

## LABORATOR 10

Se presupune că la un supermarket sosesc clienți doar cu mașina, astfel încât sosirea clienților este distribuită exponențial, începând cu 600 de clienți pe oră, timp de o jumătate de oră, 900 de clienți pe oră, timp de o oră, 450 de clienți pe oră, timp de o oră și 300 de clienți pe oră după aceea.

Parcarea supermarketului are 650 de locuri. Dacă un client nu găsește loc de parcare, pleacă fără să-și facă cumpărăturile. Distanța de la locul de parcare, la intrarea în supermarket este parcursă de un client în 60 de secunde.

Clienții cumpără între 5 și 100 de produse, cu o distribuție uniformă. Un client care cumpără mai puțin de 10 produse va folosi un coș (există 70 de coșuri). Un client care cumpără mai mult de 10 produse va folosi un cărucior (există 650 de cărucioare). Timpul pe care îl petrece un client în magazin depinde de numărul de produse cumpărate (10 secunde pentru alegerea fiecărui produs).

Există 17 case, iar clienții care au cumpărat cel puțin 10 produse aleg casa cu coada cea mai mică. Clienții care cumpără mai puțin de 10 produse aleg o casă specială. Timpul petrecut la casă depinde de numărul de produse cumpărate (2 secunde pentru fiecare produs, plus un timp suplimentar de 25, 30 sau 35 de secunde). Timpul suplimentar depinde de modalitatea de plată (cu bani, cu cec sau cu card, aceste trei metode fiind egal probabile).

După ce a plătit, un client merge la mașina (60 de secunde), își pune produsele în mașină și părăsește parcarea.

Simulați modelul pentru 3 ore. Determinați distribuția timpului petrecut de clienți în sistem. Determinați utilizarea parcarii, a cărucioarelor, a coșurilor și a caselor. Determinați histograma numărului de clienți din sistem într-un interval de un minut.

### Indicații de rezolvare:

Prima parte a programului va consta din următoarele definiții și inițializări:

- o variabilă care să reprezinte cantitatea cumpărată (un număr aleator între 5 și 100);
- o variabilă care să reprezinte timpul suplimentar petrecut la case în funcție de modalitatea de plată (o valoare aleatoare aleasă din 25, 30, 35);
- graficul pentru timp;
- graficul pentru produsele cumpărate;
- graficul pentru numărul de clienți;

- entitățile STORAGE pentru parcare, coșuri și cărucioare;
- o variabilă care să reprezinte timpul de plată;
- o variabilă care să reprezinte durata alegerii produselor;
- inițializarea entității SAVEVALUE în care este memorat numărul de clienți.

În a doua parte a programului se vor genera clienții:

- Timpul se consideră exprimat în secunde.
- Pentru că frecvențele de sosire diferă pentru diferite intervale de timp, vor exista patru blocuri GENERATE, generând nu clienți diferiți, ci același tip de clienți, dar cu frecvențe de sosire diferite. De exemplu, blocul GENERATE pentru prima jumătate de oră va fi de forma:
  - GENERATE (Exponential(1,0,6)),,300;
  - Exponential apare pentru că repartiția timpilor este exponențială. Nu există operanzii B și C, dar există operandul D = limita de generare. 6 din expresia exponențialei provine din faptul că al treilea parametru din expresia exponențialei reprezintă (1/frecvența cu care sosesc clienții). Frecvența este pentru prima jumătate de oră  $[600 \text{ clienți/oră}] = [600 \text{ clienți}/(60 \times 60 \text{ secunde})] = 1/6$ . Operandul 300 provine de la limita de generare. Dacă frecvența este de 600 clienți/oră și durata este de o jumătate de oră, rezultă că se vor genera 300 de clienți cu această frecvență.
  - În mod analog se generează clienți pentru următoarea oră. De data aceasta va exista și operandul C = întârzierea cu care se generează tranzacțiile. Pentru că acum se vor genera clienți cu o întârziere de jumătate de oră,  $C=1800$ . Găsiți al treilea parametru al exponențialei și limita de generare în acest caz.
  - Se procedează la fel pentru următoarea oră.
  - Se procedează la fel pentru ultimul interval de timp. De data aceasta nu va mai exista limita de generare.
- Toți clienții, indiferent de frecvența la care sosesc, vor parcurge sistemul în același fel. Pentru a descrie o singură dată ceea ce se întâmplă în sistem, după fiecare GENERATE se va face un transfer necondiționat către un bloc de la care începe descrierea activității din sistem.
- Mai întâi, pentru clienții care sosesc în parcare, se verifică dacă există un loc liber. Acest lucru se face cu ajutorul unui bloc TRANSFER care lucrează în modul Both:
  - TRANSFER Both,EtichetaBloc1,EtichetaBloc2
  - În acest fel, blocul TRANSFER verifică dacă tranzacția este acceptată fie de blocul etichetat cu EtichetaBloc1, fie de blocul cu eticheta EtichetaBloc2. Tranzacția va merge la primul bloc care o acceptă. În cazul nostru EtichetaBloc1 poate fi eticheta

