**Problem #1:**

Given:

* System A generates messages (simple strings) in random way. That system may generate N messages per second and then be idle for hours. Every message has its own priority.
* System B can process messages in some way, e.g. by sending them to stdout/console. Message processing logic is very slow, it is limited by 1 message/second.

Problem definition:

* Implement mentioned program logic (systems A and B).
* The implementation of the system A should generate messages. The implementation of the System B should receive generated messages and process them (e.g. send to stdout) with the mentioned performance limitation.
* Processing should be priority based - messages with higher priority must be processed first
* No messages generated by the System A can be lost, all messages should be processed according to their priority

Implementation limitations:

* Use only native Java v6+ API
* Do not use any external software and database servers
* Do not use any external open source frameworks, all program routines should be implemented by the candidate

Additional statements:

* Usage of Maven & Gradle is optional
* Usage of any DI framework is optional

**Problem #2:**

Additional complicated problem definition.

Implement the Problem #1 with the following additional requirements:

* The number of System B instances varies (>1)
* Every B instance should receive all messages generated by the System A