

## Esercitazione Java: "Quiz Inglese e Gestione File"

### Obiettivo:

Sviluppare un programma Java che legga un file di testo strutturato, modelli i dati in oggetti, li utilizzi per creare una scheda di apprendimento, e successivamente trasformi il tutto in un quiz interattivo con memorizzazione del punteggio e creazione di un file di output.

---

### Parte 1: Lettura File e Modello Definizione (18 punti)

#### Descrizione:

Il file di input contiene informazioni lessicali e traduzioni, strutturato come segue:

```
Nr. Vocabolo inglese Traduzione Frase inglese Traduzione frase
1 the il, lo, la, i, gli, le The cat is on the table Il gatto è sul
tavolo
2 of di, dei A lot of cats Un sacco di gatti
...
```

#### 1. Creare una classe **Definizione**

La classe deve rappresentare una singola riga del file e contenere:

- Numero identificativo (**int**)
- Vocabolo in inglese (**String**)
- Traduzione italiana (**String**)
- Frase di esempio in inglese (**String**)
- Traduzione della frase (**String**)

#### 2. Lettura del file

- Leggere il file di testo riga per riga.
- Ignorare la prima riga (intestazione).
- Creare un oggetto **Definizione** per ogni riga del file.
- Salvare tutti gli oggetti **Definizione** in una lista.

#### 3. Vista: Scheda di apprendimento

- Stampare a console, in ordine casuale, una scheda con il vocabolo, la traduzione e la frase di esempio con traduzione.
- Ripetere questa operazione finché l'utente non decide di terminare.

### Output previsto (Esempio):

```
Vocabolo: the
Traduzione: il, lo, la, i, gli, le
Frase: The cat is on the table
Traduzione frase: Il gatto è sul tavolo
```

## Parte 2: Quiz Inglese (12 punti)

### Descrizione:

Estendere il programma per trasformare i dati in un quiz interattivo.

#### 1. Quiz interattivo

- Proporre all'utente un vocabolo in inglese e tre opzioni di traduzione, una delle quali corretta.
- Mescolare casualmente le opzioni.
- L'utente deve selezionare la risposta corretta (1, 2 o 3).
- Stampare un messaggio di conferma (giusto/sbagliato) e aggiornare il punteggio.

#### 2. Memorizzazione del punteggio

- Salvare il punteggio totale in una variabile.
- Alla fine del quiz, stampare il punteggio ottenuto.

#### 3. File di output

- Creare un file di testo in cui registrare le domande, le risposte date dall'utente e il risultato (giusto/sbagliato).
- Il file deve contenere un riepilogo finale del punteggio.

### Output Console (Esempio):

```
Qual è la traduzione di "the"?  
1) di, dei  
2) il, lo, la, i, gli, le  
3) un, uno, una  
Risposta: 2  
  
Corretto!  
  
Prossima domanda...  
  
Punteggio attuale: 1
```

### File di Output (Esempio):

```
Domanda 1: Qual è la traduzione di "the"?  
Risposta utente: il, lo, la, i, gli, le  
Esito: CORRETTO  
  
Domanda 2: Qual è la traduzione di "of"?  
Risposta utente: il, lo, la, i, gli, le  
Esito: SBAGLIATO  
  
Punteggio totale: 1/2
```

Punteggio totale: 30 punti

- **18 punti:** Lettura del file, creazione del modello **Definizione** e stampa casuale delle schede di apprendimento.
- **12 punti:** Quiz interattivo con memorizzazione del punteggio e creazione di un file di output.

Suggerimenti tecnici:

- Usare la classe **Scanner** per leggere da file e console.
- Usare la classe **Random** per generare numeri casuali.
- Usare **ArrayList** o **List** per gestire le definizioni.
- Per scrivere su file, utilizzare **FileWriter** o **BufferedWriter**.
- Organizzare il codice secondo il pattern MVC (Model-View-Controller) per una struttura più chiara.

Buon lavoro! 😊

Griglia di Valutazione: Esercitazione "Quiz Inglese e Gestione File"

Aspetto Valutato	Criteri di Valutazione	Punti Massimi	Punti Ottenuti
Parte 1: Lettura File e Modello Definizione			
Creazione della classe <b>Definizione</b>	<ul style="list-style-type: none"><li>- La classe è ben definita con i campi: <b>int</b>, <b>String</b> (4 campi richiesti).</li><li>- Utilizzo appropriato di costruttore, getter e setter.</li></ul>	4	
Lettura del file di testo	<ul style="list-style-type: none"><li>- Lettura corretta del file.</li><li>- Ignorata la riga di intestazione.</li><li>- Creazione corretta di oggetti <b>Definizione</b>.</li></ul>	4	
Archiviazione dei dati	<ul style="list-style-type: none"><li>- Gli oggetti <b>Definizione</b> vengono salvati in una lista.</li><li>- Utilizzo appropriato di strutture dati (<b>ArrayList</b>).</li></ul>	3	
Stampa casuale delle schede di apprendimento	<ul style="list-style-type: none"><li>- Le schede vengono stampate correttamente.</li><li>- L'ordine delle schede è casuale.</li><li>- Traduzione e frasi ben formattate.</li></ul>	4	
Interazione utente per terminare la stampa	<ul style="list-style-type: none"><li>- L'utente può terminare la visualizzazione delle schede con un comando.</li><li>- Gestione di input non validi.</li></ul>	3	
Subtotale Parte 1		18	

Aspetto Valutato	Criteri di Valutazione	Punti Massimi	Punti Ottenuti
<b>Parte 2: Quiz Inglese</b>			
Generazione domande del quiz	<ul style="list-style-type: none"> <li>- Le domande vengono proposte con vocabolo in inglese e 3 opzioni.</li> <li>- La risposta corretta è inclusa tra le opzioni.</li> </ul>	3	
Mescolamento delle risposte	<ul style="list-style-type: none"> <li>- Le opzioni sono presentate in ordine casuale.</li> </ul>	2	
Valutazione della risposta	<ul style="list-style-type: none"> <li>- Risposte corrette/errate sono riconosciute.</li> <li>- Viene stampato un messaggio relativo all'esito della risposta.</li> </ul>	2	
Memorizzazione del punteggio	<ul style="list-style-type: none"> <li>- Il punteggio viene calcolato correttamente e aggiornato dopo ogni domanda.</li> <li>- Visualizzazione finale del punteggio.</li> </ul>	2	
Creazione del file di output	<ul style="list-style-type: none"> <li>- Il file contiene le domande, risposte dell'utente, esito (giusto/sbagliato) e punteggio totale.</li> </ul>	3	
<b>Subtotale Parte 2</b>		<b>12</b>	
<b>Totale Punti</b>		<b>30</b>	

## Criteri di Valutazione Generali

- **Chiarezza del codice:** Il codice è ben commentato e leggibile.
- **Struttura del programma:** Il programma è organizzato secondo il pattern MVC (Model-View-Controller) o ha una struttura chiara e modulare.
- **Gestione degli errori:** Input non validi o errori di lettura/scrittura del file sono gestiti correttamente.
- **Creatività ed estensione:** Eventuali miglioramenti o funzionalità aggiuntive saranno premiati con punti bonus (fino a 3 punti extra).

## Note per il valutatore

- Il codice deve essere eseguito correttamente senza errori di runtime.
- È possibile assegnare punti parziali per aspetti implementati in modo incompleto.
- Assegnare bonus solo se la funzionalità aggiuntiva è ben implementata e documentata.