

Algorithmique procédurale

Les Listes, Piles et Files

Équipe pédagogique

CY Tech

Bibliographie - Sitographie

- Transparents Houcine Senoussi
- Transparents Jean-Paul Forest
- Wikipedia : <https://fr.wikipedia.org/wiki/Algorithme>
- Laurence Pilard (Université de Versailles)
- Introduction à l'algorithmique - Cormen, Leiserson, Rivest, Stein
- Edition DUNOD

Plan

- Introduction
- Structures de données élémentaires
- Les Listes
- Les Piles
- Les Files
- Déclaration algorithmique

Introduction

- En algorithmique, il est souvent nécessaire :

Introduction

- En algorithmique, il est souvent nécessaire :
 - ▶ regrouper des données

Introduction

- En algorithmique, il est souvent nécessaire :
 - ▶ regrouper des données
 - ▶ de les organiser pour s'adapter aux traitements prévus par les algorithmes.

Introduction

- En algorithmique, il est souvent nécessaire :
 - ▶ regrouper des données
 - ▶ de les organiser pour s'adapter aux traitements prévus par les algorithmes.
- Une structure de données se définit par la manière dont elle organise les éléments qu'elle contient et par les opérations qu'elle autorise sur cet ensemble d'éléments.

Introduction

- Par exemple : un tableau

Introduction

- Par exemple : un tableau
 - ▶ Chaque élément est identifié par un indice unique.

Introduction

- Par exemple : un tableau
 - ▶ Chaque élément est identifié par un indice unique.
 - ▶ Les indices varient de 1 à la longueur du tableau.

Introduction

- Par exemple : un tableau
 - ▶ Chaque élément est identifié par un indice unique.
 - ▶ Les indices varient de 1 à la longueur du tableau.
 - ▶ Ces indices permettent d'accéder en lecture et en écriture à tout élément, à tout moment.

Introduction

- Il y a donc une notion d'ensemble de données :

Introduction

- Il y a donc une notion d'ensemble de données :
 - ▶ Notion d'ensemble indispensable pour l'informatique comme pour les mathématiques.

Introduction

- Il y a donc une notion d'ensemble de données :
 - ▶ Notion d'ensemble indispensable pour l'informatique comme pour les mathématiques.
 - ▶ Différence : ensemble mathématique stable versus ensemble informatique pouvant croître, diminuer, se modifier au cours du temps.

Introduction

- Il y a donc une notion d'ensemble de données :
 - ▶ Notion d'ensemble indispensable pour l'informatique comme pour les mathématiques.
 - ▶ Différence : ensemble mathématique stable versus ensemble informatique pouvant croître, diminuer, se modifier au cours du temps.
 - ▶ C'est des ensembles dynamiques.

Introduction

- Le terme de structure de données désigne donc une composition de données unies par une même sémantique.

Introduction

- Le terme de structure de données désigne donc une composition de données unies par une même sémantique.
- On peut aussi la définir comme une donnée abstraite dont le comportement est modélisé par des opérations abstraites.

Structures de données élémentaires

- Les structures linéaires sont un des modèles de données les plus élémentaires et utilisés en algorithmique.

Structures de données élémentaires

- Les structures linéaires sont un des modèles de données les plus élémentaires et utilisés en algorithmique.
- C'est des données sous forme de séquence non ordonnée d'éléments accessibles de façon séquentielle.

Structures de données élémentaires

- Les structures linéaires sont un des modèles de données les plus élémentaires et utilisés en algorithmique.
- C'est des données sous forme de séquence non ordonnée d'éléments accessibles de façon séquentielle.
- Tout élément d'une séquence sauf le dernier possède un successeur.

Structures de données élémentaires

- Les opérations d'ajout et de suppression sont les opérations de base des structures linéaires.

Structures de données élémentaires

- Les opérations d'ajout et de suppression sont les opérations de base des structures linéaires.
- Selon la façon dont fonctionnent ces opérations, on distingue plusieurs structures linéaires.

Structures de données élémentaires

- Les opérations d'ajout et de suppression sont les opérations de base des structures linéaires.
- Selon la façon dont fonctionnent ces opérations, on distingue plusieurs structures linéaires.
- Les listes : ajout et suppression n'importe où dans la séquence.

Structures de données élémentaires

- Les opérations d'ajout et de suppression sont les opérations de base des structures linéaires.
- Selon la façon dont fonctionnent ces opérations, on distingue plusieurs structures linéaires.
- Les listes : ajout et suppression n'importe où dans la séquence.
- Les piles et les files : ajout et suppression qu'aux extrémités (cas particulier de la liste).

