

# Esercizi in Java

|   |          |
|---|----------|
| <b>Esercizi in Java</b>   | <b>1</b> |
| <b>Livello: facile</b>  | <b>3</b> |
| 1. Rubrica Telefonica (ArrayList + Incapsulamento)                          | 3        |
| 2. Somma da Input (args + array)  | 3        |
| 3. Forma Astratta (Classi astratte + ereditarietà)                          | 3        |
| 4. Registro Studenti (ArrayList + Classi)                                   | 4        |
| 5. Filtro Numeri Pari (Array + if)  | 4        |
| 6. Interfaccia Animale (Interfacce + Ereditarietà)                          | 4        |
| 7. Contatore Generico (Generics)  | 4        |
| 8. Classifica Sportiva (ArrayList + Sorting)                                | 4        |
| 9. Incapsulamento Base (get/set)  | 4        |
| 10. Calcolatrice con switch (args)  | 4        |
| 11. Magazzino Prodotti (ArrayList + Ricerca)                                | 5        |
| 12. Registro Presenze (Array 2D)  | 5        |
| 13. Macchina Astratta (Classe Astratta + Overriding)                        | 5        |
| 14. Parole Lunghe (Array + Condizioni)                                      | 5        |
| 15. Biblioteca (ArrayList + Oggetti)  | 5        |
| 16. Stack Generico (Generics + ArrayList)                                   | 5        |
| 17. Scacchiera (Array 2D + Logica)  | 5        |
| 18. Ordina Parole (args + Arrays.sort)                                      | 6        |
| 19. Interfaccia Pagabile (Interfacce)                                       | 6        |
| 20. Diario di Bordo (ArrayList + Date)                                      | 6        |
| <b>Livello: facile-intermedio</b>   | <b>6</b> |
| 21. Sistema di Prenotazione Cinema (OOP + ArrayList + Logica)               | 6        |
| 22. Gestione Ordini con Interfacce (Interfaccia + ArrayList + Polimorfismo) | 6        |
| 23. Registro Medico (Classi + ArrayList + Ricerca Avanzata)                 | 6        |
| 24. Gestione Catalogo Prodotti con Filtri (Generics + Funzioni + Filtri)    | 7        |
| 25. Sistema Banca (Classi astratte + Ereditarietà + Overriding)             | 7        |
| 26. Gestione File Studenti (File I/O + ArrayList + Parsing)                 | 7        |
| 27. Parser JSON Manuale (Stringhe + Logica + ArrayList)                     | 7        |
| 28. Magazzino Multiclasse (ArrayList + Ereditarietà + instanceof)           | 7        |
| 29. Gestione Playlist (ArrayList + Oggetti + Metodi Statici)                | 7        |
| 30. Rubrica con Ricerca Avanzata (ArrayList + Stream-like)                  | 8        |
| 31. Sistema di Login (HashMap + Stringhe + Sicurezza base)                  | 8        |
| 32. File Logger (FileWriter + Singleton + Data)                             | 8        |
| 33. Parcheggio (OOP + Array + Occupazione posti)                            | 8        |
| 34. Classifica Mondiale (Ereditarietà + Comparable)                         | 8        |
| 35. Gioco dei Dadi (OOP + Random + Logica)                                  | 8        |
| 36. Libreria con Ricerca Multi-criterio (ArrayList + Filtri Combinati)      | 8        |
| 37. Analisi Statistica Voti (Array + Metodi Statici + Media/Moda/Mediana)   | 9        |

