Analisi casse

## Struttura menù:

La struttura del menù riassume le funzioni offerte dal programma:

### Menù principale:

1. Esci (Chiude il programma).
2. Simula (Simulazione cassa che continua nel tempo in automatico).

### menù esercizio 1 (automatico):

Modalità di esecuzione del programma in base al momento del giorno selezionato:

1. Mattina-Pomeriggio.
2. Notte.
3. Orario di punta.

# punti chiave delle casse:

* Ci sono 3 casse di cui una aperta.
* Le code sono strutture FIFO (First-In-First-Out).
* La seconda cassa viene aperta quando si raggiungono 15 clienti sulla prima in coda e ci confluiscono i clienti in posizione pari rispetto alla prima.
* Se una delle due code raggiunge nuovamente 15 clienti, si apre la terza cassa in cui confluiscono i clienti over-70 anni.
* Probabilità di uscita di un cliente rispetto a uno entrante (che nella simulazione automatica dipende dal periodo del giorno).
  + Mattina-Pomeriggio: 50% probabilità entrata e 50% uscita.
  + Notte: 80% uscita e 20% entrata.
  + Orario di punta: 80% entrata e 20% uscita.

## Funzionamento:

### Simulazione automatica:

Chiedo l’orario di simulazione, quanti “turni” si vogliano simulare e la velocità del programma, una volta avviato viene:

* Incrementato un contatore che quando raggiunte il numero di turni voluti, ferma il timer ed esce stampando la coda attuale.
* Ogni volta che entra una persona tramite random e probabilità del periodo del giorno, tramite IF decido in quale coda metterlo se attivo, Coda è un oggetto composto dal nome e le persone con le loro rispettive età.
* Quando una situazione particolare si verifica, come il raggiungimento di 15 persone, si apre una cassa, una volta aperta si presume non venga richiusa, e si leggono le persone nelle posizioni delle casse che erano già aperte, e si spostano in base alla posizione se pari, oppure se è la terza cassa, si spostano gli over 70 anni.
* Ad ogni “turno” stampo cos’è avvenuto, come se è entrata o uscita una persona (sempre randomico), insieme ad ulteriori informazioni utili.

### Struttura classe coda:

Contenuto:

* int maxLista (Dimensione massima della coda, può essere cambiata).
* Int[] lista (Le singole persone per età nel vettore).
* int età (Condizione speciale che viene usata se la cassa è dedicata a persone di una certa età).
* boolean etaMinima (Attiva la condizione specificata prima di età minima).
* boolean attiva (Ritorna true o false se la cassa è attiva).
* String nomeLista (Nome della cassa).

NB: Al posto di un vettore potrei anche usare una queue che è una lista FIFO integrata in Java, ma ho comunque creato la mia per “farmi del male”.

La Coda della cassa contiene anche diversi metodi:

* isAttiva (ritorna se la cassa è attiva o meno).
* numeroPersone (ritorna il numero di persone).
* Inizializza (Inizializza la cassa e la coda).
* Diversi getter e setter per modificare direttamente il suo contenuto.
* addElemento (Aggiunge una persona alla coda in ultima posizione).
* removeElemento (Rimuove la persona in prima posizione e trasla tutte le altre per colmare il vuoto).
* isEtaMinima (Dando in input l’età di qualcuno, ritorna un booleano che comunica se una persona ha l’età minima per la cassa in questione se con tale condizione attiva).

## eventi:

### Se entra una persona

Aggiungo la persona appena entrata alla prima cassa, se questa raggiunge 15 persone, si attiva l’evento successivo che attiva un’altra cassa o le sposta in base alla posizione e l’età.

### Se una coda/Cassa ha più di 15 persone

Apro una nuova cassa se non già aperta, e sposto tutte persone in posizioni pare o che soddisfano le condizioni in essa.

### Fine programma

Se si raggiunge il numero di persone prestabilito, stampo le code grezze con il loro contenuto, e anche il numero di persone in essa.

### se esce una persona

Viene rimossa dalla coda della cassa la persona, se la cassa ha 0 persone in coda, può essere chiusa.

# obiettivo:

In base all’orario e al numero di persone, queste devono uscire dalle casse nel minor tempo possibile, tramite questi “limiti o target”, la posizione e l’età, si migliorano i tempi.

## configurabilità:

Le casse funzionano tramite oggetti, le loro caratteristiche vengono caricate tramite file casse.json, le quali contengono dati come:

#### Per ogni singola cassa:

* Età minima attiva (vero o falso).
* Età minima per mettersi in coda (se attiva modalità precedente).
* Nome cassa (Una stringa che è possibile impostare per dare un nome specifico alla cassa).
* Numero massimo accettabili in coda per cassa, oltre il quale saranno “SATURE” e non accetteranno ulteriori persone in coda.

#### GLOBALI:

* Numero casse.
* Numero di persone a cui si attiva la successiva cassa.

## Gestione dei file tramite software:

Al menu sarà aggiunta una sezione “configurazione” con le seguenti opzioni:

1. Esci e salva.
2. Visualizza informazioni globali.
3. Visualizza informazioni casse.
4. Modifica impostazioni globali.
5. Modifica impostazioni casse.

Scegliendo le varie opzioni, sarà possibile eseguire delle modifiche.

Quelle di modifica, aprono dei sotto-menu con ulteriori opzioni, e quella delle singole casse propone la scelta di una cassa da modificare.

L’opzione di modifica permette di cambiare le impostazioni globali (numero casse e numero persone che attivano la successiva cassa), e opzioni per cassa (Età minima se necessaria per entrare in cassa, l’età necessaria, nome cassa, numero massimo di persone accettabili prima che si ingolfi).

# domande:

* A cosa serve questa analisi?
* Perché qualcuno (e chi) userà questo programma?
* Come userà questo programma?