Les listes

- Une liste est la forme générale d'une séquence finie d'éléments repérés selon leur rang.

Les listes

- Une liste est la forme générale d'une séquence finie d'éléments repérés selon leur rang.
- Une liste est définie par sa composition :

Les listes

- Une liste est la forme générale d'une séquence finie d'éléments repérés selon leur rang.
- Une liste est définie par sa composition :
 - ▶ soit elle est vide,

Les listes

- Une liste est la forme générale d'une séquence finie d'éléments repérés selon leur rang.
- Une liste est définie par sa composition :
 - ▶ soit elle est vide,
 - ▶ soit elle est composée d'un élément (appelé tête) suivi d'une liste (appelée reste).

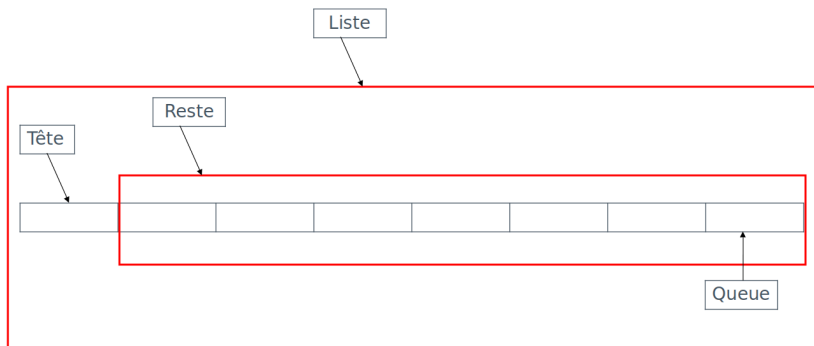
Les listes

- Une liste est la forme générale d'une séquence finie d'éléments repérés selon leur rang.
- Une liste est définie par sa composition :
 - ▶ soit elle est vide,
 - ▶ soit elle est composée d'un élément (appelé tête) suivi d'une liste (appelée reste).
- Notons que cette définition est récursive (voir cours suivant).

Les listes

- Une liste est la forme générale d'une séquence finie d'éléments repérés selon leur rang.
- Une liste est définie par sa composition :
 - ▶ soit elle est vide,
 - ▶ soit elle est composée d'un élément (appelé tête) suivi d'une liste (appelée reste).
- Notons que cette définition est récursive (voir cours suivant).
- Il existe des listes dont le premier élément (tête de liste) n'a pas de prédécesseur et le dernier élément n'a pas de successeur – on l'appelle queue de liste.

Les listes



Opérations de base sur les listes



- Créer une liste vide.

Opérations de base sur les listes



- Créer une liste vide.
- Tester si la liste est vide.

Opérations de base sur les listes



- Créer une liste vide.
- Tester si la liste est vide.
- Récupérer la tête de la liste.

Opérations de base sur les listes



- Créer une liste vide.
- Tester si la liste est vide.
- Récupérer la tête de la liste.
- Récupérer le reste de la liste.

Opérations de base sur les listes



- Créer une liste vide.
- Tester si la liste est vide.
- Récupérer la tête de la liste.
- Récupérer le reste de la liste.
- Ajouter un élément (qui deviendra la tête).

Opérations de base sur les listes



- Créer une liste vide.
- Tester si la liste est vide.
- Récupérer la tête de la liste.
- Récupérer le reste de la liste.
- Ajouter un élément (qui deviendra la tête).
- Supprimer la tête.

Opérations de base sur les listes



- Créer une liste vide.
- Tester si la liste est vide.
- Récupérer la tête de la liste.
- Récupérer le reste de la liste.
- Ajouter un élément (qui deviendra la tête).
- Supprimer la tête.
- Mais aussi : détruire, vider, récupérer la queue ou un élément quelconque, ajouter un élément en queue ou insérer l'élément dans la liste, supprimer un élément.

Les listes

Liste de chaîne



Les listes

Ajouter un élément



Les listes

Ajouter un élément



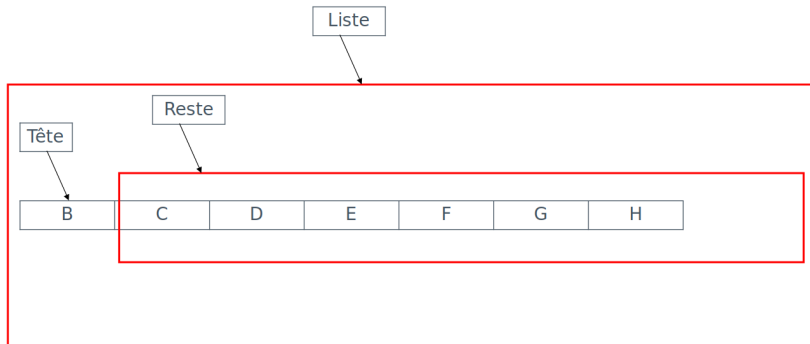
Les listes

Supprimer un élément



Les listes

Supprimer un élément



Les piles

- Une pile fonctionne selon le principe du dernier entré premier sorti : LIFO (Last-In, First-Out).

