs;
    - Return with the corn cobs;
    - Stop the harvest;
    - Exit simulation

2.2 Farm Infrastructure

The FI must implement the following requirements:

- Has a visual interface for a storehouse where farmers are waiting for a new harvest;
- Has a departure area where farmers wait for the departure order to start the harvest;
- Has a visual interface to link the departure area and the granary area;
- Has a visual interface for a granary area where farmers collect the corn cobs;
- Each farmer is implemented as a different Thread;
- 4 monitors: Storehouse, Standing Area, Path and Granary;
- Each Monitor must be implemented as an independent Java class.

2.3 Simulation States

Initial

- User must select NF, MT and NS;
- All farmers are blocked in the Storehouse;
- After receiving a “Proceed to the Standing Area” order from the CC, NF farmers are randomly selected to proceed immediately to the Standing Area.

Prepare

- On entering the Standing area, each farmer randomly selects one the boxes available;
- All farmers will be blocked until receiving a “Proceed to the Path” order from the CC;
- After having received a “Proceed to the Path” order, each farmer randomly selects and moves immediately to one of the available boxes in the first column in the Path;
- The “Proceed to the Path” order is only valid if all farmers are blocked in the Standing area.

Walk

- Farmers advance in the Path, one by one, in the same order they entered the Path after leaving the Standing Area;
- Farmers always advance a random number of steps between 1 and NS into a random selected empty box;
- There must be a TO ms between the movement of each Farmer;
- A farmer enters the Granary whenever the next selected box is outside the Path.

Wait to Collect

- When entering the Wait to Collect state, each farmer must randomly select an empty box in the Granary;

- All farmers are blocked until receiving a Collect order form the CC;
- The Collect order is only valid if all farmers are blocked after entering the Granary.

Collect

- Farmers collect the corn cobs in any order;
- In each iteration (which takes TO ms) each Farmer collects one corn cob;
- Each farmer, after having collected ten corn cobs, enters a blocking state;
- All farmers remain blocked until having received a “Return” order from the CC;
- After having received a “Return” order, each farmer randomly selects and moves immediately to one of the available boxes in the last column in the Path;
- The CC can only send the “Return” order after all farmers have collected the corn cobs.

Return

- Farmers advance in the Path one by one in the same order they entered the Path after leaving the Granary;
- Farmers always advance a random number of steps between 1 and NS into a random selected empty box towards the Storehouse;
- There must be a TO ms between the movement of each Farmer;
- A farmer enters the Storehouse whenever the next selected box is outside the Path.

Store

- On entering the Storehouse, each farmer randomly selects and empty box;
- Each farmer stores the 10 corn cobs and immediately enters the Initial state;

Exit

- The CC can send an Exit command at any moment;
- On receiving an Exit command, every Farmer dies;
- Before dying (after all Farmers), the FI writes the message: “End of Farmer”;
- The CC dies and writes the message: “End of Simulation”.

Additional requirements:

- At any given moment, only one farmer can be inside a box (Storehouse, Standing area, Path and Granary).
- A random delay must be provided between each lock and the correspondent unlock, [0, 100ms]

Final Report

- Communication between the CC and the FI:
  - Sockets and role of each main entity;
  - Transmitter, receiver, structure, description and goals of each message;
- FI:
  - Architecture of the FI: interaction diagram;
  - Where, how and why the usage of *synchronized* (methods and/or statements) and *ReentrantLock*
- Which parts of the project have been done by each student:
  - CC server
  - FI server
  - Storehouse monitor
  - Standing monitor
  - Path monitor
  - Granary monitor

Evaluation criteria:

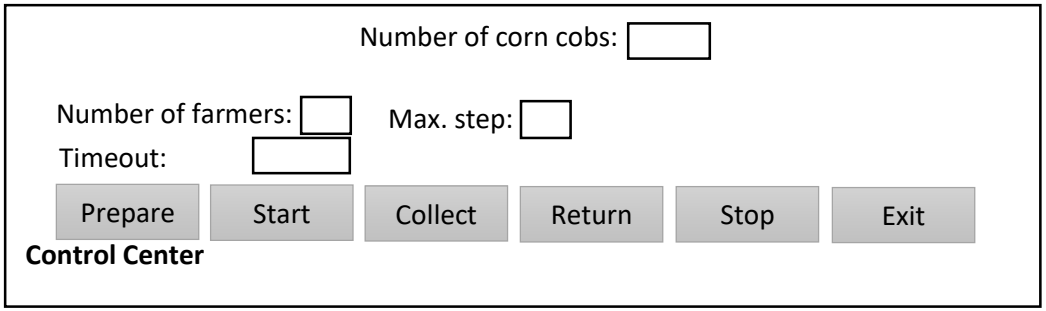
- Architecture
- Usability
- Performance
- The usage of *synchronized* (methods and/or statements) and *ReentrantLock*
- Requirements satisfaction
- JavaDoc
- Final Report

