

```
11 Scanner sc = new Scanner(System.in);
12 System.out.print("Enter the strings: ");
13 str = sc.nextLine();
14 System.out.println("Longest substring is : " +
15 }
16 }
17
18 class Solution {
19
20 public String longestPalindrome(String s) {
21     if (s == null || s.length() == 0) {
22         return "";
23     }
24     int n = s.length();
25     String maxStr = "";
26     for (int i = 0; i < n; ++i) {
27         for (int j = i; j < n; ++j) {
28             if (isValid(s, i, j) == true) {
29                 if (j - i + 1 > maxStr.length())
30                     maxStr = s.substring(i, j + 1);
31             }
32         }
33     }
34     return maxStr;
35 }
```



.DIEM

Wordageddon

Presentazione Progetto
2025

Corso di programmazione Java
Avanzata



Obiettivi

- Si richiede la progettazione e lo sviluppo di un'applicazione desktop, realizzata in **JavaFX**, con finalità ludico-educative, che sfidi l'utente nella memorizzazione delle parole più **frequenti** presenti in un insieme di documenti testuali.

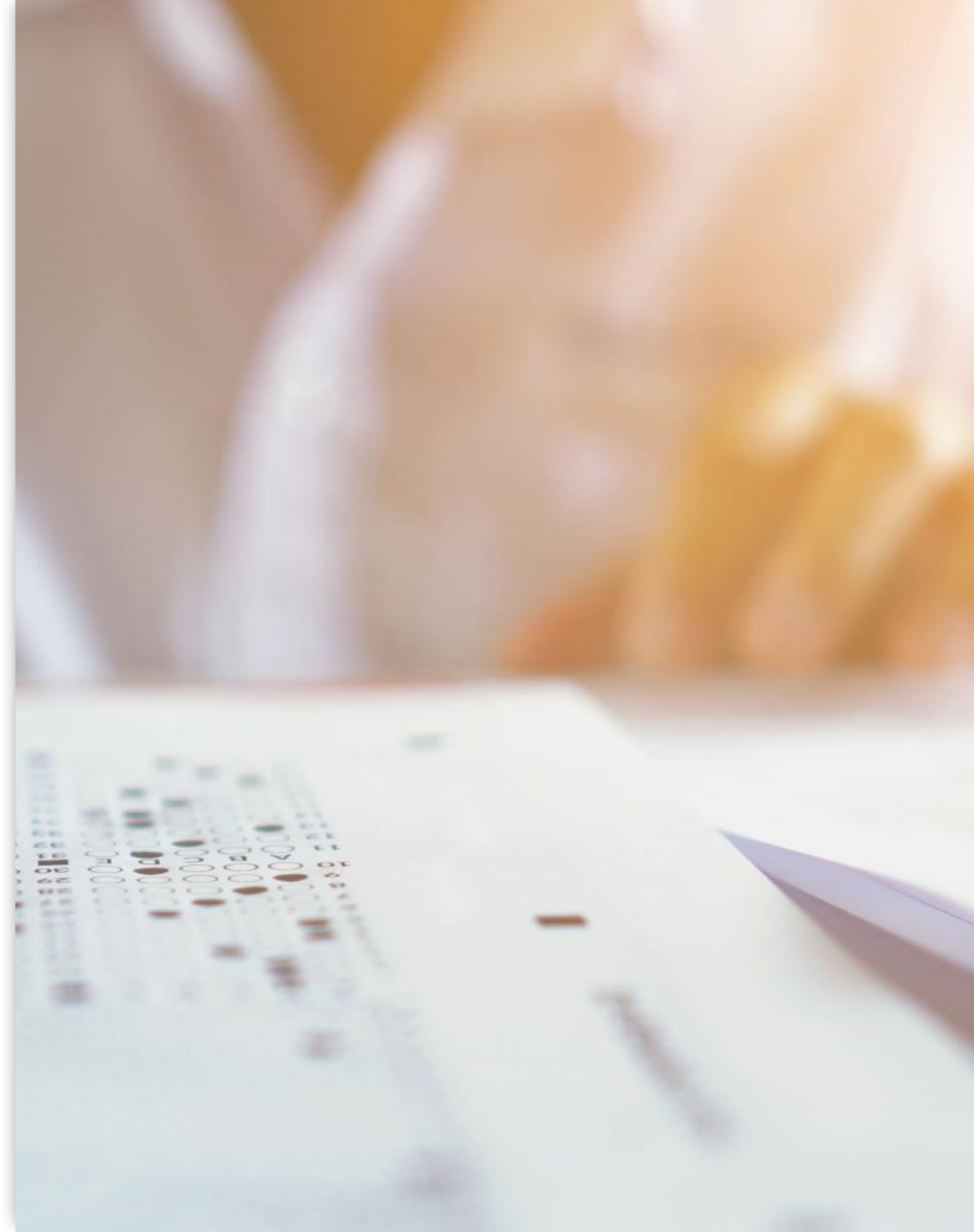
Funzionalità principali

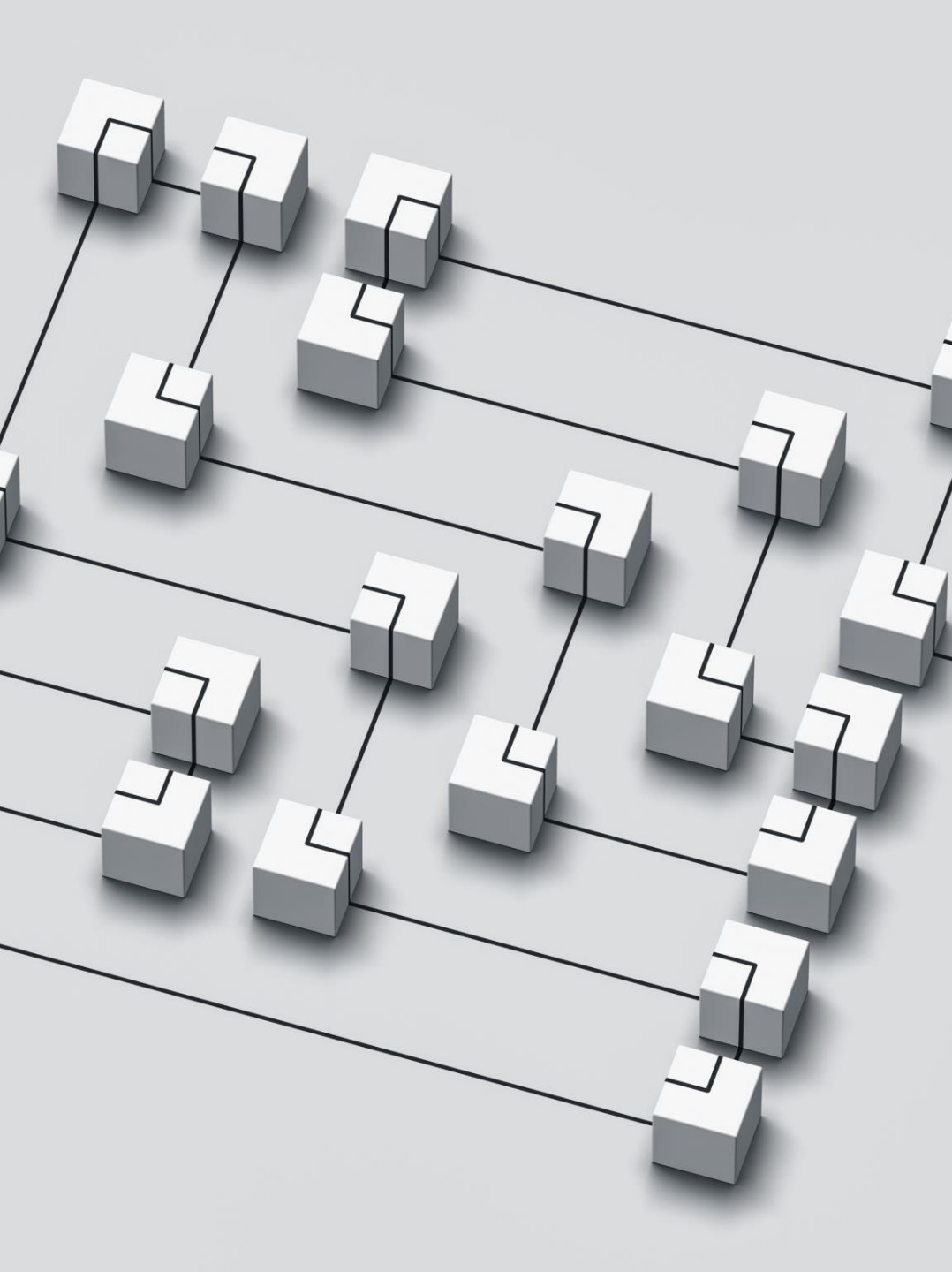


- L'applicazione mostra all'utente uno o più **documenti testuali** per un *tempo limitato*.
- I parametri di gioco (numero di documenti mostrati, lunghezza dei testi in numero di parole, tempo di lettura) **dipendono dal livello di difficoltà** selezionato (es. facile, medio, difficile).
- Terminata la fase di lettura, viene generata una **serie di domande a risposta multipla** (4 opzioni) basate sull'analisi dei documenti, come:
 - Quante volte compare la parola "cane" nel secondo documento?