|   |           |
|---|-----------|
| 38. Simulazione Metereologica (Classi + ArrayList + Random + Date)            | 9         |
| 39. Task Manager (OOP + Priorità + ArrayList)                                 | 9         |
| 40. Mini ORM (Riflessione + Classi + HashMap)                                 | 9         |
| <b>Livello: intermedio</b>  | <b>9</b>  |
| 1. Gestione di un Negozio Online (OOP + ArrayList)                            | 9         |
| 2. Sistema di Biblioteca (Ereditarietà + Interfacce)                          | 9         |
| 3. Rubrica Telefonica (ArrayList + Ricerca)                                   | 10        |
| 4. Calcolatrice Avanzata (Classi Astratte + Polimorfismo)                     | 10        |
| 5. Gestione di una Scuola (Ereditarietà + Array)                              | 10        |
| 6. Sistema di Prenotazione (Interfacce + ArrayList)                           | 10        |
| 7. Gestione di un Magazzino (Generics + HashMap)                              | 10        |
| 8. Sistema di Votazione (Array + Statistiche)                                 | 10        |
| 9. Gestione di Eventi (ArrayList + Date)                                      | 11        |
| 10. Sistema di Pagamenti (Interfacce + Polimorfismo)                          | 11        |
| 11. Gestione di una Playlist (ArrayList + File I/O)                           | 11        |
| 12. Sistema di Login (HashMap + Sicurezza)                                    | 11        |
| 13. Gestione di una Biblioteca (Array + Ricerca)                              | 11        |
| 14. Sistema di Fatturazione (Classi + Calcoli)                                | 11        |
| 15. Gestione di un Calendario (ArrayList + Date)                              | 12        |
| <b>Livello: intermedio-scomodo</b>  | <b>12</b> |
| 16. Sistema di Inventario (Generics + ArrayList)                              | 12        |
| 17. Simulazione di Gara Sportiva (OOP + Ordinamento)                          | 12        |
| 18. Analizzatore di Testo (File I/O + Map + Frequenze)                        | 12        |
| 19. Sistema di Sondaggi (Interfacce + Statistiche)                            | 12        |
| 20. Gioco "Battaglia Navale" semplificato (Array + Logica)                    | 12        |
| 21. Registro Presenze Studenti (ArrayList + Map)                              | 13        |
| 22. Convertitore di Valute (Enum + Interfaccia)                               | 13        |
| 23. Mini CRM (Clienti, Interazioni, Appuntamenti)                             | 13        |
| 24. Sistema di Statistiche Universitarie (Classi + Calcoli)                   | 13        |
| 25. Generatore di Codici Fiscali (Stringhe + Calcoli)                         | 13        |
| 26. Gestione Portafoglio Azionario (OOP + Map)                                | 13        |
| 27. Simulazione Parcheggio (Array + Stato)                                    | 14        |
| 28. Gestione Promozioni di un Supermercato (Enum + Strategie)                 | 14        |
| 29. Parser CSV (File I/O + Parsing + OOP)                                     | 14        |
| 30. Simulazione Distributore Automatico (ArrayList + Interazione)             | 14        |
| 31. Gestione Dipendenti (Ereditarietà + Calcolo Stipendi)                     | 14        |
| 32. Sistema di Allarme (Observer pattern semplificato)                        | 14        |
| 33. Simulatore Meteo (Random + OOP + Enum)                                    | 15        |
| 34. Simulazione Gioco Dadi (Random + Statistiche)                             | 15        |
| 35. Creazione Curriculum (OOP + StringBuilder)                                | 15        |
| <b>Livello: complesso/micro progetti</b>                                      | <b>15</b> |
| 1. Mini Database Relazionale (OOP + HashMap + Join Simulati)                  | 15        |
| 2. Analizzatore di Dipendenze tra Classi (Reflection + Graph + DFS)           | 15        |
| 3. Sistema di Cache a Due Livelli (Design Pattern + LinkedHashMap + Eviction) | 16        |

|   |    |
|---|----|
| 4. Compilatore Matematico Semplice (Parser + Tree + Ricorsione)         | 16 |
| 5. Chat Server/Client (Socket + Thread + Comunicazione)                 | 16 |
| 6. Microframework Web (HTTP + Sockets + Routing)                        | 16 |
| 7. Generatore di Codice Java (Reflection + Template)                    | 16 |
| 8. Simulatore di Scheduler CPU (Thread + Strategia + Sincronizzazione)  | 16 |
| 9. File System Virtuale (Composite Pattern + Navigazione + Ricorsione)  | 17 |
| 10. ORM Base con JDBC (Database + DAO + Reflection + Generics)          | 17 |
| 11. Algoritmi di Compressione (Huffman + File I/O + Bit Manipulation)   | 17 |
| 12. Sistema Event-Driven (Observer + Event Loop + Lambda)               | 17 |
| 13. Interprete Lisp Semplificato (Parser + Stack + Ricorsione)          | 17 |
| 14. Tool di Testing Automatico (Reflection + Annotations)               | 17 |
| 15. Analisi Statica del Codice (File + Regex + Parsing)                 | 18 |
| 16. Simulatore di Reti Neurali Feedforward (Math + Matrix + OOP)        | 18 |
| 17. Editor di Testo da Terminale (Input + Buffer + Undo/Redo)           | 18 |
| 18. Motore di Regole Dinamico (DSL + Parser + Strategy Pattern)         | 18 |
| 19. Simulatore di Rete Peer-to-Peer (Sockets + Hashing + Bilanciamento) | 18 |
| 20. Algoritmi di Pathfinding (Graph + A + Visualizzazione Testuale)*    | 18 |

