FONDAMENTI DI INFORMATICA L-B - APPELLO DEL 22/7/2004 - TURNO 1

Specifiche

Si richiede di realizzare una classe che implementi il funzionamento di un meccanismo di autenticazione in grado di memorizzare username e password per un certo numero di utenti e di verificare se un utente, identificato da username e password, ha diritto di accesso (login)

La classe dovrà quindi memorizzare gli username e le password di tutti gli utenti che si registrano.

Tale classe dovrà consentire di:

- Specificare all'atto di creazione di un'istanza il numero massimo di utenti che il sistema gestisce (costruttore)
- Resettare il sistema mettendo a zero il numero di utenti memorizzati (metodo reset())
- Cercare la posizione di un utente, identificato dallo username (metodo lookup()). Le posizioni partono da zero, se l'utente non viene trovato il metodo restituisce il valore -1
- Registrare un nuovo utente (metodo addUser()): vengono forniti nome e cognome, la classe determina lo username e la password sulla base delle regole sotto descritte, li memorizza e restituisce username e password attribuiti
- Determinare se un utente identificato da uno username e una password è valido o meno (metodo login()). Un utente è valido se il suo userName e memorizzato nel sistema e se la password corrisponde a quella memorizzata.
- Cambiare la password di un utente (metodo changePassword()). La password dell'utente specificato viene memorizzata al posto della precedente

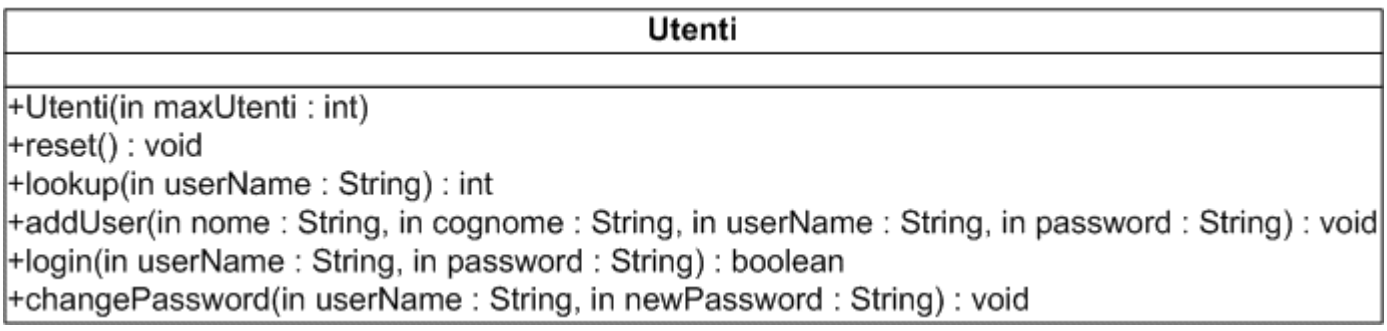
N.B. Nei metodi login() e changePassword si consiglia di utilizzare il metodo lookup per trovare la posizione dell'utente in base allo username passato come parametro.

Regole per la determinazione di username e password

Lo **username** viene determinato secondo la regola **<nome>.<cognome><numero ordine>**, il numero ordine è un numero progressivo e nome e cognome devono essere messi tutti in minuscolo (sare il metodo toLowerCase() della classe string)..Per esempio se, dopo aver resettato il sistema, registriamo nell''ordine Mario Rossi, Carlo Bianchi, Paolo Verdi e poi un altro utente di nome Mario Rossi, gli username assegnati saranno **mario.rossi1**, **carlo.bianchi2**, **paolo.verdi3**, **mario.rossi4**.

La password viene costruita con il prefisso PW e il numero d''ordine aumentato di 10000. nell''esempio precedente le password assegnate saranno quindi **PW10001** (mario.rossi1) , **PW10002** (carlo.bianchi2), **PW10003** (paolo.verdi3), **PW10004** (mario.rossi4).

Diagramma UML



Programma di esempio

```
public class EsempioUtenti
{
  public static void main(String args[])
  {
    Utenti u = new Utenti(200);

    u.reset();
    String un = "";
    String pw = "";
    u.addUser("Mario","Rossi",un,pw);
    System.out.println("username: "+un+" - password:"+pw);
    u.addUser("Carlo","Bianchi",un,pw);
    System.out.println("username: "+un+" - password:"+pw);
    u.addUser("Paolo","Verdi",un,pw);
    System.out.println("username: "+un+" - password: "+pw);

    u.addUser("Mario","Rossi",un,pw);
    System.out.println("username: "+un+" - password: "+pw);
    if (u.login("mario.rossi1","PW10001"))
      System.out.println("Utente valido");
    else System.out.println("Utente non valido");
    if (u.login("giorgio.neri6","PW10006"))
      System.out.println("Utente valido");
    else System.out.println("Utente non valido");
    u.changePassword("mario.rossi1","ciccio");
    if (u.login("mario.rossi1","PW10001"))
      System.out.println("Utente valido");
    else System.out.println("Utente non valido");
  }
}
```