

Design Patterns - Singleton Pattern

Exercises

Design Patterns - Singleton Pattern

Exercise #1

1. Create a Singleton implementation which will generate a sequence of numbers to the callers.
 - a. The idea here is that there is only a single number/sequence generator and all the numbers follow a perfect sequence.
 - b. So when we do a call to the generator like `getNextNumber()` we should get the next number in the sequence no matter how we obtained the generator.
2. Make that implementation both with **eager-instantiation** and with **lazy-instantiation**.
 - a. Which one would you prefer to use and why?

Design Patterns - Singleton Pattern

Exercise #2

1. Create a Singleton implementation which will write to a common file.
 - a. Imagine that you wanted to **audit all the actions in your application.**
 - b. It would make sense to log that into a file with a guarantee that all callers will write to the same file.
 - i. For each call to the **FileAuditManager** your code will generate a timestamp and will write the message entry into the file with a newline character at the end of each message.
2. Make that implementation both with **eager-instantiation** and with **lazy-instantiation**.
 - a. Which one would you prefer to use and why?

Design Patterns - Singleton Pattern

Good Luck!