Livello: facile

## 1. Rubrica Telefonica (ArrayList + Incapsulamento)

Crea una classe `Contatto` con nome, numero di telefono (privati) e metodi getter/setter.  
Nel `main`, usa un `ArrayList<Contatto>` per gestire l'aggiunta e la stampa dei contatti.

---

## 2. Somma da Input (args + array)

Nel `main(String[] args)`, prendi una serie di numeri interi da riga di comando e stampa la loro somma.

---

## 3. Forma Astratta (Classi astratte + ereditarietà)

Crea una classe astratta `Forma` con metodo astratto `double area()`.  
Implementa `Cerchio` e `Rettangolo`, e nel `main` crea un array di forme e stampa l'area di ciascuna.

---

#### 4. Registro Studenti (ArrayList + Classi)

Crea una classe `Studente` con `nome`, `matricola`, `voti[]`.

Nel main, aggiungi studenti a un `ArrayList`, calcola e stampa la media voti.

---

#### 5. Filtro Numeri Pari (Array + if)

Nel `main`, dichiara un array di interi. Stampa solo i numeri pari usando un ciclo.

---

#### 6. Interfaccia Animale (Interfacce + Ereditarietà)

Crea un'interfaccia `Animale` con metodo `verso()`.

Implementa `Cane` e `Gatto`. Nel main, chiama il verso su una lista di animali.

---

#### 7. Contatore Generico (Generics)

Crea una classe `Contatore<T>` che conta elementi di tipo `T`.

Nel main, conta quanti numeri e quante stringhe ci sono in due array diversi.

---

#### 8. Classifica Sportiva (ArrayList + Sorting)

Crea una classe `Squadra` con `nome` e `punti`.

Nel main, aggiungi squadre a una lista e ordina in base ai punti.

---

#### 9. Incapsulamento Base (get/set)

Crea una classe `Persona` con `nome` e `età` privati.

Nel main, crea una persona e modifica i dati con i metodi pubblici.

---

#### 10. Calcolatrice con switch (args)

Usa `args` per ricevere due numeri e un'operazione (`+`, `-`, `*`, `/`).

Stampa il risultato.

---

## 11. Magazzino Prodotti (ArrayList + Ricerca)

Crea una classe `Prodotto` con nome, codice, prezzo.

Nel main, cerca un prodotto per codice tra quelli inseriti in un `ArrayList`.

---

## 12. Registro Presenze (Array 2D)

Gestisci un array bidimensionale per tracciare le presenze di studenti in una settimana.

Nel main, stampa le presenze totali per ogni studente.

---

## 13. Macchina Astratta (Classe Astratta + Overriding)

Crea una classe astratta `Macchina` con metodo `avvia()`.

Implementa `Auto` e `Moto`. Nel main, chiama `avvia()` su entrambe.

---

## 14. Parole Lunghe (Array + Condizioni)

Nel main, prendi da `args` una lista di parole.

Stampa solo quelle più lunghe di 5 lettere.

---

## 15. Biblioteca (ArrayList + Oggetti)

Crea una classe `Libro` con titolo, autore, anno.

Nel main, inserisci e stampa una lista di libri in una `ArrayList`.

---

## 16. Stack Generico (Generics + ArrayList)

Crea una classe `Stack<T>` con metodi `push`, `pop`, `peek`.

