

ADT_FIFOQueue specificatie Communicatie buffer Datastructuren college 2

Elementen: zijn van bepaald datatype (b.v. karakters)

Structuur: mechanisme om volgorde van binnenkomst te bepalen

Domein: [0..max] elementen; 0=leeg,max=vol

Type: FIFOqueue

Operaties:

Procedure Enqueue(var Q:FIFOqueue; var kar:Element);

Preconditie: Vol(Q) is False

Postconditie: kar is toegevoegd aan Q als meest recente item

Procedure Dequeue(var Q:FIFOqueue;var kar:Element);

Preconditie: Voorraad>0

Functie Voorraad(Q:FIFOqueue):integer;

Postconditie: aantal elementen in FIFOqueue

Functie Vol(Q:FIFOqueue):boolean;

Postconditie: Vol=Y als max elementen in queue, anders Vol=N

Procedure Create(var Q:FIFOqueue;var created:boolean);

Postconditie: lege FIFOqueue is aangemaakt als created=Y, niet als created=N

Procedure Delete(var Q:FIFOqueue);

Postconditie: Q bestaat niet meer

N.B.: Voorraad in plaats van EMPTY als functie omdat een geheel gevulde en geheel lege buffer anders niet uit elkaar gehouden kunnen worden.

FIFOqueue Array implementatie

Const

```
Maxsize=nnn;
```

Type

```
Datatype = char;
```

```
FIFOqueue = ^FIFOqueueType;
```

```
FIFOqueueType = record
```

```
    Head : 1..Maxsize;
```

```
    Tail : 1..Maxsize;
```

```
    Size : 0..Maxsize;
```

```
    Queue :array[1..Maxsize] of Datatype
```

```
end;
```

```
Procedure Create(var Q:FIFOqueue;var created:Boolean);
```

```
begin
```

```
    Created:=Y;
```

```
    New(Q);
```

```
    With Q^ do begin
```

```
        Head:=1;
```

```
        Tail:=Maxsize;
```

```
        Size:=0
```

```
    end
```

```
end;
```

```
Procedure Delete(var Q:FIFOqueue);
```

```
begin
```

```
    Dispose(Q)
```

end;

Procedure Enqueue(var Q:FIFOqueue; elem:Datatype); #aan voorkant toevoegen is sluitstuk (tail)

begin

 with Q^ do begin

 Tail:=(Tail modulo Maxsize)+1; # NB! In C++ ((Tail+1) modulo Maxsize)

 Queue[Tail]:=elem;

 Size:=Size+1

 end

End;

Procedure Dequeue(var Q:FIFOqueue;elem:Datatype); # aan achterzijde afhalen is oudste (head)

begin

 with Q^ do begin

 elem:=Queue[Head];

 Head:=(Head modulo Maxsize)+1;

 Size:=Size-1

 end

end;

Function Length(Q:FIFOqueue):integer;

begin

 Length:=Q^.Size;

end;

Function Full(Q:FIFOqueue):Boolean;

begin

 Full:=(Q^.Size = Maxsize);

end;