



5 - Structuration des information :

Types nommés

Définition de nouveaux types

Notion de type nommé

- **Type** : nom donné à un domaine de valeurs
- Exemples de domaines de valeurs :
 - entier entre 0 et 59
 - réel > 0
 - chaîne
 - booléen
- Il est possible de se **définir ses propres types**
- Ces nouveaux types permettent d'alléger l'écriture et d'améliorer la lisibilité des algorithmes
- Ils permettent aussi d'effectuer des regroupements logiques d'informations

Exemple de types nommés

Positif : type réel ≥ 0

Jour : type entier entre 1 et 31

Mois : type entier entre 1 et 12

Année : type entier entre 1583 et 2100

Minute : type entier entre 0 et 59

Heure : type entier entre 0 et 23

Où définir les types nommés ?

→ dans un **lexique partagé** entre l'algorithme principal et les fonctions ou actions qui s'y réfèrent

Exemple : somme de deux durées

- Ecrire un algorithme qui calcule et affiche la somme de deux durées données
- Une durée est formée d'heures, de minutes et de secondes

Exemple : somme de deux durées (sans type nommé)

lexique principal

cl : clavier // périphérique d'entrée
e : écran // périphérique de sortie
h1, h2: entiers ≥ 0 // données : heures de deux durées
m1, m2 : entiers entre 0 et 59 // données : minutes de deux durées
s1, s2 : entiers entre 0 et 59 // données : secondes de deux durées
h : entier ≥ 0 // résultat: heures de la somme des durées
m, s : entiers entre 0 et 59 // résultats: minutes et secondes de la somme
ns : entier ≥ 0 // intermédiaire: somme des durées convertie en secondes
r : entier entre 0 et 3599 // intermédiaire: reste de la division de ns par 3600

Algorithme principal

```
e.afficher("entrez les durées à additionner:")
cl.saisir (h1,m1,s1); cl.saisir (h2,m2,s2)
ns ← h1*3600+m1*60+s1+h2*3600+m2*60+s2
h ← ns div 3600 ; r ← ns mod 3600
m ← r div 60 ; s ← r mod 60
e.afficher (h,':',m,':',s)
```

Exemple : somme de deux durées

lexique partagé

Minutes : type entier entre 0 et 59

Secondes : type entier entre 0 et 59

lexique principal

cl : clavier // périphérique d'entrée

e : écran // périphérique de sortie

h1, h2 : entier ≥ 0 // données : heures de deux durées

m1, m2 : Minutes // données : minutes de deux durées

s1, s2 : Secondes // données : secondes de deux durées

h : entier ≥ 0 // résultat: heures de la somme des durées

m : Minutes // résultats : minutes de la somme

s : Secondes // résultats : secondes de la somme

ns : entier ≥ 0 // intermédiaire: somme des durées convertie en secondes

r : entier entre 0 et 3599 // intermédiaire: reste de la division de ns par 3600

Algorithme principal

e.afficher("entrez les durées à additionner:")

cl.saisir (h1,m1,s1); cl.saisir (h2,m2,s2)

ns \leftarrow h1*3600+m1*60+s1+h2*3600+m2*60+s2

h \leftarrow ns div 3600 ; r \leftarrow ns mod 3600

m \leftarrow r div 60 ; s \leftarrow r mod 60

e.afficher (h,':',m,':',s)

Notion d'agrégat

- Les types simples sont suffisants pour représenter des informations simples
- Mais comment représenter des informations plus complexes, par exemple, les clients d'une société ?
 - Client : un nom, un prénom, un numéro, une adresse, etc.
- **L'agrégat** sert à définir une information composée

Définition d'un agrégat

Nomdutype : type agrégat

nom-composant1 : type-de-composant1

nom-composant2 : type-de-composant2

fagrégat

Exemples :

Etudiant: type agrégat

numéro : entier > 0

nom : chaine

prénom : chaine

adresse : chaine

dnaiss : Date

fagregat

Date : type agrégat

j : Jour

m : Mois

a : Année

fagrégat

Utilisation des agrégats

- Un type agrégat permet de définir dans les lexiques des informations composées de plusieurs éléments appelés « **champs** »
- Dans un algorithme, pour désigner un champ particulier d'une variable de type agrégat, on utilise la forme suivante :

`nom_de_variable.nom_du_champ`

Constantes agrégats

d : Date

Comment affecter une valeur particulière
18/9/2017 à la Date d ?