Nel main, crea uno stack di stringhe e prova tutte le operazioni.

---

## 17. Scacchiera (Array 2D + Logica)

Crea una scacchiera 8x8 in un array 2D e popola la prima riga con "Torre", "Cavallo", ecc.

Nel main, stampa la scacchiera.

---

## 18. Ordina Parole (args + Arrays.sort)

Nel main, ricevi da `args` parole da ordinare alfabeticamente e stampale in ordine.

---

## 19. Interfaccia Pagabile (Interfacce)

Crea un'interfaccia `Pagabile` con metodo `calcolaTotale()`.

Implementa in `Fattura` e `Abbonamento`. Stampa il totale da pagare.

---

## 20. Diario di Bordo (ArrayList + Date)

Crea una classe `VoceDiario` con testo e data (usa `LocalDate`).

Nel main, aggiungi voci e stampa quelle degli ultimi 7 giorni.

Livello: facile-intermedio

## 21. Sistema di Prenotazione Cinema (OOP + ArrayList + Logica)

Crea classi `Sala`, `Film`, `Prenotazione`. Ogni sala ha un numero di posti.

Nel main, consenti di prenotare posti per un film in una sala e verifica la disponibilità.

---

## 22. Gestione Ordini con Interfacce (Interfaccia + ArrayList + Polimorfismo)

Crea un'interfaccia `Spedibile` con metodo `spedizione()`.

Classi `OrdineOnline` e `OrdineNegozio` la implementano in modo diverso.

Nel main, gestisci una lista mista di ordini e stampa il metodo di spedizione per ciascuno.

---

## 23. Registro Medico (Classi + ArrayList + Ricerca Avanzata)

Crea classi `Paziente`, `Visita` e `Medico`.

Nel main, inserisci dati e permetti la ricerca delle visite per un paziente specifico.

---

## 24. Gestione Catalogo Prodotti con Filtri (Generics + Funzioni + Filtri)

Crea una classe generica `Catalogo<T>` che permette di aggiungere e filtrare prodotti tramite `Predicate<T>`.

Nel main, filtra prodotti per prezzo, nome, o disponibilità.

---

## 25. Sistema Banca (Classi astratte + Ereditarietà + Overriding)

Classe astratta `ContoBancario`, sottoclassi `ContoCorrente`, `ContoRisparmio`.

Ogni tipo ha un metodo `calcolaInteresse()` diverso.

Nel main, calcola l'interesse per una lista di conti.

---

## 26. Gestione File Studenti (File I/O + ArrayList + Parsing)

Leggi un file `.txt` contenente dati di studenti (`nome`, `voto`)

Nel main, calcola la media voti e salva il risultato in un altro file.

---

## 27. Parser JSON Manuale (Stringhe + Logica + ArrayList)

Nel main, ricevi da `args` una stringa stile JSON semplice (`{"nome": "Luca", "eta": 22}`)

Parsa e crea un oggetto `Persona`, senza usare librerie esterne.

---

## 28. Magazzino Multiclasse (ArrayList + Ereditarietà + instanceof)

Crea `Prodotto`, e classi figlie `Alimentare`, `Elettronico`.

Nel main, aggiungi vari prodotti e stampa solo quelli di un certo tipo.

---

## 29. Gestione Playlist (ArrayList + Oggetti + Metodi Statici)

Classe `Canzone` con titolo, artista, durata.

Nel main, crea una playlist e calcola la durata totale tramite metodo statico.

---

### 30. Rubrica con Ricerca Avanzata (ArrayList + Stream-like)

Classe `Contatto` con nome, numero, email.

Nel main, implementa ricerche dinamiche per nome parziale o dominio email.

---

### 31. Sistema di Login (HashMap + Stringhe + Sicurezza base)

Nel main, simula un sistema di login con utenti memorizzati in un `HashMap<username, password>`

Verifica se le credenziali inserite da `args` sono corrette.

---

### 32. File Logger (FileWriter + Singleton + Data)

Implementa un singleton `Logger` che scrive log su file con timestamp.

Nel main, genera log da eventi diversi.

---

### 33. Parcheggio (OOP + Array + Occupazione posti)

Classi `Auto`, `PostoAuto`, `Parcheggio`.

