UE IG128
Organisation et exploitation des données

Année académique 2023-2024

Contenu

- Module 1: Introduction
- Module 2: Tableaux - Compléments
  - Traitement des tableaux triés
  - Bloc logique
- Module 3: Listes chaînées
- Module 4: Piles et files
- Module 5: Arbres
- Module 6: Tables de hachage

Module 2

Tableau: Compléments

2.3. Bloc logique

2.3.1. Définition

Lorsque, dans un tableau **trié selon un champ déterminé**, les cellules successives contiennent la même valeur de champ, on dit que ce champ est constitué de *blocs logiques.*

Exemple 1

Soit un tableau (**hôtels**) reprenant les coordonnées de 500 hôtels, ce tableau est **trié par catégorie**.

Chaque cellule contient la catégorie de l'hôtel (*, **, ***…), son nom et son nombre de chambres.

*	*	*	*	**	***	***	***
Au bon dodo	Au bon sommeil	Formule one	Routard	Ibis	BestWestern	Ibis Comfort	Holliday In
20	100	50	200	200	170	200	300

2.3. Bloc logique

La mise en évidence de cette structure en « blocs logiques » va permettre un traitement plus efficace de nombreux diagrammes d'actions.

Exemple 1

Écrire le module qui, à partir de ce tableau, permet d'afficher par catégorie, la catégorie et le nombre d'hôtels dans cette catégorie

2.3. Bloc logique

Exemple 1

*	*	*	*	**	***	***	***
Au bon dodo	Au bon sommeil	Formule one	Routard	Ibis	BestWestern	Ibis Comfort	Holliday In
20	100	50	200	200	170	200	300

La sortie demandée sera :

Catégorie	Nombre d'hôtels
*	**4**
**	**1**
***	**3**

2.3. Bloc logique

Conditions pour avoir une structure en blocs logiques

- La structure de données doit être triée sur un champ

  Exemple 1 : catégorie

- Le contenu des champs triés peut se répéter dans des cellules adjacentes

  Exemple 1 :

*	*	*	*	**	***	***	***
Au bon dodo	Au bon sommeil	Formule one	Routard	Ibis	BestWestern	Ibis Comfort	Holliday In
20	100	50	200	200	170	200	300

1 étoile | 2 étoiles | 3 étoiles

2.3. Bloc logique

Exemple 2

Soit un tableau d'étudiants en DA dont chaque cellule contient le libellé de l'année (B1,B2,B3) , le groupe et le login. Ce tableau est trié par année, pour une même année par groupe et pour un même groupe par login.

Question : combien de niveaux de blocs logiques? Justifiez.

B1	A	da1Abond
B1	A	da1Aspiderman
B1	...	
B1	A	da1Asuperman
B1	B	da1Blustucru
B1	B	da1Btwix
....		
B1	E	da1Eterminator
B2		

2.3. Bloc logique

- Exemple 2

B1	A	da1Abond
B1	A	da1Aspiderman
B1	…	
B1	A	da1Asuperman
B1	B	da1Blustucru
B1	B	da1Btwix
….		
B1	E	da1Eterminator
B2		

2.3. Bloc logique

Ajout dans un tableau structuré en blocs logiques

Exercice

Soit un tableau ***étudiants*** d'étudiants en informatique de ***nbEtud*** cellules contenant le groupe et le nom.

Ce tableau est **trié par groupe** et pour un même groupe **par ordre alphabétique sur le nom.**

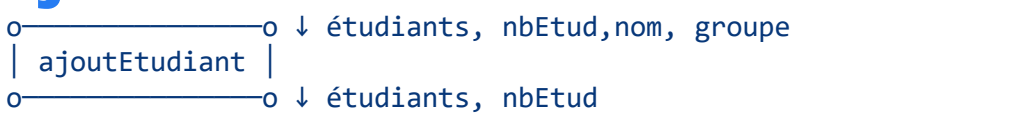
Un nouvel étudiant doit être ajouté au tableau.