 - Quale tra le seguenti parole è la più frequente in tutti i documenti?
 - Quale parola compare più spesso nel terzo documento?
 - Quale di queste parole **non appare mai** in nessun documento?
 - ...

Tipologie di domande

- Le tipologie di domande che si possono includere nel gioco, mantenendo coerenza con l'obiettivo di allenare la memoria sui termini frequenti, comprendono:
 - **domande di frequenza assoluta**, in cui si chiede quante volte un termine compare in un documento o nell'intero set
 - **domande di confronto**, che mettono a confronto le frequenze relative di più parole
 - **domande per documento specifico**, dove l'utente deve associare parole a un determinato documento
 - **domande di esclusione**, che richiedono di individuare parole mai comparse*





Gestione utenti e dati

- Il gioco è **multiutente**: ogni utente può registrarsi, effettuare il login e partecipare alle sessioni di gioco.
- Ogni sessione produce un **punteggio**, che viene **salvato in un database relazionale** associato all'account dell'utente.

Funzionalità amministrative

- È previsto un **pannello per l'amministratore** che consente di **caricare nuovi documenti di testo** tramite file, da utilizzare per le sessioni di gioco.
- L'amministratore può anche caricare una lista di **stopwords** (parole da escludere dall'analisi statistica e dalle domande, non apportando contributo semantico – articoli, preposizioni...).

Funzionalità aggiuntive (opzionali)

Classifica globale e personale

- Mostrare una **leaderboard** con i punteggi più alti globali o per livello di difficoltà.
- Mostrare i **progressi storici dell'utente** (es. punteggio medio, miglior punteggio, ultimi risultati).

