Assignment 1: Distributed PingPongPong

Final goal of this assignment is to have a working distributed PingPong application containing two Pong players, playing against one Ping player.

You will get some first practical experience with:

- structural and behavior modeling in the language you have chosen, in particular,
  - creating a model with multiple components
  - specifying the communicating protocol
  - describing the behavior using state machines
  - synchronizing the behavior of components using messages and events
- model-driven development using the tool of your choice

Preparations

1. Depending on the tool you have selected you demonstrate that the model behaves as required either by generating and running code, or by simulating the model.
2. Using available documentation learn how to generate code from your model, how to build and compile the generated code. Alternatively, learn how to simulate the model if any simulation support is available for your tool.
3. Study various examples that are available on the internet to get the basic knowledge needed for the assignment.
4. Register for a reliable forum for the tool you use. Questions related to the tool should go there or you should use the available documentation of the tool itself. Questions about the assignment should go to the Cavas discussions.

Making model: Application with one Ping and two Pongers

We want to create an application with one Ping and two Pongers: Pong1 and Pong2, such that each of the two Pongers "print" different message.

The final goal is to develop a *distributed* software application that prints numbered Pong sent messages using two Pong parts and one Ping. The Ping component sends alternatively a 'ping' to the two Pongs.

The application should meet the following MUST requirements:

- The output we require of this system looks like this:

```
Pong1 says: Ponger is ready
Pong2 says: Ponger is ready
Pong1 says: Ping received 1
Pong1 says: Pong sent 1
```

```
Pong2 says: Ping received 1
Pong2 says: Pong sent 1
Pong1 says: Ping received 2
Pong1 says: Pong sent 2
Pong2 says: Ping received 2
Pong2 says: Pong sent 2
Pong1 says: Ping received 3
Pong1 says: Pong sent 3
Pong2 says: Ping received 3
Pong2 says: Pong sent 3
Pong1 says: Ping received 4
Pong1 says: Pong sent 4
Pong2 says: Ping received 4
Pong2 says: Pong sent 4
Pong1 says: Ping received 5
Pong1 says: Pong sent 5
Pong2 says: Ping received 5
Pong2 says: Pong sent 5
```

- The Ping must send messages to both Pongers to produce the required output. The number of sent pings should be counted.
- When initialized each Ponger must print one line with the following text:
  - *<name> says: Ponger is ready*
- Whenever a Pong receives a ping with parameter "seqnr", it should print 2 lines:
  - *<name> says: Ping received <seqnr>*
  - *<name> says: Pong sent <seqnr>*
- Between each "Ping received *<seqnr>*" and each "Pong sent *<seqnr>*" line there must be a 1 second pause

It is your task to reason about the requirements and create a model: the appropriate structure, protocol and behavior. Furthermore, your task is to implement the model in the language/tool of your choice, to demonstrate the behavior of your model.