blocului care simulează intrarea în parcare, iar EtichetaBloc2 poate fi eticheta unui bloc TERMINATE asociat clienților care sosesc și ies din sistem imediat, pentru că nu există loc de parcare.

- Se simulează drumul de 1 minut până la intrarea în magazin.
- Entitatea SAVEVALUE care numără clienții crește cu o unitate.
- Se definește cu ajutorul unui bloc ASSIGN un parametru al tranzacției active (al clientului) care reprezintă cantitatea cumpărată.
- Se testează cu ajutorul unui bloc TEST dacă clientul cumpără mai puțin de 10 produse, alegând astfel un coș, sau cumpără mai mult de 10 produse, mergând în acest caz la un bloc de unde va începe descrierea a ceea ce se întâmplă dacă alege un cărucior.
- Se simulează luarea unui coș prin ocuparea unui loc în entitatea SORAGE care simulează mulțimea de coșuri.
- Printr-un bloc ASSIGN este definit un parametru al tranzacției active, care reprezintă modul în care sunt duse cumpărăturile (în coș sau în cărucior).
- Tranzacția este transferată către un bloc de unde începe simularea cumpărării.
- Analog cu descrierea luării unui coș, se descrie luarea unui cărucior.
- Urmează descrierea cumpărării:
- Se simulează durata cumpărării
- Se testează dacă s-au cumpărat mai puțin de 10 produse.
  - Dacă da, se merge la următorul bloc de unde începe simularea plății la casa specială (ocuparea casei speciale, durata plății, eliberarea casei speciale, eliberarea coșului și transferarea la un bloc de unde va începe descrierea ieșirii din magazin).
  - Dacă nu, se merge la un bloc de unde începe descrierea cumpărării a mai mult de 10 produse.
- Dacă s-au cumpărat mai mult de 10 produse, trebuie aleasa casa în dreptul căreia se află coada cea mai mică. Acest lucru se face cu ajutorul unui bloc SELECT.
  - Forma generală a blocului SELECT este:
  - SELECT O A,B,C,D,E,F
  - Unde
    - O este un operator condițional sau logic. În cazul nostru va fi MIN (pentru alegerea minimului)
    - A este un număr sau un nume de parametru care va primi valoarea selectată. De exemplu în cazul nostru poate fi CodaMin.

- [B,C] reprezintă intervalul din care sunt luate entitățile care se testează. În cazul nostru B=2 și C=18 (numărul caselor obișnuite).
  - D este o valoare de referință pentru operandul E. În cazul nostru nu va exista.
  - E este tipul de entitate. În cazul nostru este Q (coadă)
  - F este eticheta unui bloc la care este trimisă tranzacția în cazul în care nu poate fi selectată nici o entitate. În cazul nostru nu va exista.
- Se simulează intrarea în coada cu numărul minim de clienți și apoi servirea, folosind atât pentru identificarea cozii, cât și pentru identificarea facilității, P\$CodaMin (parametrul cu numele CodaMin a fost definit la blocul SELECT și reprezintă numărul casei la care coada este minimă). După simularea servirii, este eliberat căruciorul și urmează ieșirea din magazin.
  - La ieșirea din magazin se adună informații pentru histograme, se scade numărul de clienți cu o unitate, se simulează durata deplasării până la parcare, ieșirea din parcare și ieșirea din sistem.
  - În a treia parte se vor genera tranzacții (pentru fiecare minut) care să simuleze trecerea timpului (se va da START 180 – nr. de minute din 3 ore).