https://netbeans.org/

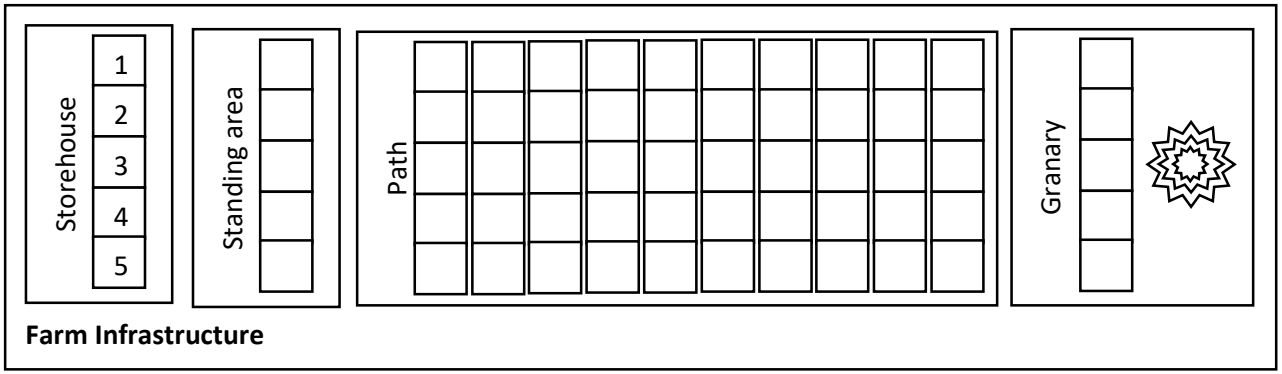
https://docs.oracle.com/javase/tutorial/uiswing/

https://docs.oracle.com/javase/tutorial/networking/sockets/index.html

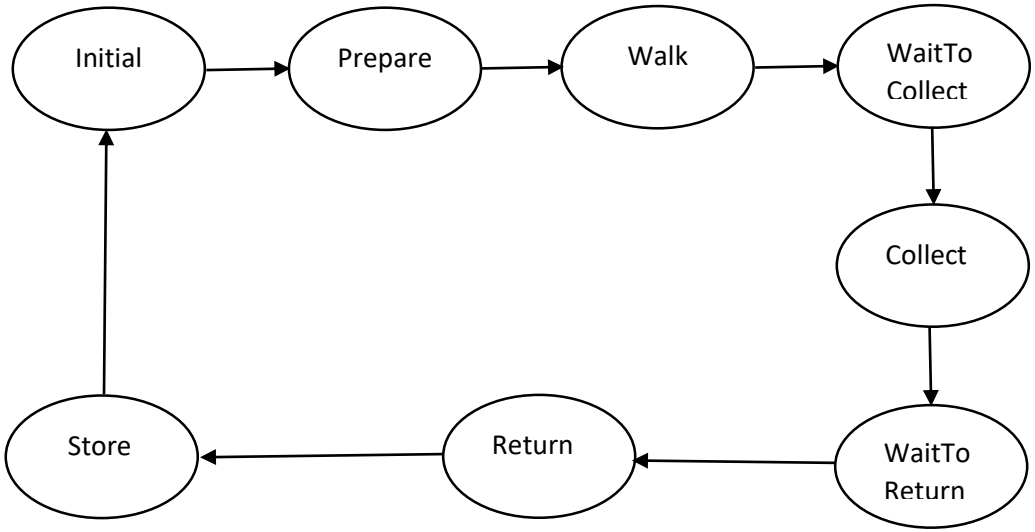
https://docs.oracle.com/javase/tutorial/essential/concurrency/index.html



Control Center



Farm Infrastructure



1
2
3
4
5