Les piles

- Une pile fonctionne selon le principe du dernier entré premier sorti : LIFO (Last-In, First-Out).
- Dans cette structure, le seul élément accessible est le dernier entré qui se nomme sommet.

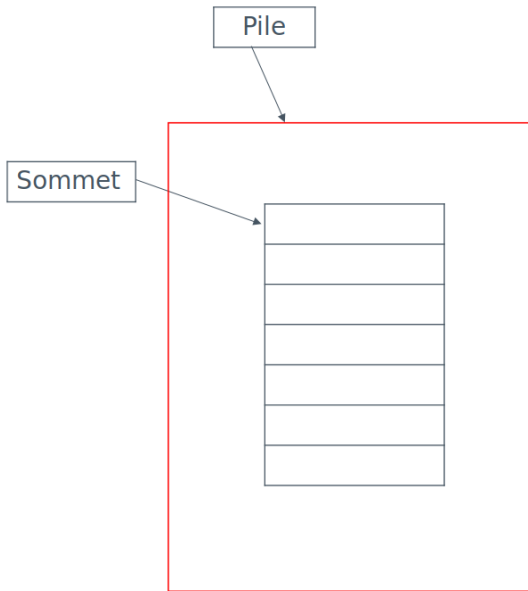
Les piles

- Une pile fonctionne selon le principe du dernier entré premier sorti : LIFO (Last-In, First-Out).
- Dans cette structure, le seul élément accessible est le dernier entré qui se nomme sommet.
- Exemple : une pile d'assiettes, une pile de pièces

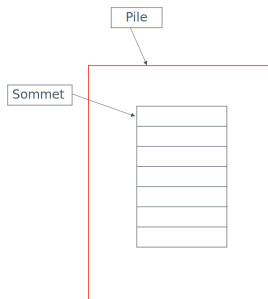
Les piles

- Une pile fonctionne selon le principe du dernier entré premier sorti : LIFO (Last-In, First-Out).
- Dans cette structure, le seul élément accessible est le dernier entré qui se nomme sommet.
- Exemple : une pile d'assiettes, une pile de pièces
- En informatique, les piles sont très utilisées comme par exemple les logiciels qui permettent une fonction annuler (undo), le bouton retour du navigateur internet, la pile d'exécution dans un système d'exploitation.

Les piles

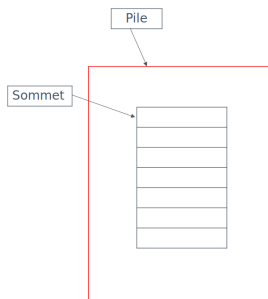


Opérations de base sur les piles



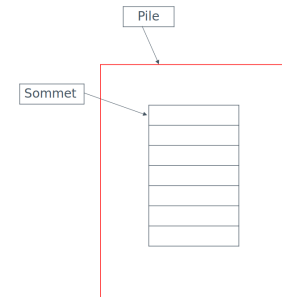
- Créer une pile vide.

Opérations de base sur les piles



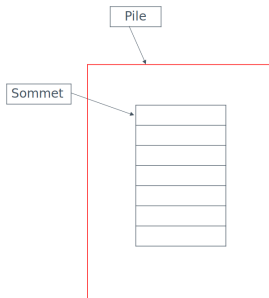
- Créer une pile vide.
- Tester si la pile est vide.

Opérations de base sur les piles



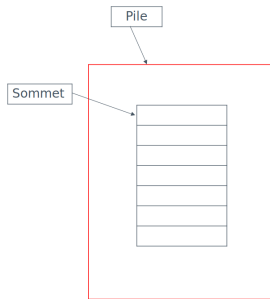
- Créer une pile vide.
- Tester si la pile est vide.
- Récupérer la valeur du seul élément accessible (sommet).

Opérations de base sur les piles



- Créer une pile vide.
- Tester si la pile est vide.
- Récupérer la valeur du seul élément accessible (sommet).
- Ajouter un élément : cette opération s'appelle Empiler.

