# 3-Pile et File

- Définitions
- Primitives de piles, exemples
- Primitives de files, exemples
- Implémentation des piles
- Implémentation des files

# 3.1-Pile et File: Définition

Les piles et les files sont des containeurs dans lesquels on ne peut accéder qu'à **un** objet particulier.

**Définition 3.1**. :
Dans une pile, l'objet accessible est le dernier inséré (LIFO, Last-In, First-Out).

**Définition 3.2**:
Dans une file, l'objet accessible est le plus ancien dans la file (FIFO, First-In, First-Out).

On écrira pour déclarer des variables :

```
type_pile=pile de objet;
type_file=file de objet;
```

# 3.2-Primitives de pile

Une pile est définie par les opérations suivantes :

**accès**

```
fonction valeur(val P:pile de objet):objet;
fonction pileVide(val P:pile de objet):booléen;
```

**modification**

```
fonction creerPile(ref P:pile de objet):vide;
fonction empiler(ref P:pile de objet;
                 val:objet):vide;
fonction depiler (ref P:pile de objet):vide;
fonction detruirePile(ref P:pile de objet):vide;
```

# 3.2-Construire une listeSC inversée à partir d'une listeSC

```
fonction listeInverse(ref L:listeSC de objet):listeSC de objet;

var P:pile de objet;
var LR:liste de objet;

debut
 creerListe(LR);
 creerPile(P);
 debutListe(L);
   tant que !finListe(L) faire
     empiler(P,valeur(L));
     suivant(L);
   fintantque
   insererEnTete(LR,valeur(P))
   depiler(P);
   tant que non(pileVide(P)) faire
     insererApres(LR,valeur(P));
     suivant(LR);
     depiler(P);
   fintantque;
   detruirePile(P)
   retourner(LR);
fin
finfonction
```

## 3.3-Primitives de File

Une file est définie par les opérations suivantes :

**Accès :**

```
fonction valeur(val F:file de objet):objet;
fonction fileVide(val F:file de objet):booléen;
```

**Modification :**

```
fonction creerFile(ref F:file de objet):vide;
fonction enfiler(ref F:file de objet;
                 val v:objet):vide;
fonction defiler(ref F:file de objet):vide;
fonction detruireFile(ref F:file de objet):vide;
```

## 3.3-Compter le nombre d'élément d'une file d'entiers non nuls

```
fonction compteFile(ref F:file de entier): entier;
  var v,compt:entier;
  debut
    compt=0;
    enfiler(F,0);
    tant que valeur(F)!=0 faire
      compt=compt+1;
      v=valeur(F);
      defiler(F);
      enfiler(F,v);
    fintantque;
    defiler(F);
    retourner(compt);
  fin
finfonction
```

## 3.3- Inverser une file d'entiers non nuls

```
fonction inverserFile(ref F:file de entier):file d'entier;
  var P:pile d'entier;
  var FS:file d'entier;
  debut
    creerPile(P);
    creerFile(FS);
    enfiler(F,0);
    tant que valeur(F)!=0
    faire
      v=valeur(F);
      defiler(F);
      enfiler(F,v);
      empiler(P,v);
    fintantque
    defiler(F);
    tant que non(pileVide(P))
    faire
          v=valeur(P);
          enfiler(FS,v);
          depiler(P);
       fintantque;
       detruirePile(P);
       retourner(FS);
     fin
finfonction
```

## 3.4- Implémentation de pile dans un tableau

Chaque objet de la pile est un élément du tableau.
On doit de plus avoir un champs qui permet d'accéder au sommet de pile.

```
pile d'objet=structure
                taille:entier;
                sommet:entier;
                pile:tableau[1..taille] d'objets;
             finstructure;
```

## 3.4- Implémentation de pile dans un tableau

### accès

```
fonction valeur(ref P:pile de objet):objet;
  debut
    retourner(P.pile[P.sommet]);
  fin
finfonction

fonction pileVide(ref P:pile de objet):booléen;
  debut
    retourner(P.sommet==0);
  fin
finfonction
```

## 3.4- Implémentations de pile dans un tableau

### modification

```
fonction empiler(ref P:pile de objet; x:objet):booleen;
  /* l'espace de stockage peut être saturé */
  debut
    si P.sommet==P.taille alors
      retourner(FAUX)
    sinon
      P.sommet=P.sommet+1;
      P.pile[P.sommet]=x;
      retourner(VRAI)
    finsi
  fin
finfonction
```

## 3.4- Implémentations de pile dans un tableau

### modification

```
fonction depiler(ref P:pile de objet):vide;
  debut
    P.sommet=P.sommet-1;
  fin
finfonction

fonction creerPile(ref P:pile de objet):pile de
objet;
  debut
    P.sommet=0;
  fin
finfonction
```

## 3.5- Implémentations de pile avec une liste-SC

Chaque objet de la pile est un objet de la listeSC.

```
pile d'objet=listeSC de objet;
```

**accès :**

```
fonction valeurPile(ref P:pile de objet):objet;
  debut
    debutListe(P);
    retourner(valeurListe(P));
  fin
finfonction

fonction pileVide(ref P:pile de objet):booléen;
  debut
    retourner(listeVide(P));
  fin
finfonction
```

