

# Structures

## I2 - Algorithmique et Programmation structurée

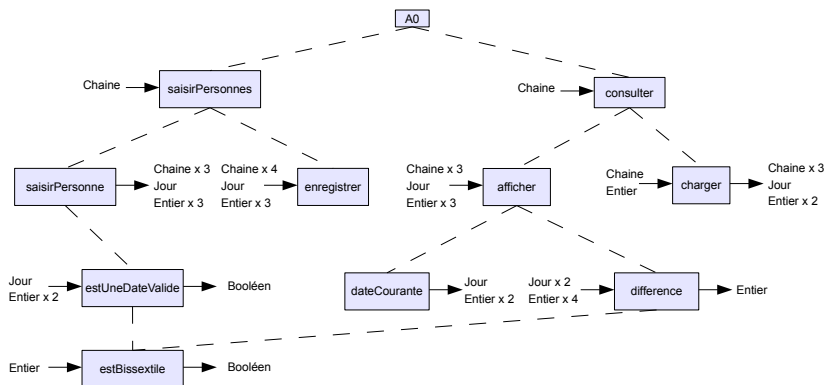
Julien SAUNIER, Alexandre Pauchet, Pierrick Tranouez

# Plan

- 1 Structures
  - Principe
  - Utilisations
- 2 Fichiers de structures
  - Rappels et pseudo-code
  - Fichier de structures
- 3 Exercices

# Structures

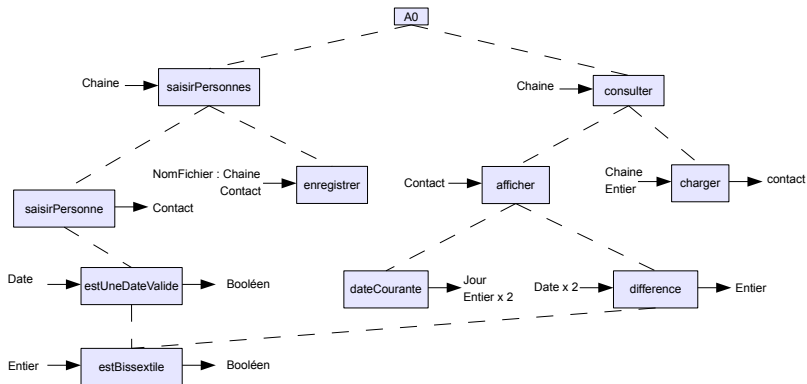
# Rappel : Analyse descendante carnet d'adresses



# Problèmes

- Comment simplifier l'analyse descendante ?
- En regroupant les éléments liés à un même concept en un seul type.

# Rappel : Analyse descendante carnet d'adresses



# Structure/enregistrement 1 / 3

## Principe

Une structure (également appelée enregistrement) est un type de données composite regroupant un ensemble fini de types de données.

L'intérêt est de stocker dans une seule variable plusieurs valeurs relatives à un même objet.

On pourrait représenter dans une même variable d'un type structure une personne caractérisée par :

- Son nom (chaîne de caractères)
- Son prénom (chaîne de caractères)
- Son âge (naturel)

# Structure/enregistrement 2 / 3

## Écriture en Pseudo-code

On définit un type de données spécifique (par le mot-clé **Type**) désignant une structure avec la syntaxe :

**Type** <nomType> = **Structure**

<champ1> : <type1>

<champ2> : <type2>

...

**finstructure**

Les sous-éléments d'une structure sont appelés des champs.

## Structure Personne

**Type** Personne = **Structure**

nom : **Chaîne de caracteres**

prenom : **Chaîne de caracteres**

age : **Entier**

**finstructure**



# Structure/enregistrement 3 / 3

## Écriture en Pascal

On définit habituellement un type de données spécifique (par le mot-clé `Type`) désignant une structure avec le mot-clé `record`). Syntaxe :

```
Type <nomType> = record
    <champ1> : <type1>;
    <champ2> : <type2>;
    ...
end;
```

## Structure Personne

```
Type Personne = record
    nom : string;
    prenom : string;
    age : integer;
end;
```

# Exercice

Représenter une date et un contact sous forme d'enregistrements.

# Accès aux champs d'une structure 1 / 2

## symbole d'accès

Pour accéder à un des champs d'une structure on utilise le symbole `.` après le nom de la variable structure et avant le nom du champ. Si `p` est une variable de type `Personne`, `p.nom` désigne un de ses champs.