Statistiche dettagliate post-gioco

- Dopo ogni sessione, mostrare le parole più frequenti, gli errori commessi, e un **confronto tra risposta scelta e corretta**.



Funzionalità aggiuntive (opzionali)

Supporto per documenti in più lingue

- L'amministratore può caricare documenti in diverse lingue (IT, EN,...), e l'utente sceglie la lingua del quiz.

Salvataggio sessioni interrotte

- L'utente può **interrompere una sessione** e riprenderla in un secondo momento.

Styling CSS





Note per l'implementazione

- Per l'implementazione delle funzionalità descritte, non è possibile usare librerie Java di terze parti, ma solo quanto offerto dalla JDK 8 o superiore.
 - Privilegiare uso di Stream API e interfacce funzionali standard con lambda e riferimenti a metodi.
 - Privilegiare l'uso di servizi JavaFX per le fasi di **analisi** dei documenti
- Per analisi dei documenti si intende una fase di trasformazione dal flusso testuale ad una rappresentazione intermedia (es. word document matrix...) con strutture dati che semplificano e velocizzano la produzione delle domande del quiz.

*può essere utile rendere persistente la fase di analisi



Deliverable

- Codice del progetto
- Documentazione javadoc
- **Breve** relazione finale
 - Descrizione del progetto
 - Specifica requisiti (Use Cases, Mockup)
 - Progettazione (Class Diagram, Sequence Diagram)

La consegna è fissata **7 giorni** prima della data dell'appello.

Prova d'esame

- Presentazione powerpoint di gruppo con descrizione delle funzionalità e scelte progettualmente rilevanti
- Demo **live** del funzionamento
- Colloquio individuale (su tutti gli argomenti trattati al corso)



Domande?

