Simulazione d’esame dell’ 11 Gennaio 2022

Esercizio 1 prova di programmazione

Implementare una classe eseguibile il cui nome deve essere CognomeMatricolaRic.java, dove Cognome e Matricola devono essere il proprio nome e la propria matricola.

Importante: nella prima riga del file scrivere in un commento il proprio nome, cognome, numero di matricola.

Nel main si dovrà:

- chiedere all'utente di inserire da standard input una stringa che potrà contenere lettere e numeri
- invocare un metodo statico ricorsivo (che dovrete implementare voi) che calcoli la somma dei caratteri che corrispondono a cifre
- stampare in output il risultato
-

Di seguito trovate alcuni esempi di stringhe e dei risultati attesi:
La somma delle cifre in CI12A3o4 è 10
La somma delle cifre in ab3cd3ef3gh3 è 12
La somma delle cifre in abcdefg è 0
La somma delle cifre in 12345 è 15
La somma delle cifre in 00000 è 0

La documentazione java può essere consultata. E' ammesso l'uso di java.util.Scanner, dei metodi della classe String e della classe Character. Se in dubbio, chiedete alla docente o ai sorveglianti se è ammesso l’utilizzo di una particolare classe e/o metodo.

Esercizio 2 prova di programmazione

Sia data un'implementazione di un ADT coda come coda circolare su array a dimensione fissa. **Implementare:**

- **una sua sottoclasse**, chiamata **Sportello**, con i metodi sotto definiti. La classe simula la coda a uno sportello delle poste. I valori in essa contenuti sono i minuti necessari per sbrigare la pratica del cliente corrispondente. Ad esempio, in una coda con valori 3 15 4 ci sono tre clienti, il primo ha una pratica che dura 3 minuti, il secondo 15 minuti e il terzo 4 minuti.
- **una classe PostalOfficeCognomeMatricola (dove "CognomeMatricola" va sostituito con i propri dati) che la utilizza** secondo le indicazioni date più sotto.

Oltre alla classe ***FixedCircularArrayQueue.java*** viene fornita la classe ***SportelloTester.class*** che permette di fare alcuni test sulla classe sportello. I test non sono esaustivi: se tutto funziona non è detto che tutto sia giusto, ma se qualcosa non funziona, sicuramente c'è un errore.

IMPORTANTE: nella prima riga di tutte le classi realizzate inserite anche un commento con cognome, nome, numero di matricola.

Metodi della classe Sportello, sottoclasse di FixedCircularArrayQueue:

/* Metodo che restituisce il numero di elementi presenti nella coda.
NB: potete elaborare questa informazione dalle variabili d'istanza della coda, oppure potete aggiungere una variabile d'istanza nella classe Sportello che tenga conto della numerosità (in questo caso ricordarsi di sovrascrivere, ove necessario, costruttori e/o metodi in modo che il suo valore sia sempre aggiornato). Entrambe le scelte progettuali vanno bene e verranno valutate allo stesso modo.
public int getSize()

/* Il metodo dequeue decrementa il valore in testa alla coda di una unità. Se il valore ottenuto è 0, allora lo elimina effettivamente dalla coda. Altrimenti lo lascia nella sua posizione, decrementato. Ad esempio, se nella coda ci sono gli elementi 3 1 6, dopo l'invocazione di dequeue il contenuto sarà 2 1 6. Se nella coda ci sono gli elementi 1 3 6, dopo l'invocazione di dequeue il contenuto sarà 3 6.
public Object dequeue()

/* Restituisce una stringa che riporta il numero di elementi nella coda, seguiti dalla dicitura " clienti: " seguita dal contenuto della coda (si veda il metodo toString della superclasse). Ad esempio, se la coda contiene gli elementi 3 1 6 la stringa restituita sarà: "3 clienti: 3 1 6"
public String toString()

Classe PostalOfficeCognomeMatricola

- Creare un array di 5 sportelli e inizializzarli
- Creare una variabile Random che abbia come **seed il numero 123.**
- Simulare il trascorrere di un numero ***n*** di minuti di tempo, come descritto sotto. Il valore n è letto da riga di comando quando viene eseguita la classe. Se non riuscite o non vi ricordate come leggere da riga di comando, potete fissare n nel codice (perderete qualche punto ma potete implementare e testare il resto).

Ad ogni minuto trascorso succedono le seguenti cose:

- arriva un nuovo cliente che sarà rappresentato da un tempo di attesa (int) generato come numero intero casuale da 1 a 15.
- viene individuato l'indice, nell'array inizialmente creato, dello sportello con il **minor numero di clienti** in coda. Questo avviene attraverso la chiamata a un metodo statico, da implementare, con firma

  public static int minSizeQueue(Sportello[] p)

- Si inserisce il nuovo cliente (ovvero il valore del suo tempo di attesa generato prima in modo random) nella coda dello sportello individuato
- Si fa trascorrere un minuto in tutte le code (ovvero si invoca dequeue su ciascuno Sportello dell'array) stampando per ciascuna il proprio stato. Ad esempio, alla prima iterazione il numero random generato è 3 e lo sportello individuato come quello con il minor numero di elementi è lo sportello di indice 0 (ovvero il primo sportello). Si dovrebbe stampare:
  Nuovo cliente con pratica 3 minuti inserito in coda allo sportello 1
  Situazione code dopo 1 minuti
  Sportello 1 : 1 clienti 2
  Sportello 2 : 0 clienti
  Sportello 3 : 0 clienti
  Sportello 4 : 0 clienti
  Sportello 5 : 0 clienti

Si riportano altri esempi di output dopo 10 e 20 minuti, utili come controllo:

Nuovo cliente con pratica 14 minuti inserito in coda allo sportello 1
Situazione code dopo 10 minuti
Sportello 1 : 2 clienti 8 14
Sportello 2 : 1 clienti 5
Sportello 3 : 1 clienti 4
Sportello 4 : 1 clienti 4
Sportello 5 : 1 clienti 6

Nuovo cliente con pratica 7 minuti inserito in coda allo sportello 5
Situazione code dopo 20 minuti
Sportello 1 : 2 clienti 12 10
Sportello 2 : 2 clienti 10 10
Sportello 3 : 2 clienti 1 8
Sportello 4 : 2 clienti 3 6
Sportello 5 : 2 clienti 2 7