Nel main, simula entrate/uscite e stampa lo stato del parcheggio.

---

### 34. Classifica Mondiale (Ereditarietà + Comparable)

Classe `SquadraNazionale` implementa `Comparable` per ordinamento per punti.

Nel main, stampa la classifica ordinata.

---

### 35. Gioco dei Dadi (OOP + Random + Logica)

Classe `Giocatore` con nome e punteggio.

Nel main, due giocatori lanciano 3 dadi per turno. Vince chi ha punteggio più alto dopo 5 turni.

---

### 36. Libreria con Ricerca Multi-criterio (ArrayList + Filtri Combinati)



Classe **Libro** con titolo, autore, genere, anno.

Nel main, filtra libri per più criteri contemporaneamente (ad es. autore + genere).

---

### 37. Analisi Statistica Voti (Array + Metodi Statici + Media/Moda/Mediana)

Dati un array di voti, crea metodi statici per calcolare media, moda, e mediana.

---

### 38. Simulazione Metereologica (Classi + ArrayList + Random + Date)

Classe **Previsione** con temperatura, data, stato meteo.

Nel main, genera 10 giorni casuali di meteo e stampa il più caldo/freddo.

---

### 39. Task Manager (OOP + Priorità + ArrayList)

Classe **Task** con descrizione, priorità, scadenza.

Nel main, ordina i task per priorità e scadenza.

---

### 40. Mini ORM (Riflessione + Classi + HashMap)

Crea un sistema base per convertire oggetti in **HashMap<String, Object>** e viceversa.

Nel main, salva e carica una **Persona** come mappa di attributi.

---

Livello: intermedio

## 1. Gestione di un Negozio Online (OOP + ArrayList)

Crea le classi **Prodotto**, **Cliente** e **Ordine**. Utilizza un **ArrayList** per gestire gli ordini di ciascun cliente. Nel **main**, consenti l'aggiunta di prodotti al carrello e la visualizzazione del totale dell'ordine.

---

## 2. Sistema di Biblioteca (Ereditarietà + Interfacce)

Implementa una classe astratta **MaterialeBibliotecario** con sottoclassi **Libro** e **Rivista**. Crea un'interfaccia **Prestabile** con metodi per il prestito e la restituzione. Nel **main**, gestisci il prestito e la restituzione di materiali.

---

### 3. Rubrica Telefonica (ArrayList + Ricerca)

Crea una classe `Contatto` con nome, numero di telefono ed email. Utilizza un `ArrayList` per memorizzare i contatti. Nel `main`, consenti l'aggiunta, la rimozione e la ricerca di contatti per nome.

---

### 4. Calcolatrice Avanzata (Classi Astratte + Polimorfismo)

Definisci una classe astratta `Operazione` con un metodo `calcola()`. Implementa sottoclassi per le operazioni di somma, sottrazione, moltiplicazione e divisione. Nel `main`, esegui operazioni basate sull'input dell'utente.

---

### 5. Gestione di una Scuola (Ereditarietà + Array)

Crea una classe `Persona` con sottoclassi `Studente` e `Insegnante`. Utilizza un array per memorizzare gli studenti e gli insegnanti. Nel `main`, consenti l'aggiunta e la visualizzazione delle informazioni delle persone.

---

### 6. Sistema di Prenotazione (Interfacce + ArrayList)

Definisci un'interfaccia `Prenotabile` con metodi per prenotare e cancellare. Implementa classi `VoLo` e `Hotel` che implementano l'interfaccia. Nel `main`, gestisci le prenotazioni per voli e hotel.

---

### 7. Gestione di un Magazzino (Generics + HashMap)

Crea una classe generica `Magazzino<T>` che utilizza un `HashMap` per memorizzare gli oggetti e le loro quantità. Nel `main`, gestisci l'aggiunta e la rimozione di prodotti dal magazzino.

---

### 8. Sistema di Votazione (Array + Statistiche)

Crea una classe `Votazione` che utilizza un array per memorizzare i voti. Implementa metodi per calcolare la media, il voto massimo e minimo. Nel `main`, consenti l'inserimento dei voti e visualizza le statistiche.

