

Fondamenti di Informatica 1 - anno accademico 200? - 200?
Prova di Programmazione, appello simulato

Si vuole realizzare un programma che simula il comportamento di una biglietteria composta da una sequenza di N sportelli. Presso ogni sportello si forma una diversa coda di persone in attesa di essere servite; una persona che arriva presso la biglietteria si accoda allo sportello con la coda più breve; ogni sportello serve la prima persona della propria coda. Questo comportamento viene rappresentato dal tipo di dato astratto “multicoda” (multiqueue).

Il candidato ha a disposizione il file **MultiQueueTester.java**, il quale contiene classi ed interfacce da utilizzare e/o completare secondo quanto qui indicato:

- **Queue** ed **EmptyQueueException**. Rispettivamente, interfaccia che rappresenta il tipo di dati astratto “coda”, ed eccezione che segnala che una coda è vuota. *Non modificare.*
- **MultiQueue**. Interfaccia che rappresenta il tipo di dati astratto “multicoda”. *Leggere con attenzione* i commenti all’interfaccia ed ai suoi metodi. *Non modificare.*
- **ArrayQueue**. Classe che implementa **Queue** usando un array. *Completare la classe* scrivendone la parte privata e realizzandone i metodi pubblici. La classe deve inoltre *sovrascrivere il metodo toString*, che restituisca una stringa contenente gli elementi della coda, ciascuno su una riga diversa.
- **ArrayMultiQueue**. Classe che implementa l’interfaccia **MultiQueue** usando un array di code. *Completare la classe* scrivendone la parte privata e realizzandone i metodi pubblici. La classe deve realizzare un costruttore che riceve un parametro esplicito intero $N > 0$ e crea una multicoda vuota, costituita da una sequenza di N code vuote. La classe deve inoltre *sovrascrivere il metodo toString*, che restituisca una stringa contenente gli elementi delle N code della multicoda, secondo il seguente formato: (1) gli elementi delle N code vengono scritti in sequenza, secondo il formato del metodo toString della classe **ArrayQueue** (si veda il corpo di tale metodo), (2) la scrittura degli elementi della i -esima coda ($0 \leq i < N$) comincia con la riga “CODA i :”.
- **MultiQueueTester**. Classe di collaudo. *Completare la classe* in modo che il suo metodo main realizzi il seguente comportamento:
 - riceve un numero intero N come argomento sulla riga di comando;
 - crea un oggetto di tipo **ArrayMultiQueue**, contenente N code, inizialmente vuote;
 - accetta ripetutamente comandi dall’utente, introdotti da tastiera, finchè l’utente non introduce il comando di terminazione del programma; i comandi disponibili sono:

Comando	Significato
“A”	Aggiunge una persona alla multicoda. In questo caso il programma chiede di inserire da tastiera una stringa rappresentante il nome della persona, e la inserisce nella biglietteria.
“R”	Rimuove una persona dalla multicoda. In questo caso il programma chiede di inserire da tastiera un numero intero rappresentante l’indice della coda da cui effettuare la rimozione, rimuove la persona dalla coda corrispondente, e ne stampa il nome a standard output.
“P”	Stampa (Print) a standard output il contenuto aggiornato della multicoda.
“Q”	Termina (Quit) il programma.

Nello svolgimento si possono usare, se ritenute utili, solo le seguenti classi della libreria standard:

- tutte le classi dei package java.lang e java.io;
- le classi Scanner e NoSuchElementException del package java.util.

Usando il file **input.txt** allegato, il programma può essere provato con il comando

java MultiQueueTester 4 < input.txt

Al termine della prova lasciare nella directory di lavoro tutti i file. La prima riga del file .java dovrà contenere un commento con nome/cognome del candidato, matricola, data, numero postazione. Prima di uscire dall’aula consegnare questa scheda compilata.

Cognome e Nome:	Corso di Laurea:
Matricola:	Postazione: ADT
Consegno l’elaborato.	Non consegno l’elaborato e mi ritiro dall’esame.
Firma	Firma