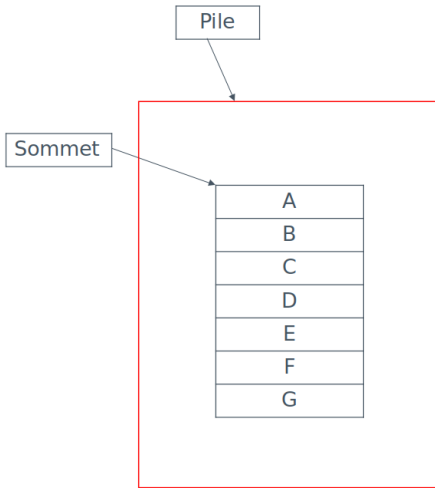
Opérations de base sur les piles



- Créer une pile vide.
- Tester si la pile est vide.
- Récupérer la valeur du seul élément accessible (sommet).
- Ajouter un élément : cette opération s'appelle Empiler.
- Supprimer l'élément : cette opération s'appelle Dépiler.

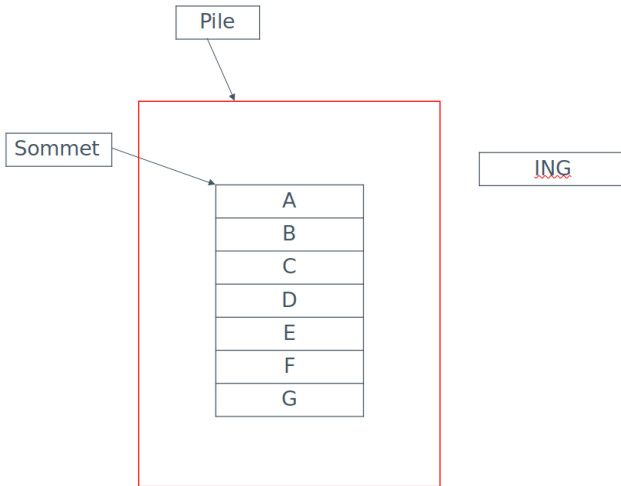
Les piles

Pile de chaîne



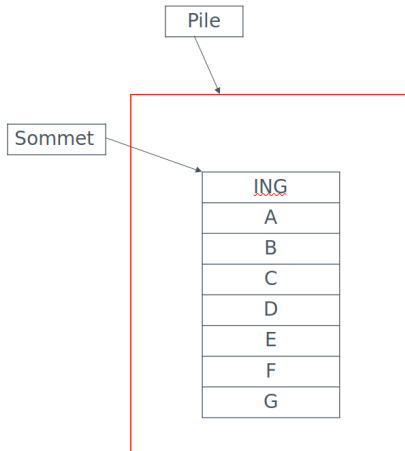
Les piles

Empiler



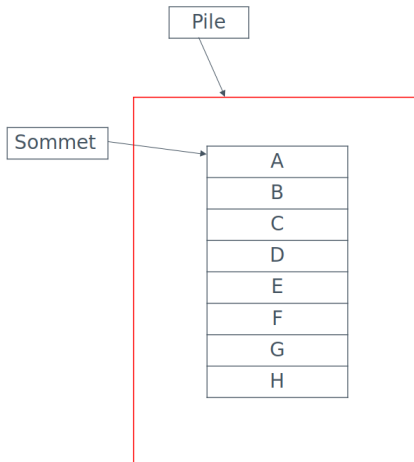
Les piles

Empiler



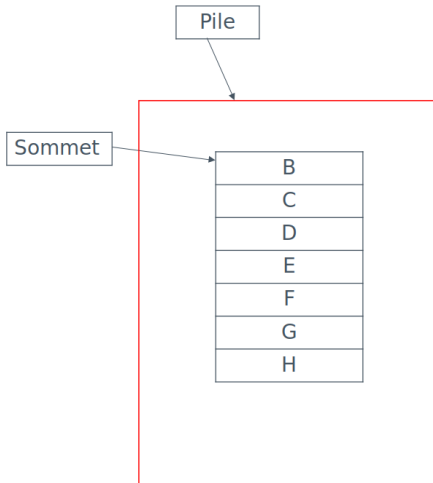
Les piles

Dépiler



Les piles

Dépiler



Les files

- Une file fonctionne selon le principe du premier entré premier sorti : FIFO (First-In, First-Out).

Les files

- Une file fonctionne selon le principe du premier entré premier sorti : FIFO (First-In, First-Out).
- Dans cette structure, les éléments sont insérés dans la séquence par une des extrémités et sont extraits de l'autre.

Les files

- Une file fonctionne selon le principe du premier entré premier sorti : FIFO (First-In, First-Out).
- Dans cette structure, les éléments sont insérés dans la séquence par une des extrémités et sont extraits de l'autre.
- Un seul élément est accessible, c'est le premier entré appelé premier.