---

## 9. Gestione di Eventi (ArrayList + Date)

Crea una classe `Evento` con nome, data e luogo. Utilizza un `ArrayList` per memorizzare gli eventi. Nel `main`, consenti l'aggiunta e la visualizzazione degli eventi in ordine cronologico.

---

## 10. Sistema di Pagamenti (Interfacce + Polimorfismo)

Definisci un'interfaccia `MetodoPagamento` con un metodo `paga()`. Implementa classi `CartaDiCredito` e `PayPal` che implementano l'interfaccia. Nel `main`, esegui pagamenti utilizzando diversi metodi.

---

## 11. Gestione di una Playlist (ArrayList + File I/O)

Crea una classe `Canzone` con titolo, artista e durata. Utilizza un `ArrayList` per memorizzare le canzoni. Nel `main`, consenti l'aggiunta di canzoni e salva/carica la playlist da un file.

---

## 12. Sistema di Login (HashMap + Sicurezza)

Crea una classe `Utente` con username e password. Utilizza un `HashMap` per memorizzare gli utenti. Nel `main`, consenti la registrazione e il login degli utenti, verificando le credenziali.

---

## 13. Gestione di una Biblioteca (Array + Ricerca)

Crea una classe `Libro` con titolo, autore e ISBN. Utilizza un array per memorizzare i libri. Nel `main`, consenti l'aggiunta e la ricerca di libri per titolo o autore.

---

## 14. Sistema di Fatturazione (Classi + Calcoli)

Crea una classe `Fattura` con cliente, elenco di prodotti e totale. Nel `main`, consenti la creazione di fatture e il calcolo del totale con IVA.

---

## 15. Gestione di un Calendario (ArrayList + Date)

Crea una classe `Appuntamento` con data, ora e descrizione. Utilizza un `ArrayList` per memorizzare gli appuntamenti. Nel `main`, consenti l'aggiunta e la visualizzazione degli appuntamenti.

Livello: intermedio-scomodo

## 16. Sistema di Inventario (Generics + ArrayList)

Crea una classe generica `Inventario<T>` che memorizza oggetti con quantità. Aggiungi metodi per inserire, rimuovere e cercare elementi. Nel `main`, crea un inventario di prodotti e uno di strumenti.

---

## 17. Simulazione di Gara Sportiva (OOP + Ordinamento)

Crea una classe `Atleta` con tempo di gara. Memorizzali in un `ArrayList`, ordina per tempo crescente. Nel `main`, stampa la classifica finale.

---

## 18. Analizzatore di Testo (File I/O + Map + Frequenze)

Leggi un file di testo, conta quante volte compare ogni parola (case insensitive). Usa una `HashMap<String, Integer>`. Nel `main`, stampa le parole ordinate per frequenza.

---

## 19. Sistema di Sondaggi (Interfacce + Statistiche)

Definisci un'interfaccia `Domanda` con un metodo `rispondi()`. Crea classi `DomandaSiNo`, `DomandaScala` (1–5). Nel `main`, raccogli e analizza le risposte di un sondaggio.

---

## 20. Gioco "Battaglia Navale" semplificato (Array + Logica)

Crea una griglia 5x5 con alcune navi nascoste.  
L'utente inserisce coordinate e riceve esito (colpito o mancato).  
Il gioco termina quando tutte le navi sono affondate.

---

## 21. Registro Presenze Studenti (ArrayList + Map)

Gestisci un registro dove ogni `Studente` ha un elenco di date di presenza.  
Usa una `Map<Studente, List<LocalDate>>`.  
Nel `main`, aggiungi presenze e stampa il riepilogo.

---

## 22. Convertitore di Valute (Enum + Interfaccia)

Definisci un `enum` con valute e relativi tassi di cambio rispetto all'euro.  
Crea un metodo per convertire da una valuta all'altra.  
Nel `main`, chiedi importo e valuta di destinazione.

---

## 23. Mini CRM (Clienti, Interazioni, Appuntamenti)

Crea le classi `Cliente`, `Interazione`, `Appuntamento`.  
Collega ogni cliente a un elenco di interazioni.  
Nel `main`, gestisci l'inserimento e la visualizzazione delle relazioni.

