

Esercitazione di laboratorio n. 2

---

### Esercizio n. 1: Personale in Java (livello base)

E' necessario scrivere un'applicazione Java che consente di gestire l'archivio del personale dell'Università della Basilicata. L'archivio del personale è tenuto in formato XML. Nell'archivio sono presenti informazioni relativi a (a) docenti, con nome, cognome, codice fiscale, qualifica (ordinario, associato, ricercatore), facoltà e dipartimento di appartenenza (es: Martin Funk, MTNFNK123, associato, Facoltà di Scienze, Dipartimento di Matematica); (b) tecnici e amministrativi, con nome, cognome, codice fiscale, livello funzionale (A, B, C, D, EP), ufficio di appartenenza (es: Filomena Lapenna, FLMLPN345, EP, Presidenza della Facoltà di Scienze); (c) contrattisti, con nome, cognome, codice fiscale, data di inizio, durata e struttura di riferimento (es: Mario Rossi, MRRRSS567, 1/1/2004, 3 mesi, Centro di Orientamento). L'applicazione deve consentire di caricare, modificare e salvare dati relativi al personale. In particolare, deve essere possibile eseguire i seguenti casi d'uso

#### Caso d'Uso: "Utente carica dati personale"

- Scenario principale
  - l'utente fornisce il nome di un file su disco
  - il sistema carica dal file i dati relativi all'archivio del personale
- Scenari alterativi:
  - Errore nella lettura da file: l'applicazione segnala il problema all'utente stampando un messaggio e il caso d'uso si conclude
  - Ordine non valido rispetto al DTD: l'applicazione segnala il problema all'utente stampando un messaggio e il caso d'uso si conclude

#### Caso d'Uso: "Utente ricerca dipendente per nome"

- Scenario principale
  - l'utente specifica un cognome
  - il sistema cerca nell'archivio tutti i dipendenti con il cognome dato e ne visualizza i dati sullo schermo

Scrivere l'applicazione Java che effettua le operazioni elencate sopra, secondo le seguenti specifiche:

- Utilizzare l'architettura a tre strati vista a lezione (interfaccia&controllo, modello, persistenza)
- Sviluppare test di regressione utilizzando **JUnit** per le classi del modello (in particolare, per scrivere test su scrittura e caricamento, è opportuno predisporre dei file di dati di test e verificare che vengano letti e scritti correttamente)
- E' possibile utilizzare JDOM oppure JAXP come libreria per XML

### Esercizio n. 2 Personale in C# (livello intermedio)

Sviluppare l'applicazione descritta all'Esercizio n. 1 in C#

### Esercizio n. 3: Poker in C# (livello avanzato)

E' necessario scrivere un'applicazione C# per giocare a poker. Per semplicità si assuma che la partita si svolge tra due giocatori. Le regole del poker prevedono che ciascuno dei due giocatori riceve una mano di 5 carte francesi estratte da un mazzo di carte francesi (per semplicità vengono considerate tutte e 52 le carte, da 2 fino ad Asso, dei quattro semi, cuori ?, quadri ?, fiori ?, picche ?). Ciascun giocatore può decidere di cambiare un certo numero di carte, da 0 fino a 4. Al termine della mano vince il giocatore che ha il punteggio più alto. I punteggi del poker sono i seguenti, in ordine crescente di importanza:

- coppia (due carte dello stesso valore di semi diversi: es: 7 picche ? e 7 quadri ?)
- doppia coppia (due coppie: es: 7 picche ?, 7 quadri ?, J picche ? e J quadri ?)
- tris (tre carte dello stesso valore di semi diversi: es: 7 picche ?, 7 quadri ? e 7 fiori ? )
- scala (le cinque carte hanno valori consecutivi: es: 7 picche ?, 8 quadri ?, 9 fiori ?, 10 quadri ? e J fiori ? )
- full (un tris e una coppia: 7 picche ?, 7 quadri ? , 7 fiori ? , J picche ? e J quadri ?)
- colore (le cinque carte sono tutte dello stesso seme ma non consecutive :es: 5 picche ?, 8 picche ?, J picche ?, 10 picche ? e A picche ?)
- poker (quattro carte dello stesso valore: es: 7 picche ?, 7 quadri ? , 7 fiori ? , 7 cuori ?)
- scala reale (le cinque carte hanno valori consecutivi dello stesso seme: es: 7 picche ?, 8 picche ?, 9 picche ?, 10 picche ? e J picche ?)

L'applicazione deve consentire di eseguire il seguente caso d'uso:

#### Caso d'Uso: “Giocatori giocano mano”

- Scenario principale
  - il sistema distribuisce a caso le cinque carte iniziali ai due giocatori
  - il primo giocatore decide quante e quali carte cambiare
  - il sistema sostituisce le carte scartate del primo giocatore
  - il secondo giocatore decide quante e quali carte scartare
  - il sistema sostituisce le carte scartate dal secondo giocatore
  - il sistema visualizza i punteggi dei due giocatori e stabilisce il vincitore

### Istruzioni per la creazione dell'applicazione

- Effettuare il “login” sulla macchina assegnata utilizzando come nome utente la propria matricola e come password la password assegnata dal CISIT
- Creare i file della propria applicazione nella cartella “**Documenti**” del disco C: (**c:\Documents and Settings\<matricola>\Documenti**); es: lo studente di matricola 12654 lavorerà nella cartella c:\Documents and Settings\12654\Documenti
- Per eseguire il compilatore, lanciare il prompt dei comandi (*Start >> Tutti i Programmi >> Accessori >> Prompt dei Comandi*), e spostarsi nella cartella **Documenti** con le seguenti istruzioni:
  - o cd \Documents\*
  - o cd <matricola> (es: cd 12654)
  - o cd Documenti
- La documentazione del linguaggio Java è disponibile nella cartella **c:\jdk1.4.2\docs**. Per consultarla aprire il file **index.html** con un qualsiasi browser Web
- La documentazione del framework .NET è disponibile a partire dal file **startHere.htm** nella cartella **C:\Programmi\Microsoft.NET\SDK\v1.1\**; aprire il file utilizzando Internet Explorer
- **JUnit** è già installato sulle macchine ed aggiunto al CLASSPATH; per eseguire il testRunner grafico è possibile utilizzare il comando **java junit.swingui.TestRunner** <nomeDellaClasseDiTest>
- La classe **it.unibas.utilita.Console** è disponibile nella cartella **c:\lib**
- Durante l'utilizzo di una macchina sarà disponibile – come sui calcolatori del Centro di Calcolo Studenti del CISIT – la propria cartella personale di rete, montata come disco Z:; è possibile salvare il lavoro svolto sul disco Z: in modo da poterlo recuperare successivamente anche dalle altre postazioni di lavoro
- Al termine dell'esercitazione disconnettersi e NON spegnere il computer