## en pseudo-code

```

procédure saisirAge (S p : Personne)
debut
    lire(p.age)
fin
  
```

## en Pascal

```

procedure saisirAge(var p : Personne);
begin
    readln(p.age);
end;
  
```

# Accès aux champs d'une structure 2 / 2

## Parcours

Comme pour les tableaux, l'ensemble des valeurs d'une structure ne peuvent pas être parcourues d'une seule instruction (par exemple pour un affichage).

## en pseudo-code

**procédure** afficher (**E** p : Personne)

**debut**

**ecrire**(p.prenom)

**ecrire**(' ', p.nom, ' a ')

**ecrire**(p.age, ' ans')

**fin**

## Exercice 1 / 2

- quelle structure pour représenter une ligne du tableau ?
- quel type de données pour représenter tout le tableau (sauf les en-têtes des colonnes) ?

<i>Num</i>	<i>Nom</i>	<i>I1</i>	<i>I2</i>	<i>I3</i>	<i>Moyenne</i>
1	Alice	12,5	15	14,5	14
2	Bob	10	14,5	18,5	14,33
3	Charles	15	12	15	14
4	Dave	9	17	19	15
5	Eve	13	13	13	13
6	Franck	10	6	8	8,33

# Exercice 2 / 2

**Type Eleve = Structure**

id : **Entier**

nom : **Chaine de caracteres**

notel1 : **Reel**

notel2 : **Reel**

notel3 : **Reel**

**finstructure**

fonction **moyenne**(etud : Eleve) : **Reel**

Et pour la liste ?

# Tableau de structure

Une structure est un type de donnée comme un autre :

- Un seul tableau contenant des valeurs de type différents (mais stockées dans un même type de structure)
- Solution alternative à plusieurs tableaux ayant une correspondance d'indice

**Type** ListeEleve = **Structure**

liste : **Tableau**[1..MAX] d'Eleve

taille : **Entier**

**finstructure**

**Type** Identifiants = **Tableau**[1..MAX] de **Entier**

**Type** Noms = **Tableau**[1..26] de **Chaine de caracteres**

**Type** Notes = **Tableau**[1..26] de **Reel**

**Déclarations**

id :Identifiants ; names : Noms, notes1, notes2, notes3 : Notes ; taille : Entier ;

# Encapsulation

## Définition (Wikipedia)

L'encapsulation en général est la notion de mettre une chose dans une autre.

## Définition en informatique

En informatique l'encapsulation consiste à regrouper dans un même objet plusieurs informations qui lui sont attachées.

En programmation structurée, l'encapsulation consiste à définir une structure regroupant toutes les variables propres à un objet, de manière à le représenter par une unique variable.

## Exemple des tableaux à 1 dimension

```
Type Tableau1D = Structure  
    valeurs : Tableau[1..MAX] de Reel  
    nbElements : Entier  
finstructure
```



# Algorithmes sur une structure Tableau1D 1 / 2

## Accéder à une valeur $M_i$

**Constante** MAX = 1000

**Type** Tableau1D = **Structure**

nbElements : **Entier**

valeur : **Tableau**[1..MAX] de **Reel**

**finstructure**

**fonction** aPourValeur (**E** mat : Tableau1D, ind : **Entier**) : **Reel**

**debut**

**retourner** mat.valeur[ind]

**fin**

# Algorithmes sur une structure Tableau1D 2 / 2

## Modifier une valeur $M_i$

**procédure** modifierValeur (**E/S** mat : Tableau1D, **E** ind : Entier, val : Reel)

**debut**

**si** (ind  $\geq$  1) et (ind  $\leq$  mat.nbElements) **alors**

    mat.valeur[ind]  $\leftarrow$  val

**finsi**

**fin**

# Structures dans des structures 1 / 2

## Principe

Une structure étant un type de données comme les autres, on peut l'utiliser pour un champ d'une autre structure

## Structure adresse

### Type Adresse = Structure

numRue : **Naturel**

nomRue : **Chaîne de caracteres**

codePostal : **Entier**

ville : **Chaîne de caracteres**

pays : **Chaîne de caracteres**

**finstructure**

# Structures dans des structures 2 / 2

## Structure employe

**Type Employe = Structure**

nom : **Chaîne de caracteres**

fonction : **Chaîne de caracteres**

salaire : **Reel**

domicile : Adresse

**finstructure**

**Type Entreprise = Structure**

nom : **Chaîne de caracteres**

SIRET : **Naturel**

pdg : Employe

employes : **Tableau[1..1000] de** Employe

nbEmployes : **Naturel**

siegeSocial : Adresse

**finstructure**

# Fichiers de structures

# Rappels : les fichiers

## Opérations de lecture/écriture

- 1 Choix du fichier
- 2 Choix du mode d'ouverture (lecture/écriture)
- 3 Lecture(s) ou écriture dans le fichier
- 4 Fermeture du fichier