Ecrire le module qui reçoit la lettre de son groupe et son nom et qui ajoute l'étudiant au bon endroit dans le tableau.

Prévoir le cas d'une erreur dans la lettre du groupe.

2.3. Bloc logique

Ajout dans un tableau structuré en blocs logiques




```
*
 //boucle de positionnement sur le groupe
iEtud = 0
while (iEtud < nbEtud and groupe > étudiants[iEtud].groupe)
 iEtud ++

if (iEtud == nbEtud or groupe < étudiants[iEtud].groupe)
 sortir "le groupe n'existe pas"
else
 // boucle de positionnement sur l'étudiant au sein du groupe
 while (iEtud < nbEtud and groupe == étudiants[iEtud].groupe  and nom > étudiants[iEtud].nom)
  iEtud ++

 iDécal = nbEtud
 while (iDécal > iEtud)
  étudiants[iDécal] = étudiants[iDécal - 1]
  iDécal --

 étudiants[iEtud].groupe = groupe
 étudiants[iEtud].nom = nom
 nbEtud ++
```

Il faut s'assurer de rester dans le bon groupe

2.3. Bloc logique

Exemple 3

Soit un tableau (**hôtels**) reprenant les coordonnées de **nbHôtels** hôtels, ce tableau est **trié par catégorie** et chaque cellule contient la catégorie de l'hôtel (*, **, ***…), son nom et son nombre de chambres.

Écrire le module qui, à partir de ce tableau, permet d'afficher pour chaque catégorie, son nombre d'étoiles et le nombre d'hôtels dans cette catégorie

Description du tableau

hôtels { cellule (nbHôtels*) { catégorie (↗) ; nom ; nbChambres }

2.3. Bloc logique

Description du tableau

- hôtels
  - cellule (nbHôtels*)
    - catégorie (↗)
    - nom
    - nbChambres

Description **logique** du tableau

- hôtels
  - blocCatégorie (…*)
    - catégorie
    - par établissement (…*)
      - nom
      - nbChambres

