

PROGRAMARE ORIENTATĂ PE OBIECTE

GENGE BÉLA

LABORATOR 4

Clase singleton

Task 1

- Implementați clasa LogMessages conform șablonului de proiectare Singleton.
- Clasa va avea trei metode de instanță:
 - initialize(String fName): void.
 - logMessage(String msg): void.
 - close(): void.
- Fiecare mesaj scris în fișier va conține o marcă de timp, urmat de mesajul propriu-zis.
- Se va folosi codul de la curs pentru scrierea în fișier, respectiv pentru generarea mărcii de timp.
- Să se adauge clasa LogMessages la aplicația din LAB3.
 - Metoda logMessage() se va apela la fiecare apăsare a butonului. Se va scrie în fișier un mesaj text, respectiv informațiile introduse de utilizator.

Task 2

- Implementați clasa LogMessagesEx **fără** șablonul de proiectare Singleton (fără metode de clasă).
 - Clasa va defini aceleași metode de instanță.
- Să se utilizeze clasa LogMessagesEx în locul clasei LogMessages în aplicația creată.

Task 3

- Să se implementeze o clasă ajutătoare (HelperChecksum) care să definească doar metode de clasă pentru calculul unor sume de control pentru un String.
- Clasa nou creată va fi adăugată la aceeași aplicație.
- Se vor considera două metode:
 - `computeCkSum1(String s): int`. Va calcula suma de control adunând valorile caracterelor din șirul dat.
 - `computeCkSum2(String s): int`. Va calcula suma de control conform algoritmului lui Fletcher:
https://en.wikipedia.org/wiki/Fletcher%27s_checksum

Task 4

- Să se calculeze suma de control pentru Username și Password la fiecare apăsare a butonului de Login.
- Să se logheze în fișierul de logare cele două sume de control.

Task 5

- Să se modifice metodele din clasa HelperChecksum ca toate metodele să aibă tipul returnat void.
- Rezultatul va fi “returnat” prin intermediul unui nou parametru adăugat la fiecare metodă (se vor folosi parametrii care să asigure persistența modificărilor – vezi exemplul de la curs).

Task 6

- Să se adauge o nouă metodă la clasa HelperChecksum care pentru un String să returneze suma de control folosind clasa CRC32 din pachetul java.util.zip.
 - `computeCRC32(String s): int.`
- Exemplificați apelul metodei similar cu task-urile anterioare.
- Notă: CRC reprezintă algoritmi pentru detectarea erorilor. Algoritmul implementat în clasa CRC32 din Java este conform RFC1952, descris aici:
 - <https://www.ietf.org/rfc/rfc1952.txt>
- Documentația Java:
 - <https://docs.oracle.com/javase/7/docs/api/java/util/zip/CRC32.html>