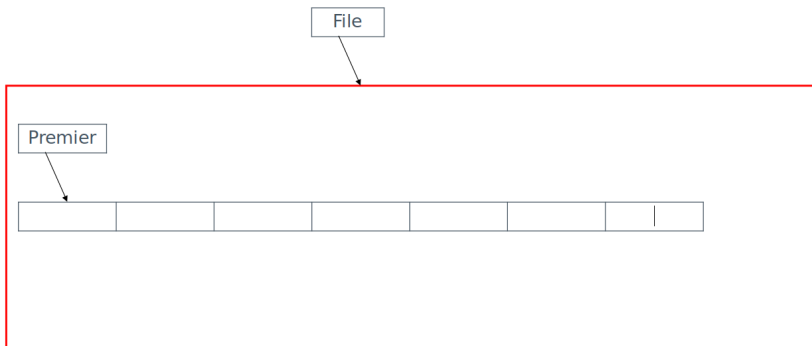
Les files

- Une file fonctionne selon le principe du premier entré premier sorti : FIFO (First-In, First-Out).
- Dans cette structure, les éléments sont insérés dans la séquence par une des extrémités et sont extraits de l'autre.
- Un seul élément est accessible, c'est le premier entré appelé premier.
- Exemple : une file d'attente à un guichet

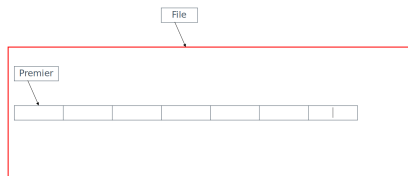
Les files

- Une file fonctionne selon le principe du premier entré premier sorti : FIFO (First-In, First-Out).
- Dans cette structure, les éléments sont insérés dans la séquence par une des extrémités et sont extraits de l'autre.
- Un seul élément est accessible, c'est le premier entré appelé premier.
- Exemple : une file d'attente à un guichet
- En informatique, la file d'impression, la file d'exécution des processus dans un système d'exploitation.

Les files

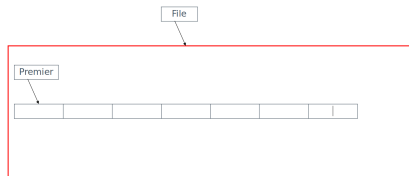


Opérations de base sur les files



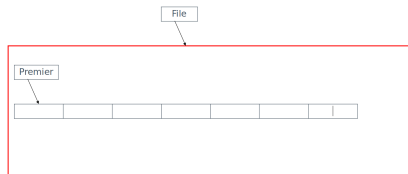
- Créer une file vide.

Opérations de base sur les files



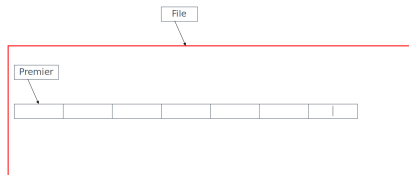
- Créer une file vide.
- Tester si la file est vide.

Opérations de base sur les files



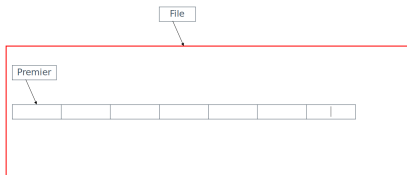
- Créer une file vide.
- Tester si la file est vide.
- Récupérer la valeur du seul élément accessible (premier).

Opérations de base sur les files



- Créer une file vide.
- Tester si la file est vide.
- Récupérer la valeur du seul élément accessible (premier).
- Ajouter un élément : cette opération s'appelle Enfiler.

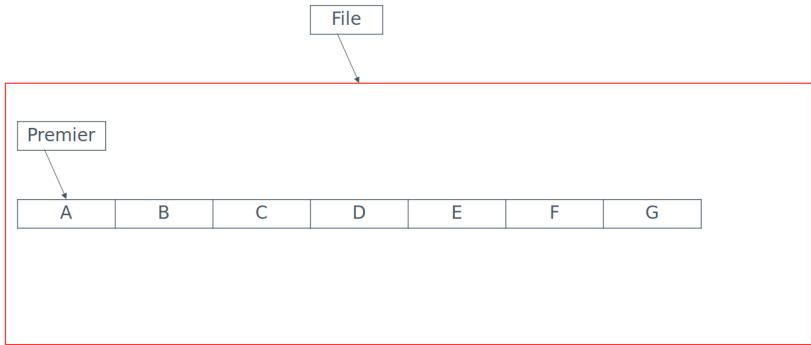
Opérations de base sur les files



- Créer une file vide.
- Tester si la file est vide.
- Récupérer la valeur du seul élément accessible (premier).
- Ajouter un élément : cette opération s'appelle Enfiler.
- Supprimer l'élément : cette opération s'appelle Défiler.