Sortie

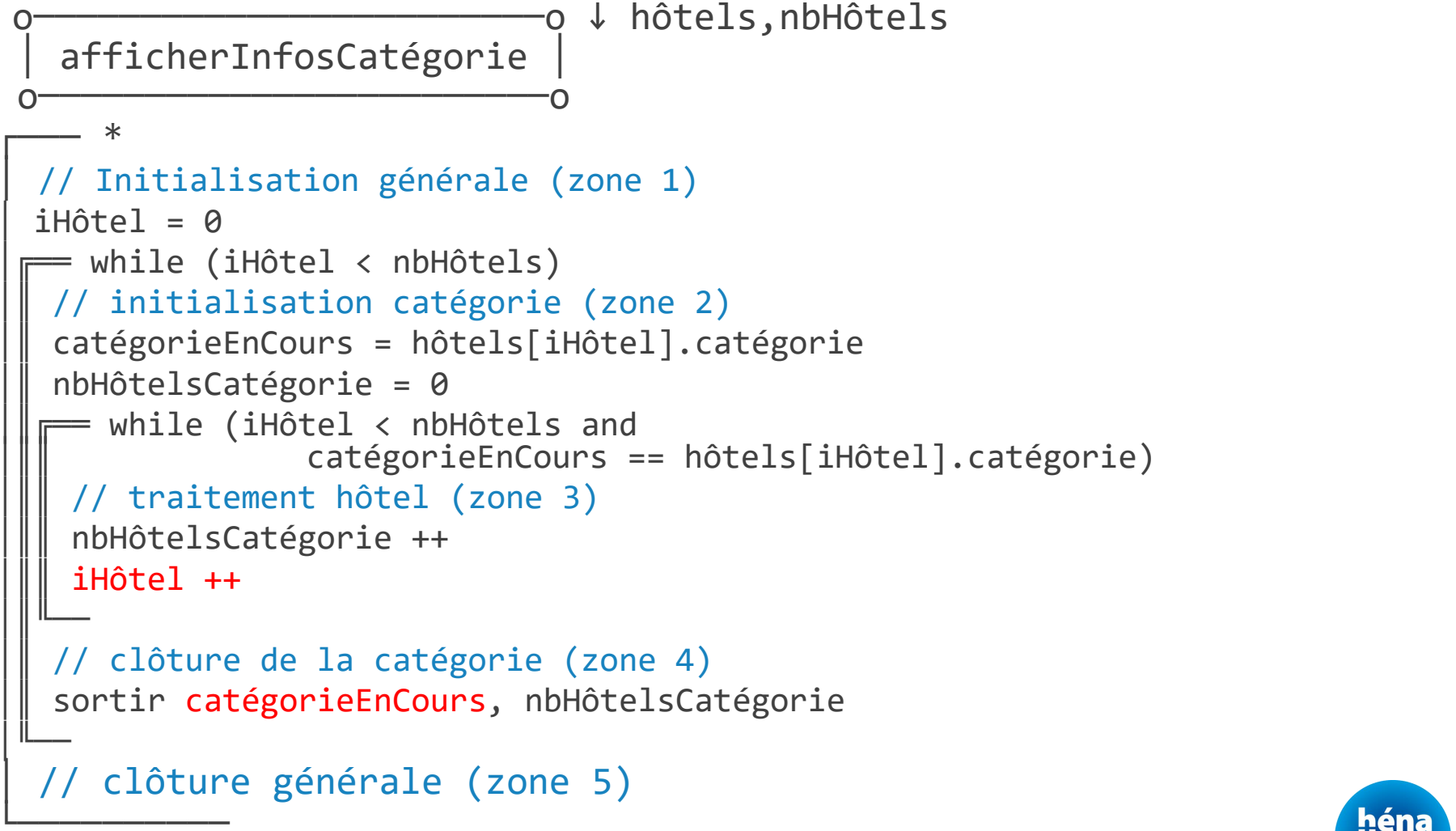
- Par catégorie
  - catégorie
  - nombre d'hôtels

2.3. Bloc logique

Ossature générale du DA

```
// Initialisation générale (zone 1)
indHotel = 0
while ( indHotel < nbHôtels )
    // initialisation de la catégorie (zone 2)
    …
    while (indHotel < nbHôtels and « même catégorie »)
        // traitement de la cellule (de l'hôtel) (zone 3)
        …
        indHotel ++
    // clôture de la catégorie (zone 4)
    …
// clôture générale (zone 5)
…
```

2.3. Bloc logique



2.3. Bloc logique: Exercice

Une société de leasing gère le parc automobile d'un certain nombre d'entreprises.

Les informations relatives aux **nbVéhicules** véhicules en leasing de la société sont retenues dans le tableau **véhicules** décrit ci-dessous:

Chaque cellule de ce tableau concerne un véhicule actuellement en leasing dans une entreprise et reprend :

- **nomEntreprise** : nom de l'entreprise
- ***plaque*** : plaque du véhicule
- ***dateFin*** : date de fin du leasing (MMJJ).

Les cellules sont classées par ordre alphabétique sur le nom de l'entreprise.

Ecrire le module qui reçoit une date (MMJJ) et affiche:

- pour chaque entreprise
  - Son nom
  - La liste des plaques de véhicules qui seront encore en leasing à la date reçue
  - Le nombre total de voitures actuellement en leasing
- le nom de l'entreprise qui a actuellement le plus de véhicules en leasing.

