Langages objets

Tableaux

M2 Pro CCI, Informatique
Emmanuel Waller, LRI, Orsay

les tableaux

notion de tableau

déclaration

création

utilisation

représentation en mémoire

affectation de tableaux

déroulement mémoire

notion de tableau

Ensemble d'éléments de même type

Désigné par un nom

Chaque élément est repéré par un indice

but : regrouper 1000 variables int sous un seul nom (au lieu int x1, x2, ..., x1000;)

exemple

Création, remplissage, affichage certaines cases

Démonstration (Ex1)

Déclaration

int [ ] t;

Déclaration d'une variable de type tableau d'entiers

Valeur attribuée à t : comme toute variable Java

(rappel : néant + inaccessible)

Taille du tableau non précisée

Éléments : tous types Java : char, String, boolean, etc.

Rem : si tableaux = tableaux à 2 indices (vu ult.)

Variantes de syntaxe :

```
int t[];
```

Si deux tableaux :

```
int [ ] t1, t2;    // t1, t2 tableaux // la bonne : type var
int t1[ ], t2[ ];
int t1[ ], n, t2 [ ];    // n entier
```

Création

new int[5];

Création d'un tableau de 5 cases contenant chacune un entier, et initialisées à 0 (la valeur par défaut du type)

t = new int[5];

Création du tableau et affectation à la variable tableau t

Taille

t.length

Décidée à l'exécution (new int[5]), pas à compilation

Ne peut changer pendant l'exécution (voir subtilités)

Positive ou nulle

Ex : utile pour afficher tableau dont on ne connaît pas la taille

Initialisateur

Que à la déclaration

Ex 1 :

```
int[] t = {1, -2, 7, 12, 8 };
```

: crée un tableau de 5 entiers avec ces valeurs

Ex 2 :

```
int n, p;
n = . . . ; p = . . . ;    // affectation de valeurs à n, p
int [ ] t = { 1, n, n+p, 2*p, 12, -4 };
```

crée un tableau de 6 entiers ayant les valeurs données

Utilisation

Accès individuel aux éléments

Les 5 éléments : t[0], t[1], t[2], t[3], t[4]

Chacun se manipule comme une variable entière ordinaire : valeur et affectation

```
t[0] = 7;
n = t[1] + 8
System.out.println(t[2] * t[3] + 12);
```

t[-1] ou t[5] (hors bornes) : arrêt programme et message d'erreur:ArrayIndexOutOfBoundsException

exemple

création et affichage d'un tableau (démo : Ex2)

```
int [ ] t;
t = new int[5];
t[0] = 1; t[1] = -2; t[2] = 7; t[3]=12; t[4] = 8;
System.out.println(t[0]);
for (int i=0; i<t.length; i++)
    System.out.print(t[i]+" ");
```

exemples

Démonstrations

Ex3 : création par initialisateur

Ex3bis : manipulations

Ex3ter : variable non initialisée

les tableaux

notion de tableau

déclaration

création

utilisation

représentation en mémoire

affectation de tableaux

déroulement mémoire

Représentation en mémoire

Rappel : entiers : dans la mémoire

int n;

Réserve une case mémoire pour un entier (4 octets)

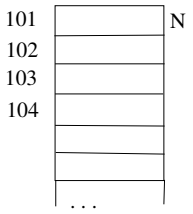
Alloue 4 octets de la mémoire, associés à la variable n de type int

n ne contient aucune valeur et est inaccessible (compilateur)

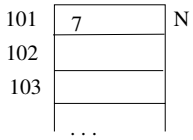
n = 7;

Écrit 7 dans cet espace de 4 octets

Int n;



n = 7;



exemple

Ex4Memoire : code source

Tableaux : dans la mémoire

int [ ] t;

Réserve un emplacement mémoire pour une référence (une adresse)

Autrement dit : alloue des octets de la mémoire, associés à la variable t de type tableau d'int

t contient la valeur « null » (= « aucune adresse »)

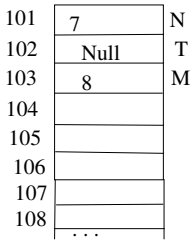
new int[5]

Alloue l'emplacement nécessaire pour 5 entiers

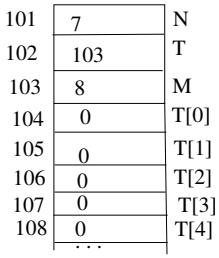
t = new int[5];

Place en plus adresse cet emplacement dans la case t

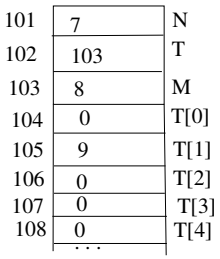
Int [ ] t;
int m = 8;



