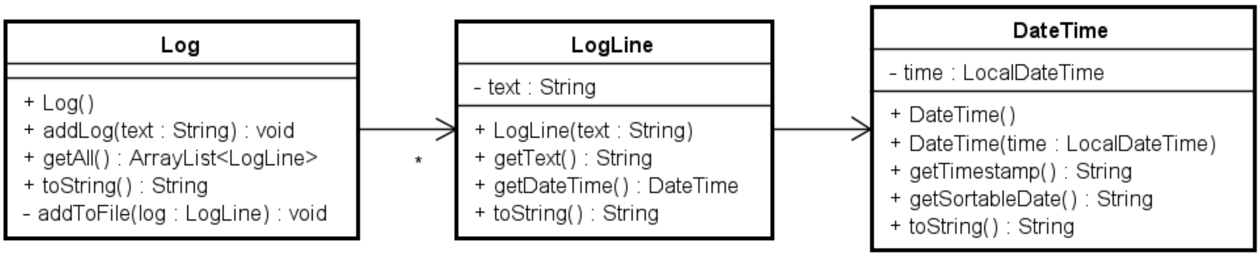
Exercise – Log

Imagine a system in which you (as a feature) are going to log all actions. Example:
“15/03/21 12:11:16 removing a cd has been pressed”
“15/03/21 12:11:23 title ‘Hello’ for cd to remove has been entered”
“15/03/21 12:11:24 cd with title ‘Hello’ has been removed in the model”
“15/03/21 12:11:26 cd with title ‘Hello’ has been removed in the database”

Part 0: Implement the Log (not a Singleton yet)

Implement the classes `Log` and `LogLine` (class `DateTime` is given at the end of this document and the private method in class `Log` is also given). *Note that you are allowed to add extra instance variables and/or private methods not shown:*


Log
+ Log()
+ addLog(text : String) : void
+ getAll() : ArrayList<LogLine>
+ toString() : String
- addToFile(log : LogLine) : void
*
LogLine
- text : String
+ LogLine(text : String)
+ getText() : String
+ getDateTime() : DateTime
+ toString() : String
DateTime
- time : LocalDateTime
+ DateTime()
+ DateTime(time : LocalDateTime)
+ getTimestamp() : String
+ getSortableDate() : String
+ toString() : String

Method `addLog` in class `Log` is creating a `LogLine`, and 1) adding it to the List, 2) adding it to the file and 3) printing it to the console.

Part 1: Converting class Log to a Singleton

Change the `Log` class from above such that it is a thread safe **Singleton**.

Test your solution and make sure to test that you are able to get the same instance again and append to the same file.

Part 2: Converting class Log to a Multiton

Change the `Log` class, this time to become a thread safe **Multiton**. Use a filename as the key in the Multiton pattern.

Add an instance variable with the filename as a String, and initialize it in the constructor. Use the instance variable for the filename in the private method `addToFile` (and delete the statement creating a filename - in the `try`-block)

Test your solution.

Use e.g. the following class for class `DateTime`:

```java
package model;

import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;

public class DateTime
{
  private LocalDateTime time;

  public DateTime() {this.time = LocalDateTime.now();}

  public DateTime(LocalDateTime time) {this.time = time;}

  public String getTimestamp()
  {
    DateTimeFormatter dtf;
    dtf = DateTimeFormatter.ofPattern("dd/MM/yyyy HH:mm:ss");
    return time.format(dtf);
  }

  public String getSortableDate()
  {
    DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy-MM-dd");
    return time.format(dtf);
  }

  @Override public String toString() {return getTimestamp();}
}
```

Use e.g. the following method for class Log:

```java
  // Appending a logLine to a file (for date 15/3/2021, the file is: "Log-2021-03-15.txt")

  private void addToFile(LogLine log)
  {
    if (log == null)
    {
      return;
    }
    BufferedWriter out = null;
    try
    {
      String filename = "Log-"
                    + log.getDateTime().getSortableDate() + ".txt";
      out = new BufferedWriter(new FileWriter(filename, true));
      out.write(log + "\n");
    }
    catch (Exception e) {e.printStackTrace();}
    finally
    {
      try
      {
        out.close();
      }
      catch (Exception e)
      {
        e.printStackTrace();
      }
    }
  }
```