2.3. Bloc logique: Exercice

Description du tableau

véhicules { cellule (nbVéhicules *) { nomEntreprise, plaque, dateFin }

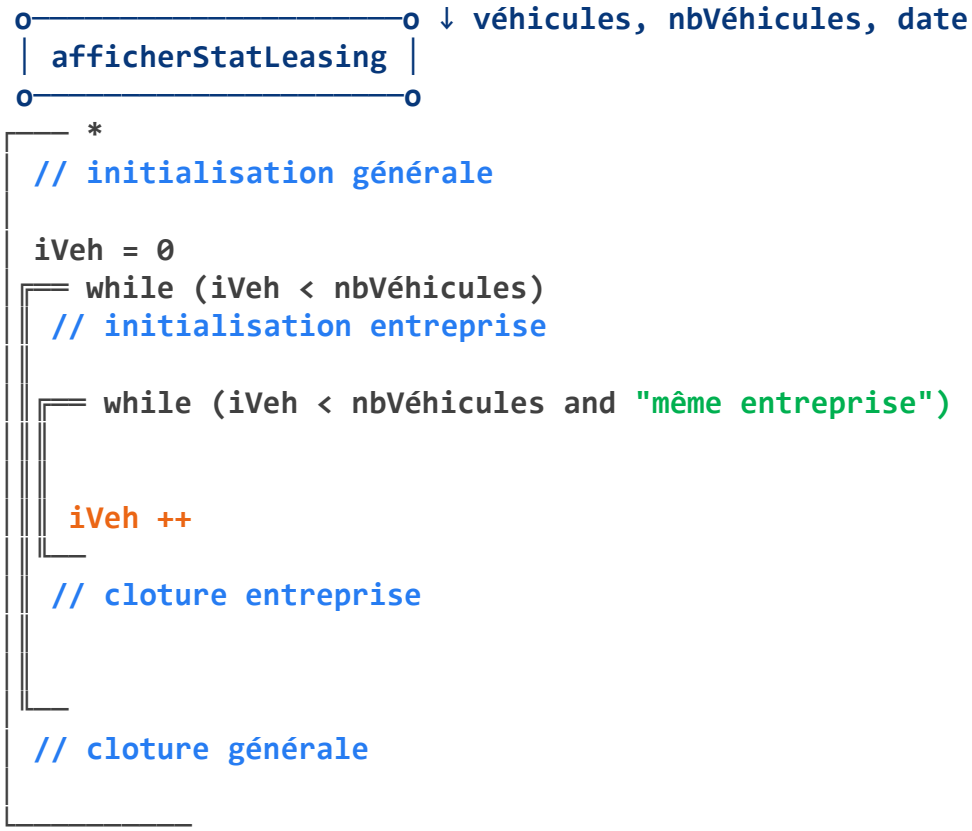
Description **logique** du tableau

véhicules { blocEntreprise (…*) { nomEntreprise, par véhic (…*) { plaque, dateFin } }

Sortie

- Par entreprise
  - Nom
  - par véhic encore en leasing à la date reçue { plaque }
  - nbVéhicules en leasing
- Nom de l'entreprise avec le max de véhic en leasing

2.3. Bloc logique: Exercice



2.3. Bloc logique: Exercice

```
o──────────────────────o ↓ véhicules, nbVéhicules, date
│ afficherStatLeasing │
o──────────────────────o
┌── *
│ // initialisation générale
│ nbVehMax = 0
│ iVeh = 0
│╔═ while (iVeh < nbVéhicules)
│║ // initialisation entreprise
│║ entrepriseEnCours = véhicules[iVeh].nomEntreprise
│║ nbVehEntreprise = 0
│║ sortir entrepriseEnCours
│║╔═ while (iVeh < nbVéhicules and entrepriseEnCours == véhicules[iVeh].nomEntreprise)
│║╠── if( véhicules[iVeh].dateFin > date)
│║│ sortir véhicules[iVeh].plaque
│║╠──
│║│ nbVehEntreprise ++
│║│ iVeh ++
│║╚──
│║ // cloture entreprise
│║ sortir nbVehEntreprise
│║╠── if (nbVehEntreprise  > nbVehMax)
│║│ nbVehMax = nbVehEntreprise
│║│ nomEntrepriseMax = entrepriseEnCours
│║╠──
│╚═
│ // cloture générale
│ sortir nomEntrepriseMax
└──────────
```