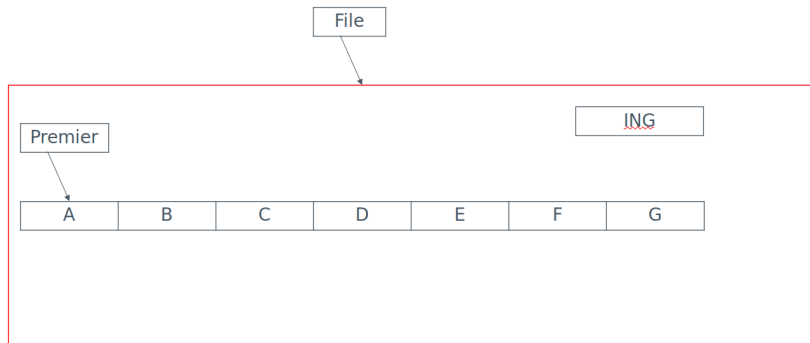
Les files

File de chaîne



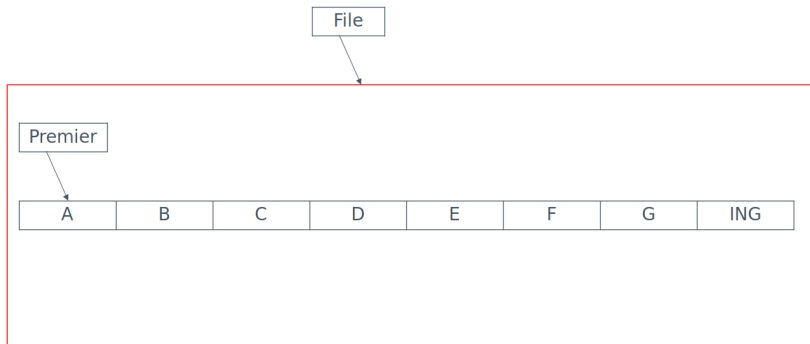
Les files

Enfiler



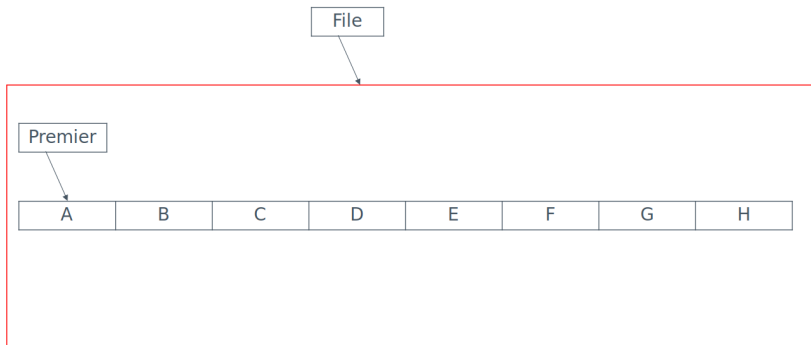
Les files

Enfiler



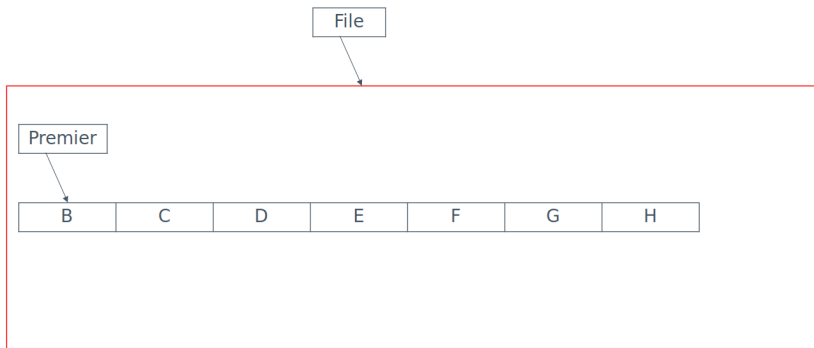
Les files

Défiler



Les files

Défiler



Déclaration algorithmique

- On a trois types de données Pile, File et Liste dont les éléments seront déclarés de la manière suivante :

Déclaration algorithmique

- On a trois types de données Pile, File et Liste dont les éléments seront déclarés de la manière suivante :
 - ▶ nomvariable : Liste de typededonnées