Deux manières possibles

- Champ par champ :

$d.j \leftarrow 18 ; d.m \leftarrow 9 ; d.a \leftarrow 2017$

- Par affectation d'une constante agrégat :

$d \leftarrow \text{Date}(18, 9, 2017)$

Date(18, 9, 2017) représente une valeur particulière
du type Date

Exemple d'utilisation des agrégats

lexique partagé

Minutes : type entier entre 0 et 59

Secondes : type entier entre 0 et 59

Durée : type agrégat

h : entier ≥ 0 // heures

m : Minutes

s : Secondes

fagrégat

lexique principal

cl : clavier // périphérique d'entrée

e : écran // périphérique de sortie

d1, d2 : Durée // données : durées à additionner

d : Durée // résultat : somme des durées d1 et d2

ns : entier ≥ 0 // intermédiaire: somme de d1 et d2 convertie en secondes

r : entier entre 0 et 3599 // intermédiaire: reste de la division de ns par 3600

Algorithme principal

e.afficher("entrez les durées à additionner:")

cl.saisir (d1.h,d1.m,d1.s); cl.saisir (d2) *les 2 formes de saisie sont admises*

ns \leftarrow d1.h*3600+d1.m*60+d1.s + d2.h*3600+d2.m*60+d2.s

d.h \leftarrow ns div 3600 ; r \leftarrow ns mod 3600

d.m \leftarrow r div 60 ; d.s \leftarrow r mod 60

e.afficher (d.h,',',d.m,',',d.s)

Transformons le calcul de cet algorithme en une **fonction sommeD** qui calcule la somme de deux durées

lexique partagé (lexique des types)

Minutes : type entier entre 0 et 59

Secondes : type entier entre 0 et 59

Durée : type agrégat

h : entier ≥ 0 // heures

m : Minutes

s : Secondes

agrégat

lexique principal

cl : clavier // périphérique d'entrée

e : écran // périphérique de sortie

d1, d2 : Durée // données : durées à additionner

ds : Durée // résultat : somme des durées d1 et d2

Fonction utilisée : sommeD // fonction calculant la somme de deux durées

Algorithme principal

e.afficher("Entrez les durées à additionner:")

cl.saisir (d1, d2)

ds \leftarrow sommeD(d1,d2)

e.afficher (ds.h,',',ds.m,',',ds.s)

Transformons le calcul de cet algorithme en une **fonction sommeD** qui calcule la somme de deux durées

Fonction convDS(**x** : Durée) → entier ≥ 0
// convDS(x) renvoie le nombre de secondes correspondant à la durée x
// x : paramètre : durée à convertir en secondes
lexique de convDS
ns : entier ≥ 0 // valeur renvoyée : x convertie en secondes
Algorithme de convDS
ns ← x.h*3600 + x.m*60 + x.s
renvoyer (ns)

Fonction convSD(**n** : entier ≥ 0) → Durée
// convSD(n) renvoie la durée correspondant à n secondes
// n : paramètre : nbre de secondes à convertir en durée
lexique de convSD
d : Durée // valeur renvoyée : durée calculée à partir de n
r : entier entre 0 et 3599 // intermédiaire : n mod 3600
Algorithme de convSD
d.h ← n div 3600 ; r ← n mod 3600
d.m ← r div 60 ; d.s ← r mod 60
renvoyer (d)

Fonction sommeD(**x, y** : Durée) → Durée
// sommeD(x,y) renvoie la durée correspondant à la sommes des durées x et y
// x , y : paramètres : durées dont on veut calculer la somme
lexique de SommeD
fonctions utilisées : convDS, convSD // conversion Durée en secondes
// et secondes en Durée
Algorithme de sommeD
renvoyer (convSD(convDS(x) + convDS(y)))

Exemple : comment ajouter une seconde à une Durée d

Lexique

cl : clavier

e : écran

d : Durée

fonction utilisée : sommeD

Algorithme

cl.saisir(d)

d ← sommeD(d, Durée(0, 0, 1))

e.afficher(d.h, ':' , d.m, ':' , d.s)