T = new int[5];



T[1] = 9;



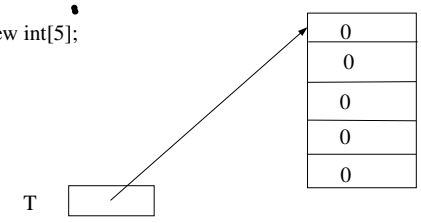
exemple

Ex4Memoire : démonstration

autre représentation mémoire ci-dessous

Int [ ] t;

T = new int[5];



Vocabulaire

On distingue

t : référence au tableau

new int[5] : le tableau lui-même

On fait des abus de langage, mais c'est clair grâce au contexte

Il existe une valeur particulière dans le type adresse : « null » (= « aucune adresse ») (démonstration : print(null))

Rem : c'est la « valeur par défaut du type »

Exemple

Ex5 : adresse d'un tableau

démonstration

les tableaux

notion de tableau

déclaration

création

utilisation

représentation en mémoire

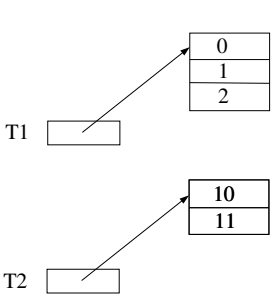
affectation de tableaux

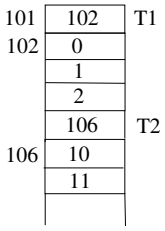
déroulement mémoire

affectation de tableaux

```
int [ ] t1 = new int[3];
for (int i=0; i<t1.length; i++) t1[i] = i;
int [ ] t2 = new int[2];
for (int i=0; i<t2.length; i++) t1[i] = 10 + i;
```

contenu (zone mémoire) : case par case
obligatoirement (rappel : sauf initialisateur)





recopie d'un tableau : erreur à éviter

t1 = t2 : pas de recopie des valeurs du tableau

le tableau créé par new int[3] ne change pas, mais il n'est plus référencé par t

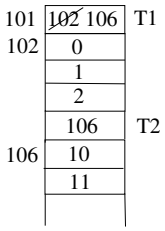
il s'agit bien d'une affectation normale, mais d'une référence

bref, affectation :

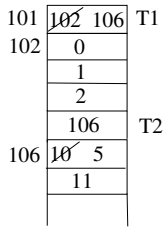
référence

zone mémoire : case par case

t1 = t2;
la valeur dans la case t2 est affectée à la case t1

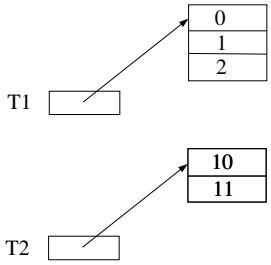



```
T1[0] = 5;
System.out.Println(t2[0]);   // affiche 5
```

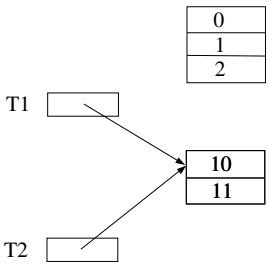


t1 = t2

avant



après



exemple

Ex6 : démonstration

la bonne recopie

t1 dans t2

case par case

dessin au tableau

exemple

Ex7 : démonstration

exemples

Ex8 : les paramètres de la ligne de commande : args est un... tableau de String (démonstration)

Ex9 : somme des éléments d'un tableau (démonstration)

Ex10 : recherche d'un élément dans un tableau (démonstration)

les tableaux

notion de tableau

déclaration

création

utilisation

représentation en mémoire

affectation de tableaux

déroulement mémoire

Déroulement mémoire

Ex11Memoire : au tableau + démonstration

les tableaux

notion de tableau

déclaration

création

utilisation

représentation en mémoire

affectation de tableaux

déroulement mémoire

(Delannoy chapitre 7)

délégués ?

Gestion d'un nombre variable d'éléments dans un tableau

- Connu : gérer un nombre fixe (ex : 25) d'entiers : tableau de taille 25
- Comment gérer un nombre variable k d'entiers, borné par n, dans un tableau ?

- On va :
  - Les mettre dans un tableau
  - De taille n
  - Dans les cases 0 à k-1
  - Mémoriser k, et le modifier quand ajout ou suppression d'élément
- Avantage : nombre « variable » d'éléments
- Inconvénients (négligeables ici)
  - On perd la place des cases non utilisées du tableau
  - Il faut gérer k

exemple

- Démonstration (Ex.java)
- Déclarations (k : prochaine case libre)
- Initialisation : créer t; k=0
- Affichage : cases 0 à k
- Saisie : placer args dans t; k=args.length
- Retirer le dernier : k--
- Ajouter i : t[k]=i; k++