Déclaration algorithmique

- On a trois types de données Pile, File et Liste dont les éléments seront déclarés de la manière suivante :
 - ▶ nomvariable : Liste de typededonnées
 - ▶ nomvariable : Pile de typededonnées

Déclaration algorithmique

- On a trois types de données Pile, File et Liste dont les éléments seront déclarés de la manière suivante :
 - ▶ nomvariable : Liste de typededonnées
 - ▶ nomvariable : Pile de typededonnées
 - ▶ nomvariable : File de typededonnées

Déclaration algorithmique

- On a trois types de données Pile, File et Liste dont les éléments seront déclarés de la manière suivante :
 - ▶ nomvariable : Liste de typededonnées
 - ▶ nomvariable : Pile de typededonnées
 - ▶ nomvariable : File de typededonnées
- Exemple : Piecemonnaie : Pile de réel (2;1;0,5;0,2;...)

Opérations de base sur les listes

- Créer une liste vide \rightarrow Procédure créerListe(l : Liste (S)).

Opérations de base sur les listes

- Créer une liste vide \rightarrow Procédure créerListe(l : Liste (S)).
- Tester si la liste est vide \rightarrow Fonction estVide(l : Liste) : Booléen.

Opérations de base sur les listes

- Créer une liste vide \rightarrow Procédure créerListe(l : Liste (S)).
- Tester si la liste est vide \rightarrow Fonction estVide(l : Liste) : Booléen.
- Récupérer la tête de la liste \rightarrow Fonction tête(l : Liste) : Type des éléments de la liste.

Opérations de base sur les listes

- Créer une liste vide \rightarrow Procédure $\text{créerListe}(l : \text{Liste}(S))$.
- Tester si la liste est vide \rightarrow Fonction $\text{estVide}(l : \text{Liste}) : \text{Booléen}$.
- Récupérer la tête de la liste \rightarrow Fonction $\text{tête}(l : \text{Liste}) : \text{Type des éléments de la liste}$.
- Récupérer le reste de la liste \rightarrow Fonction $\text{reste}(l : \text{Liste}) : \text{Liste}$.

Opérations de base sur les listes

- Créer une liste vide \rightarrow Procédure créerListe(l : Liste (S)).
- Tester si la liste est vide \rightarrow Fonction estVide(l : Liste) : Booléen.
- Récupérer la tête de la liste \rightarrow Fonction tête(l : Liste) : Type des éléments de la liste.
- Récupérer le reste de la liste \rightarrow Fonction reste(l : Liste) : Liste.
- Ajouter un élément (qui deviendra la tête) \rightarrow Procédure ajouter(l : Liste (E/S), e : Type des éléments de la file).

Opérations de base sur les listes

- Créer une liste vide \rightarrow Procédure `créerListe(l : Liste (S))`.
- Tester si la liste est vide \rightarrow Fonction `estVide(l : Liste) : Booléen`.
- Récupérer la tête de la liste \rightarrow Fonction `tête(l : Liste) : Type des éléments de la liste`.
- Récupérer le reste de la liste \rightarrow Fonction `reste(l : Liste) : Liste`.
- Ajouter un élément (qui deviendra la tête) \rightarrow Procédure `ajouter(l : Liste (E/S), e : Type des éléments de la file)`.
- Supprimer la tête \rightarrow Procédure `supprimer(l : Liste (E/S))`.

Remarques sur les listes

- Suivant le type de d'implémentation informatique d'une liste, il est possible de créer plusieurs opérations complémentaires comme par exemple :

Remarques sur les listes

- Suivant le type de d'implémentation informatique d'une liste, il est possible de créer plusieurs opérations complémentaires comme par exemple :
 - ▶ Compter le nombre d'éléments,

Remarques sur les listes

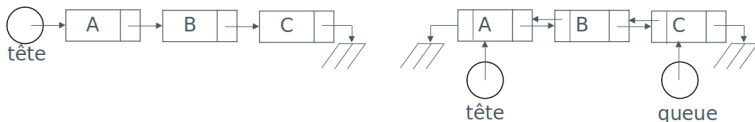
- Suivant le type de d'implémentation informatique d'une liste, il est possible de créer plusieurs opérations complémentaires comme par exemple :
 - ▶ Compter le nombre d'éléments,
 - ▶ Récupérer le ième éléments ...