## Généralités

- Un fichier est accédé par son nom et éventuellement son chemin d'accès
- dans un programme on référence un fichier par une variable

# Variables sur les fichiers

## Variables de référencement d'un fichier

- `FichierTexte` : fichier texte
- `Fichier de` : fichier typé

## Exemples

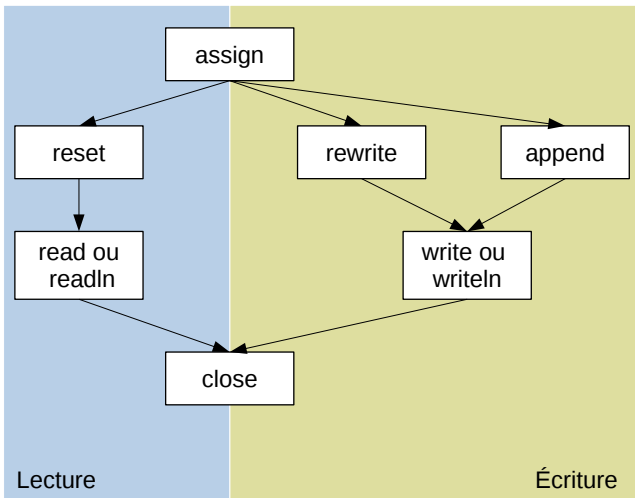
### Déclaration

`ficTex` : **FichierTexte**

`ficEnt` : **Fichier d'Entier**

`ficPer` : **Fichier de Personne**

# Récapitulatif





# Opérations

```
assigner(ficPer, '../perso/carnetAdresses.dat');
```

## Lecture

```
ouvrirLecture (ficPer)  
lire(ficPer, unePersonne)
```

## Ecriture

```
créer (ficPer)  
ecrire(ficPer, unePersonne)
```

```
fermer(ficPer);
```

# Exemple d'un fichier de structures 1 / 2

```
var nomFichier : string;
    fichier      : File of Entreprise;
    entr         : Entreprise;
begin
  write('nom_de_fichier_: ');
  readln(nomFichier);
  saisieEntreprise(entr);
  assign(fichier, nomFichier);
  rewrite(fichier);
  write(fichier, entr);
  close(fichier);
end.
```

# Exemple d'un fichier de structures 2 / 2

```
var nomFichier : string;
    fichier      : File of Entreprise;
    entr         : Entreprise;
begin
  write('nom_de_fichier_: ');
  readln(nomFichier);
  assign(fichier, nomFichier);
  reset(fichier);
  read(fichier, entr);
  close(fichier);
  afficheEntreprise(entr);
end.
```

# Ajout d'éléments dans un fichier

Il n'est pas possible en Pascal de faire :

```
var nomFichier : string;
    fichier      : File of Entreprise;
    entr         : Entreprise;
begin
    assign(fichier, nomFichier);
    append(fichier);
    write(fichier, entr);
    close(fichier);
end.
```

## Solutions

- Lire tous les éléments du fichier, les stocker dans un tableau, puis réécrire tout le fichier et ajouter le nouvel élément à la fin.
- Utiliser une structure intermédiaire de type **Tableau** de maStructure

# Exercices

# Exercice 1 / 2

Faire un logiciel qui gère l'emploi du temps d'une personne. L'emploi du temps contient un ensemble d'événements. Chaque événement a un nom et se déroule à une date donnée entre 2 horaires. Au démarrage, le programme affiche un rappel de tous les événements qui se déroulent dans les 7 jours à venir.

Définir les types de données de ce programme.

## Exercice 2 / 2

Faire un logiciel qui permet de gérer une collection de films, selon les spécifications suivantes :

Un film a

- un réalisateur, qui a un nom, prénom et une date de naissance
- un ou plusieurs acteurs, qui ont un nom, prénom et une date de naissance
- un titre,
- une année de sortie.

Définir les types de données de ce programme.

Ecrire les fonctions ou procédures permettant de saisir un nouveau film, d'en afficher un, d'afficher tous les films, de sauvegarder la collection de films, de la charger.