Esercizio pratico

Si vuole creare un programma per gestire **3** code per l’entrata ad una fiera. Nella prima coda possono accedervi persone **senza l’entrata prioritaria o con l’entrata prioritaria**. Invece nella seconda coda vi possono accedere **solamente** quelli con l’**entrata prioritaria**. Nella terza vi possono **solo** accedere i **VIP**.

Si vuole memorizzare ogni persona che lascia una delle due code in una struttura dati per poi calcolare alcune statistiche (quanti maschi/femmine c’erano, quanti avevano la priorità ecc…) e che possa ricercare una determinata persona.

Una persona è composta da nome, cognome, sesso, età. (solo questi attributi deve avere)

Scrivere quale struttura dati si ha intenzione di utilizzare per memorizzare le persone e motivarne la scelta.

Esercizio teoria

|  |  |  |  |
| --- | --- | --- | --- |
| Collection | Superclasse da cui derivano | Pro | Contro |
| ArrayList | AbstractList |  |  |
| LinkedList | AbstractSequenceList |  |  |
| PriorityQueue | AbstractQueue |  |  |
| HashSet | AbstractSet<T> |  |  |
| TreeSet | AbstractSet<T> |  |  |
| HashMap | AbstractMap<K,T> | Veloce nel inserimento e nel eliminazione | Difficile scambiare gli ordini |
| TreeMap | AbstractMap<K,T> | Elementi sempre ordinati |  |

Spiega il significato: <T extends S>.

Il tipo generico T deve implementare la classe o l’interfaccia S

Spiega il significato: <?>

Questo indica un tipo senza limiti

Nella rimozione di un oggetto è più performante un contenitore di tipo ArrayList o LinkedList?

La LinkedList è più performante perché la rimozione di un solo oggetto della lista basta semplicemente cambiare i soli riferimenti.