---

## 24. Sistema di Statistiche Universitarie (Classi + Calcoli)

Gestisci una lista di `Studente` con media, esami sostenuti.  
Calcola media generale, numero studenti sopra/sotto soglia.  
Nel `main`, leggi dati da `args` o da input testuale.

---

## 25. Generatore di Codici Fiscali (Stringhe + Calcoli)

Crea una classe `CodiceFiscaleGenerator` che, dato nome, cognome, data e luogo di nascita, genera un codice fiscale simulato.  
Nel `main`, mostra il codice per più persone.

---

## 26. Gestione Portafoglio Azionario (OOP + Map)

Gestisci azioni tramite una `Map<String, Integer>` (simbolo, quantità).  
Implementa acquisto, vendita e visualizzazione del valore totale.  
Nel `main`, simula un paio di transazioni.

---

## 27. Simulazione Parcheggio (Array + Stato)

Rappresenta un parcheggio a 10 posti come array.  
Ogni posto può essere libero o occupato.  
Nel `main`, consenti ingresso e uscita auto.

---

## 28. Gestione Promozioni di un Supermercato (Enum + Strategie)

Usa un `enum` per tipo di promozione (`SCONTO`, `2x1`, `PUNTI`).  
Applica diverse strategie di calcolo su un carrello.  
Nel `main`, stampa il totale e la strategia applicata.

---

## 29. Parser CSV (File I/O + Parsing + OOP)

Leggi un file CSV e crea oggetti da ogni riga (es. studenti, prodotti).  
Mostra i dati in tabella nel `main`.

---

## 30. Simulazione Distributore Automatico (ArrayList + Interazione)

Crea un sistema per selezionare prodotti, inserire soldi, e ricevere resto.  
Usa una lista di oggetti `Prodotto`.  
Nel `main`, simula un acquisto completo.

---

## 31. Gestione Dipendenti (Ereditarietà + Calcolo Stipendi)

Definisci classi `Dipendente`, `Impiegato`, `Dirigente`, ognuna con un metodo per calcolare lo stipendio mensile.  
Nel `main`, gestisci una lista e calcola la spesa totale.

---

## 32. Sistema di Allarme (Observer pattern semplificato)

Crea un sistema dove più sensori (fumo, movimento) notificano un gestore di allarme.  
Nel `main`, attiva/disattiva i sensori e ricevi notifiche.

---

### 33. Simulatore Meteo (Random + OOP + Enum)

Crea una classe `Previsione` che contiene temperatura, umidità, e condizioni (`enum`).  
Simula le previsioni per 7 giorni e stampa grafico testuale.

---

### 34. Simulazione Gioco Dadi (Random + Statistiche)

Simula lanci di due dadi 100 volte, calcola media, min, max, istogramma.  
Nel `main`, stampa i risultati in formato leggibile.

---

### 35. Creazione Curriculum (OOP + StringBuilder)

Crea classi `EsperienzaLavorativa`, `Istruzione`, `Lingua`.  
Usa queste per generare un curriculum ben formattato come stringa.  
Nel `main`, stampa un esempio completo.

## Livello: complesso/micro progetti

### 1. Mini Database Relazionale (OOP + HashMap + Join Simulati)

Costruisci un mini-DB con tabelle simulate da `HashMap<String, ArrayList<Record>>`.  
Ogni `Record` è una `Map<String, Object>`.  
Nel `main`, consenti: creazione tabelle, inserimento, `select`, e `join` semplificato.

---

### 2. Analizzatore di Dipendenze tra Classi (Reflection + Graph + DFS)

Dai in input una lista di classi.  
Costruisci un grafo delle dipendenze tra di esse tramite `java.lang.reflect`, e rileva eventuali cicli (dipendenza circolare).

---

### 3. Sistema di Cache a Due Livelli (Design Pattern + LinkedHashMap + Eviction)

Implementa una cache con due livelli (es. L1 in RAM, L2 su disco).

Usa `LinkedHashMap` per LRU, serializza/deserializza su file per il livello L2.

Testa con accessi nel `main`.