Remarques sur les listes

- Suivant le type de d'implémentation informatique d'une liste, il est possible de créer plusieurs opérations complémentaires comme par exemple :
 - ▶ Compter le nombre d'éléments,
 - ▶ Récupérer le ième éléments ...
- Il est aussi possible de créer des listes chaînées (simple ou double)
 - on utilise la représentation suivante :

Remarques sur les listes

- Suivant le type de d'implémentation informatique d'une liste, il est possible de créer plusieurs opérations complémentaires comme par exemple :
 - ▶ Compter le nombre d'éléments,
 - ▶ Récupérer le ième éléments ...
- Il est aussi possible de créer des listes chaînées (simple ou double)
 - on utilise la représentation suivante :



Opérations de base sur les piles

- Créer une pile vide \rightarrow Procédure créerPile(p : Pile (S)).

Opérations de base sur les piles

- Créer une pile vide \rightarrow Procédure créerPile(p : Pile (S)).
- Tester si la pile est vide \rightarrow Fonction estVide(p : Pile) : Booléen.

Opérations de base sur les piles

- Créer une pile vide \rightarrow Procédure $\text{créerPile}(p : \text{Pile } (S))$.
- Tester si la pile est vide \rightarrow Fonction $\text{estVide}(p : \text{Pile}) : \text{Booléen}$.
- Récupérer la valeur du seul élément accessible (sommet) \rightarrow Fonction $\text{sommet}(p : \text{Pile}) : \text{Type des éléments de la pile}$.

Opérations de base sur les piles

- Créer une pile vide \rightarrow Procédure $\text{créerPile}(p : \text{Pile } (S))$.
- Tester si la pile est vide \rightarrow Fonction $\text{estVide}(p : \text{Pile}) : \text{Booléen}$.
- Récupérer la valeur du seul élément accessible (sommet) \rightarrow Fonction $\text{sommet}(p : \text{Pile}) : \text{Type des éléments de la pile}$.
- Ajouter un élément : cette opération s'appelle Empiler \rightarrow Procédure $\text{empiler}(p : \text{Pile } (E/S), e : \text{Type des éléments de la pile})$.

Opérations de base sur les piles

- Créer une pile vide \rightarrow Procédure créerPile(p : Pile (S)).
- Tester si la pile est vide \rightarrow Fonction estVide(p : Pile) : Booléen.
- Récupérer la valeur du seul élément accessible (sommet) \rightarrow Fonction sommet(p : Pile) : Type des éléments de la pile.
- Ajouter un élément : cette opération s'appelle Empiler \rightarrow Procédure empiler(p : Pile (E/S), e : Type des éléments de la pile).
- Supprimer l'élément : cette opération s'appelle Dépiler \rightarrow Procédure dépiler(p : Pile (E/S)).

Opérations de base sur les files

- Créer une file vide \rightarrow Procédure créerFile($f : \text{File}(S)$).

Opérations de base sur les files

- Créer une file vide \rightarrow Procédure `créerFile(f : File (S))`.
- Tester si la file est vide \rightarrow Fonction `estVide(f : File) : Booléen`.

Opérations de base sur les files

- Créer une file vide \rightarrow Procédure $\text{créerFile}(f : \text{File}(S))$.
- Tester si la file est vide \rightarrow Fonction $\text{estVide}(f : \text{File}) : \text{Booléen}$.
- Récupérer la valeur du seul élément accessible (premier) \rightarrow Fonction $\text{premier}(f : \text{File}) : \text{Type des éléments de la file}$.

Opérations de base sur les files

- Créer une file vide \rightarrow Procédure $\text{créerFile}(f : \text{File } (S))$.
- Tester si la file est vide \rightarrow Fonction $\text{estVide}(f : \text{File}) : \text{Booléen}$.
- Récupérer la valeur du seul élément accessible (premier) \rightarrow Fonction $\text{premier}(f : \text{File}) : \text{Type des éléments de la file}$.
- Ajouter un élément : cette opération s'appelle Enfiler \rightarrow Procédure $\text{enfiler}(f : \text{File } (E/S), e : \text{Type des éléments de la file})$.

Opérations de base sur les files

- Créer une file vide \rightarrow Procédure $\text{créerFile}(f : \text{File}(S))$.
- Tester si la file est vide \rightarrow Fonction $\text{estVide}(f : \text{File}) : \text{Booléen}$.
- Récupérer la valeur du seul élément accessible (premier) \rightarrow Fonction $\text{premier}(f : \text{File}) : \text{Type des éléments de la file}$.
- Ajouter un élément : cette opération s'appelle Enfiler \rightarrow Procédure $\text{enfiler}(f : \text{File}(E/S), e : \text{Type des éléments de la file})$.
- Supprimer l'élément : cette opération s'appelle Défiler \rightarrow Procédure $\text{défiler}(f : \text{File}(E/S))$.

