

## Activitatea 2 Rework

1. Pentru clasa `AbstractFifo` explicati rolul fiecarui tip de date membru (ex. La ce este folosit, “`cQueue queue;`”).

`msgServiced`-parametrul folosit pentru a stoca temporar unele mesaje scoase din coada.

`endServiceMsg`-parametrul care semnaleaza terminarea unui mesaj.

`queue` – este o instant a clasei `cQueue` care are rolul de coada in programul nostru, cu ajutorul acestui atribut putem adauga/scoate mesaje in coada noastra

`qlenSignal` – semnalul care contine lungimea cozii

`busySignal` – semnalul care contine disponibilitatea cozii

`queueingTimeSignal` –timestamp-ul la care un mesaj iese din queue

2. Ce fac liniile de cod din metoda `AbstractFifo::initialize()`?

`qlenSignal = registerSignal("qlen");`

Stocheaza in variabila `qlenSignal` `signalID`-ul pentru semnalul cu numele `qlen`

`busySignal = registerSignal("busy");`

Stocheaza in variabila `busySignal` `signalID`-ul pentru semnalul cu numele `busy`

`queueingTimeSignal = registerSignal("queueingTime");`

Stocheaza in variabila `queueingTimeSignal` `signalID`-ul pentru semnalul cu numele `queueingTime`

3. Dar liniile de cod? La ce sunt folosite?

`emit(qlenSignal, queue.getLength());`

Emite obiectul dat ca parametru prin `queue.getLength()`; ca fiind un semnal

`emit(busySignal, false);`

Emite valoarea `false` ca un semnal.

4. Ce functionalitate implementeaza metoda `handleMessage`? Explicati succint.

`handleMessage` este metoda call-uita atunci cand modulul primeste un mesaj.

5. Explicati ce fac urmatoarele linii de cod ():

`if (!msgServiced) {`

`arrival(msg);`

## Activitatea 2 Rework

```
msgServiced = msg;  
emit(queueingTimeSignal, SIMTIME_ZERO);  
simtime_t serviceTime = startService(msgServiced);  
scheduleAt(simTime()+serviceTime, endServiceMsg);  
emit(busySignal, true); }
```

Daca msgServiced este null inseamna ca queue este gol si putem trimite mesajul, semnalăm ca mesajul a ajuns, păstrăm valoarea mesajului în msgServiced. Planificăm mesajul de endServiceMsg cu un delay adăugat de serviceTime și emitem semnalul true pentru a semnala faptul că queue-ul este busy.

6. Ce specializare/implementare oferă clasa Fifo?

Clasa Fifo moștenește clasa AbstractFifo și implementează cele două metode abstracte startService și endService.

Functia startService returnează referința către parametru serviceTime configurabil din fișierul ini

Functia endService trimite mesajul out și afișează că service-ul a fost complet.

## Activitatea 2 Rework