## 3.5- Implémentations de pile avec une liste-SC

**modification**

```
fonction empiler(ref P:pile de objet; x:objet):vide;
  debut
    insérerEnTete(P,x)
  fin
finfonction

fonction depiler(ref P:pile de objet):vide;
  debut
    supprimerEnTete(P);
  fin
finfonction
```

## 3.5- Implémentations de pile avec une liste-SC

**modification**

```
fonction creerPile(ref P:pile de objet):vide;
  debut
    creerListe(P);
  fin
finfonction
```

## 3.6- Implémentations de file dans un tableau

Chaque objet de la file est un élément du tableau. On utilise le tableau de manière circulaire avec un pointeur donnant le premier et un autre donnant le dernier.

```
file d'objet=structure
                taille : entier;
                premier : entier;
                dernier : entier;
                plein : booléen;
                file : tableau[0..taille-1] d'objets;
              finstructure;
```

## 3.6- Implémentations de file dans un tableau

**accès**

```
fonction valeur(ref F:file de objet):objet;
  debut
    retourner(F.file[F.premier]);
  fin
finfonction


fonction fileVide(ref F:file de objet):booléen;
  debut
    retourner(F.premier==F.dernier & non(F.plein));
  fin
finfonction
```

## 3.6- Implémentations de file dans un tableau

**Modification**

```
fonction enfiler(ref F:file de objet; x:objet):booleen;
  debut
    si F.plein alors
      retourner(FAUX)
    sinon
      F.file[F.dernier]=x;
      F.dernier=(F.dernier+1) mod F.taille;
      F.plein=F.dernier==F.premier;
      retourner(VRAI)
    finsi
  fin
finfonction
```

## 3.6- Implémentations de file dans un tableau

**Modification**

```
fonction creerFile(ref F:file de objet):file de
objet;
  debut
    F.premier= 0;
    F.dernier= 0;
    F.plein=FAUX;
  fin
finfonction
```

## 3.6- Implémentations de file dans un tableau

**Modification**

```
fonction defiler(ref F:file de objet):vide;
  debut
    F.premier=(F.premier+1) mod F.taille;
    F.plein=Faux
  fin
finfonction
```

## 3.7- Implémentations de file par une liste_DC

Chaque objet de la file est un objet de la liste_DC car il faut un accès au dernier.

```
file d'objet=liste_DC de objet;
```

**Accès :**

```
fonction fileVide(ref F:file de objet):booléen;
 debut
   retourner(listeVide(F));
 fin
 finfonction
```

## 3.7- Implémentations de file par une liste_DC

Chaque objet de la file est un objet de la liste_DC car il faut un accès au dernier.

```
file d'objet=liste_DC de objet;
```

**Accès :**

```
fonction valeurFile(ref F:file de objet):objet;
   debut
     debutListe(F);
     retourner(valeurListe(F));
   fin
 finfonction
```

## 3.7- Implémentations de file par une liste_DC

**modification**

```
fonction enfiler(ref F:file de objet; x:objet):vide;
  debut
    dernier(F);
    insererApres(F,x);
  fin
finfonction

fonction defiler(ref F:file de objet):vide;
  debut
    supprimerEnTete(F);
  fin
finfonction
```

## 3.7- Implémentations de file par une liste_DC

**modification**

```
fonction creerFile(ref F:file de objet):vide;
  debut
    creerListe(F);
  fin
finfonction
```

## 3.8- Implémentation par une liste_SC avec un pointeur sur le dernier élément

```
cellule=structure
                valeurElement:objet;
                pointeurSuivant:^cellule;
            finstructure;

curseur=^cellule;

file d'objet=structure
             premier:curseur;
             dernier:curseur
          finstructure
```

## 3.8- Implémentation par une liste_SC avec un pointeur sur le dernier élément

**accès**

```
fonction valeurFile(ref F:file de objet):objet;
  debut
   retourner(F.premier^.valeurElement);
  fin
finfonction


fonction fileVide(ref F:file de objet):booléen;
  debut
    retourner(F.premier==NIL);
  fin
finfonction
```

## 3.8- Implémentation par une liste_SC avec un pointeur sur le dernier élément

**modification**

```
fonction enfiler(ref F:file de objet; x:objet):vide;
  var c:curseur;
  debut
    new(c);
    c^.valeurElement=x;
    c^.pointeurSuivant=NIL;
    si F.premier==NIL alors
      F.premier=C
    finsi
    F.dernier^.pointeurSuivant=c;
    F.dernier=c;
  fin
finfonction
```

## 3.8- Implémentation par une liste_SC avec un pointeur sur le dernier élément

**modification**

```
fonction defiler(ref F:file de objet):vide;
  var c:curseur;
  debut
    c=F.premier;
    si F.premier=F.dernier alors
      F.dernier=NIL
    finsi
    F.premier=c^.pointeurSuivant;
    delete(c)
  fin
finfonction
```

## 3.8- Implémentation par une liste_SC avec un pointeur sur le dernier élément

**modification**

```
fonction creerFile(ref F:file de objet):vide;
  debut
    F.premier=NIL;
    F.dernier=NIL;
  fin
finfonction
```

## 3.8- Implémentation par une liste_SC avec un pointeur sur le dernier élément

**modification**

```
fonction detruireFile(ref F:file de objet):vide;
  debut
    tantque !fileVide(F) faire
      defiler(F)
    fintantque
  fin
finfonction
```