Autograder Submission Management

Scenario

You are performing a much needed update to a particular autograder for a certain computer science course. The autograder receives submissions asynchronously, runs a test script, and produces a report. Looking closely at the process, you notice that the process could be easily multi-threaded! You implement the multithreaded aspect immediately, however, you still need to figure out a good way to queue submissions. You have a server which accepts submissions via several different paths, each with its own thread, and a set of Grader objects, each running in its own thread.

Problem

Your task is to create a class which consists of a queue (i.e., FIFO list) of Submission objects. For this example, submissions can be some dummy class. This class must implement two methods: *add(Submission s)* and *process(Submission s)*. Additionally, you want to ensure that only one queue exists. Otherwise, every time a submission is submitted, a new queue will be created, and a Grader object will also not know which queue to select the next submission from.

Deliverables

1. Identify the design pattern you used to solve this problem, and the participants (i.e., the roles each class takes).
2. An implementation in a language of your choice.
3. A class diagram of your solution (including existing classes), so future developers can easily see how to work with your solution.