---

### 4. Compilatore Matematico Semplice (Parser + Tree + Ricorsione)

Crea un parser che converte un'espressione come `"3*(2+4)"` in un albero binario.

Esegui la valutazione ricorsivamente.

Bonus: aggiungi variabili (`x=5`, `y=2`, ecc.).

---

### 5. Chat Server/Client (Socket + Thread + Comunicazione)

Crea un'applicazione di chat: un server e più client.

Usa `ServerSocket` + `Socket` + `Thread` per ogni client.

Il main lancia uno dei due ruoli.

---

### 6. Microframework Web (HTTP + Sockets + Routing)

Implementa un mini web server in Java: parse HTTP, supporta `GET` e `POST`, routing base (`/home`, `/login`).

Il main avvia il server su una porta specifica.

---

### 7. Generatore di Codice Java (Reflection + Template)

Crea un generatore di classi Java a partire da un modello JSON o XML (es.

```
{"nome": "String", "eta": "int"}).
```

Output: file `.java` con attributi e metodi getter/setter.

---

### 8. Simulatore di Scheduler CPU (Thread + Strategia + Sincronizzazione)

Simula round-robin, FIFO e priority scheduling.

Ogni `Processo` è un thread che "consuma tempo".

Nel main, cambia algoritmo e osserva i risultati.



---

## 9. File System Virtuale (Composite Pattern + Navigazione + Ricorsione)

Modella un file system con `Directory` e `File` usando il pattern Composite.

Nel main, implementa comandi tipo `ls`, `cd`, `mkdir`, `touch`.

---

## 10. ORM Base con JDBC (Database + DAO + Reflection + Generics)

Implementa una mini ORM per SQLite:

`GenericDao<T>` che salva/carica oggetti usando riflessione e JDBC.

Nel main, inserisci e leggi oggetti da DB.

---

## 11. Algoritmi di Compressione (Huffman + File I/O + Bit Manipulation)

Implementa compressione Huffman:

- Leggi file
- Costruisci albero
- Codifica/decodifica

Nel main, comprimi e decomprimi un file di testo.

---

## 12. Sistema Event-Driven (Observer + Event Loop + Lambda)

Crea un sistema di eventi (click, hover, timer).

Usa il pattern Observer e lambda per registrare callback.

Nel main, simula eventi e listener.

---

## 13. Interprete Lisp Semplificato (Parser + Stack + Ricorsione)

Implementa un interprete per un subset di Lisp:

espressioni come `(add 1 (mul 2 3))`.

Nel main, esegui parsing e calcolo.

---

## 14. Tool di Testing Automatico (Reflection + Annotations)

Crea un mini framework stile JUnit.

Riconosci metodi annotati con `@Test`, esegui dinamicamente, mostra risultati nel main.

---

## 15. Analisi Statica del Codice (File + Regex + Parsing)

Crea un analizzatore per codice Java:

Conta classi, metodi, righe di codice, commenti.

Nel main, analizza una directory ricorsivamente.

---

## 16. Simulatore di Reti Neurali Feedforward (Math + Matrix + OOP)

Costruisci una rete neurale semplice (1 hidden layer)

Implementa: forward propagation, activation function, training base con backpropagation.

---

## 17. Editor di Testo da Terminale (Input + Buffer + Undo/Redo)

Implementa un editor base:

Inserimento, cancellazione, spostamento cursore, undo/redo (stack).

Nel main, simula operazioni da una lista comandi.

---

## 18. Motore di Regole Dinamico (DSL + Parser + Strategy Pattern)

Permetti di definire regole tipo:

`SE eta > 18 ALLORA "adulto"`

Parse e applica dinamicamente a oggetti.

Nel main, carica regole da file e applicale.

---

## 19. Simulatore di Rete Peer-to-Peer (Sockets + Hashing + Bilanciamento)

Crea nodi che comunicano tra loro e scambiano file.

Usa hash per identificare file e peer.

Nel main, lancia più nodi in porte diverse.

---

## 20. Algoritmi di Pathfinding (Graph + A + Visualizzazione Testuale)\*

Costruisci una griglia 2D con ostacoli.  
Implementa A\* per trovare il percorso da A a B.  
Nel main, stampa la mappa con percorso calcolato.