

## Progetto di esempio

Si scriva un'applicazione Java che implementa il seguente progetto.

### **Descrizione del problema**

Si vuol realizzare un'applicazione che gestisca un archivio degli studenti. Uno studente è caratterizzato da nome, cognome, matricola, data di nascita e luogo di nascita.

### **Funzionalità**

L'applicazione dovrà permettere di

- Riempire automaticamente l'archivio con alcuni dati, in modo da poterlo testare
- Inserire e rimuovere uno studente
- Stampare a video tutti gli studenti
- Ricercare gli studenti nati in certo luogo

L'applicazione dovrà possedere un menu che permetta di scegliere fra le varie funzionalità disponibili.

### **Vincoli da rispettare**

Per memorizzare l'insieme degli studenti si dovrà usare una struttura dati di tipo vettore.

## Documentazione

L'applicazione sviluppata è composta da tre classi: **Studente**, **Archivio** e **Menu**.

- **Studente**  
Contiene i dati di un singolo studente. Ha dei metodi per rappresentare uno studente attraverso una stringa e per verificare se due studenti sono uguali.
- **Archivio**  
Contiene un array di studenti e implementa le funzionalità richieste dal progetto: inserimento e rimozione di uno studente; ricerca di uno studente per luogo di nascita; restituzione di tutti gli studenti. L'inserimento è stato implementato in due metodi diversi che prendono in ingresso un oggetto studente oppure tutti le variabili di uno studente. La rimozione è stata implementata con tre metodi diversi che rimuovono lo studente usando la matricola, la posizione occupata nell'array e l'oggetto studente.
- **Menu**  
Visualizza il menu con le funzionalità richieste. Contiene un oggetto archivio che usa per memorizzare i dati. Contiene metodi corrispondenti alle funzionalità offerte più alcuni metodi di utilità (vedi sotto).

Una documentazione più dettagliata sulle classi è disponibile nei documenti javadoc allegati e generati automaticamente con Jbuilder

### Osservazioni

- Per quanto riguarda l'allocazione degli studenti nel vettore si è scelto di usare un'allocazione sequenziale con una variante rispetto a quanto visto a lezione. Uno studente può trovarsi in una qualsiasi posizione dell'array, ma si fa in modo che le prime posizioni dell'array siano tutte occupate, cioè non esista mai una posizione vuota seguita da posizioni che contengono studenti. In pratica, l'inserimento avviene come in una pila: si mantiene una variabile che indica la prima posizione libera e si inserisce lo studente in tale posizione. Per la rimozione, si sposta l'ultimo studente dell'array al posto di quello da rimuovere: in questo modo anche dopo le rimozioni le prime posizioni dell'array rimangono tutte piene. Questa soluzione rende l'inserimento più veloce.
- Nella classe menù alcune operazioni sono eseguite più volte in punti diversi del programma: la scrittura di un messaggio a video, la richiesta di conferma di una operazione, la lettura di una stringa da tastiera. Per rendere il progetto più modulare si è scelto di scrivere tre metodi che implementano tali operazioni e poi richiamarli quando necessario. Inoltre, si è scelto di creare dei metodi anche per operazioni piuttosto lunghe che erano però delle funzioni logicamente unitarie: la stampa del menu, l'esecuzione dell'interfaccia. Questa soluzione rende il codice più facile da leggere.