

REPORT

Elena Bongiovanni s349131, Elena Perotti s345824, Ludovica Scaffidi s343272

Nello svolgimento dell'homework abbiamo deciso di implementare, oltre alla simulazione del benzinaio, la simulazione dello stand dei panini visto a lezione.

L'obiettivo dell'homework era quello di implementare due simulazioni diverse cercando di fattorizzare tutto ciò che è riutilizzabile. Per questo motivo, abbiamo creato delle classi astratte che abbiamo utilizzato in entrambe le simulazioni.

CLASSE SIMULATION

Innanzitutto è stata creata la classe Simulation, che rappresenta la struttura di base delle nostre simulazioni, a cui abbiamo assegnato le seguenti properties:

- ClientQueue, ovvero la coda dei clienti;
- Clock, cioè l'orologio che tiene traccia del tempo in cui succedono i diversi eventi nella simulazione;
- ToServe, che indica il numero di clienti da servire;
- WaitingTimeQueue;
- AvgLengthQueue.

e come metodi astratti:

- StartSimulation, che implementa una specifica simulazione;
- WriteResults, che si occupa di riportare i risultati della simulazione in un file di testo.

Abbiamo poi creato le due classi:

- SimulationRollQueue;
- SimulationPetrolStation.

che, rispettivamente per le due simulazioni, estendono la classe Simulation. In queste classi abbiamo implementato gli override dei due metodi astratti.

CLASSE EVENT

La classe Event è necessaria per tenere traccia e gestire tutti gli eventi all'interno della simulazione, come ad esempio l'arrivo di un cliente, o il completamento di un servizio. Le properties assegnate sono:

- Rate e Distribution, che si occupano di definire la distribuzione del nostro evento con rate associato;

- Next, che corrisponde al tempo di avvenimento;
- TimesList, che tiene traccia del susseguirsi degli eventi in una lista.

Alla classe sono associati diversi metodi che si occupano di gestire il flusso degli eventi:

- GenerateEvent, che genera l'istante in cui avrà luogo un nuovo evento;
- AddTime e RemoveTime che aggiungono e rimuovono tempi dalla TimeList.

E un metodo astratto

- Manage, che si occupa di gestire il singolo evento nella simulazione.

Le estensioni di questa classe costituiscono l'insieme delle classi per gli eventi specifici che possono avere luogo nelle due simulazioni, e sono le seguenti:

- ClienteArrivalStation;
- FuelServiceStation;
- CashServiceStation;
- ClientArrivalRoll;
- ServiceRoll.

CLASSE QUEUE

La classe Queue si occupa di gestire una eventuale coda all'interno della simulazione. Le sue properties sono:

- ClientsList: lista dei clienti in coda;
- NumInQueue: lunghezza coda ad ogni evento;
- Lost: numero di clienti persi perché impossibilitati ad aggiungersi alla coda;
- Served: numero clienti serviti;
- Blocked: numero di clienti che sono stati eventualmente bloccati nella coda;
- NumMax: limite massimo di persone in coda.

Alla classe sono stati anche associati due metodi:

- Update, che aggiorna lo stato della coda;
- AddServed, che si occupa di gestire i serviti.

CLASSE STATEVAR

La classe StateVar si occupa di gestire alcune statistiche utili alla nostra simulazione. Le sue properties sono:

- Times;
- CurrentState;
- Values.

Inoltre, le abbiamo associato i seguenti metodi astratti che servono per aggiornare e valutare le nostre statistiche:

- Update;
- EvaluateFinalState;

- CleanState.

Le due sottoclassi ottenute dall'estensione di questa classe stratta sono:

- WaitingTime: si occupa di gestire in maniera specifica le statistiche dei tempi di attesa;
- AvgLength: si occupa di gestire in maniera specifica le statistiche della lunghezza della coda.

CLASSE PUMPS

La classe Pumps è specifica per la simulazione del benzinaio, e si occupa di gestire l'assegnazione dei clienti alle pompe di servizio. Le sue properties sono:

- NumLines: numero delle file a cui è possibile fare benzina;
- NumPumps: numero delle pompe per ogni fila;
- PumpsList: Matrice di dimensione NumLines x NumPumps per le pompe a cui viene assegnato valore 1 se la pompa è occupata, 0 altrimenti;
- NumClients: numero clienti alle pompe;
- Block: variabile booleana utilizzata per valutare se una pompa è bloccata o meno.

CLASSE CLIENT

La classe client è una classe specifica per gestire il singolo cliente. Le properties assegnate sono:

- EnterTime: tempo di arrivo di un cliente;
- EnterBlock: eventuale tempo in cui il cliente viene bloccato durante la simulazione.

La classe è caratterizzata dal metodo

- Blocked: assegna valore alla property EnterBlock in caso di bloccaggio.

e dal metodo astratto

- CleanState.

Per la simulazione del benzinaio, questa classe astratta è stata estesa alla sottoclasse:

- Driver: associa ad ogni cliente il lato sul quale possono fare benzina, il numero della pompa, il tempo in cui finiscono di fare benzina e se stanno pagando o meno.

Infine, nel main abbiamo istanziato due oggetti, uno per ciascun tipo di simulazione, con i rispettivi parametri, e chiamato il metodo StartSimulation per entrambi. Nell'impostare i parametri, abbiamo considerato come unità di misura il minuto.

CONCLUSIONE

Abbiamo infine analizzato i risultati ottenuti tramite la simulazione del benzinaio nel file di testo finale results_PetrolStation. Si possono notare i buoni risultati in termini di tempi di attesa, specialmente per quanto riguarda i clienti che rimangono bloccati dalla macchina ferma davanti la loro e l'attesa in cassa, motivo per cui non sarebbe necessario in questo caso aggiungere un ulteriore cassiere.

Inoltre, il numero di clienti persi alla fine della simulazione è risultato pari a 2: quantitativo migliorabile sebbene poco rilevante rispetto al totale dei clienti serviti. Il numero di clienti bloccati in uscita, invece, è più elevato, ma anche questo risulta poco rilevante considerando il tempo di attesa medio trascurabile.

Queste considerazioni sono state effettuate in un contesto poco affidabile, in quanto è stata effettuata una sola simulazione nella quale abbiamo inizializzato un numero massimo di clienti da servire scelto da noi. L'ideale sarebbe effettuare più simulazioni indipendenti fra loro e impostare un criterio di stop basato, per esempio, sull'affidabilità di un intervallo di confidenza.