
Esempio Prova Finale – Livello Base-Intermedio

Tempo a disposizione: 2:30 ore

Esercizio 1 (Linguaggio Java)

E' necessario scrivere un'applicazione **Java** che gestisca i dati una società che si occupa di affittare appartamenti. La società gestisce una serie di edifici. Ogni edificio è rappresentato da: a) l'indirizzo, b) l'anno di costruzione e c) un livello di qualità della zona urbana che può essere "ottimo", "buono" o "sufficiente". Un esempio di edificio può essere il seguente: {"via Napoli 110, Potenza", 2010, "buono"}.

L'applicazione deve consentire di svolgere i seguenti caso d'uso:

"Utente aggiunge edificio"

- l'utente fornisce i dati di un nuovo edificio
- il sistema salva l'edificio

Scenario alternativo

- dati scorretti: il sistema segnala l'errore ed invita a fornire dati corretti
- indirizzo esistente: il sistema segnala l'errore ed invita a fornire un indirizzo non presente

"Utente verifica data costruzione edifici"

- L'utente inserisce un anno di costruzione (es: 2001)
- Il sistema cerca tutti gli edifici che hanno come anno di costruzione lo stesso inserito o più recente
- L'utente inserisce il nome di una strada (es: "via Roma")
- Il sistema verifica che tutti gli edifici estratti al passo precedente siano nella stessa strada
- Il sistema stampa un messaggio sullo schermo per segnalare che la condizione precedente sia verificata o meno

Sviluppare l'applicazione che implementa i casi d'uso elencati, seguendo il processo di sviluppo descritto a lezione, e in particolare le seguenti operazioni:

- sviluppare il modello concettuale dell'applicazione
- scelta dei componenti del modello
- scelta dei componenti di interfaccia e controllo
- sviluppo del codice dei componenti, nel package **"it.unibas.affitti"**
- strutturare l'applicazione con i menu
- produrre gli script per la compilazione e l'esecuzione (es. **compila.bat** e **esegui.bat**)
- utilizzare la libreria **it.unibas.utilita**, presente nella cartella **c:\lib**

Esercizio 2 (Linguaggio C++)

È necessario scrivere un programma in linguaggio **C++** che gestisca una collezione di immagini. Per ciascuna immagine è necessario rappresentare (a) il nome del file (es: panorama), (b) l'estensione (es: png, jpg o pdf), (c) la dimensione in KB (es: 500). La collezione può avere **dimensione variabile**. Il programma deve consentire di effettuare le seguenti operazioni attraverso l'utilizzo di un **menu**:

- acquisire dalla tastiera i valori relativi alla collezione;
- stampare i valori della collezione sullo schermo;
- cercare tutti le immagini di un determinato formato con dimensione superiore a 200 KB, ovvero:
- leggere dalla tastiera il nome del formato;
- cercare tutte le immagini che hanno una dimensione superiore a 200 KB e del formato letto al passo precedente;
- stampare le immagini trovate al passo precedente sullo schermo

Esercizio 3 (Linguaggio C#) – Per la prova di livello Intermedio

E' necessario scrivere un'applicazione **C#** che gestisca i dati di una collezione di dipinti. Per ogni dipinto è necessario rappresentare il titolo, l'autore, l'anno di produzione e il valore in euro. Un esempio di dipinto può essere il seguente: {"Primavera", "Botticelli", 1478, 10Mln}.

L'applicazione deve consentire di svolgere i seguenti caso d'uso:

"Utente aggiunge dipinto"

- l'utente fornisce i dati di un nuovo dipinto

- il sistema salva il dipinto

Scenario alternativo

- dati scorretti: il sistema segnala l'errore ed invita a fornire dati corretti

“Utente cerca dipinto più costoso di un pittore”

- l'utente fornisce il nome di un pittore
- il sistema cerca tutti i dipinti del pittore e li stampa sullo schermo
- l'utente cerca il dipinto più costoso tra quelli del pittore
- il sistema stampa il dipinto più costoso sullo schermo

Scenario alternativo

- dati scorretti: il sistema segnala l'errore ed invita a fornire dati corretti