INF1256 Informatique pour les sciences de la gestion

– *Tableaux et Collections d'objets* –

Johnny TSHEKE, Ing. Jr.

Université du Québec à Montréal
Département d'Informatique
tsheke_shele.johnny@uqam.ca

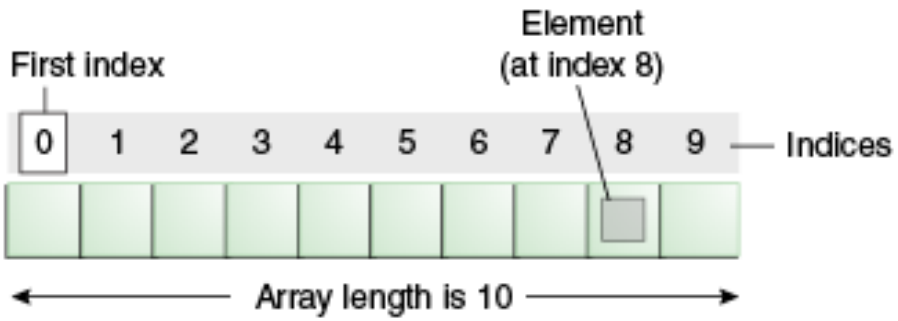
Séance 12

Tableau en Java

Dans un tableau :

- chaque élément a un ou des indices (une position précise dans la séquence)
- Les indices des tableaux sont numéroté à partir de 0. C'est à dire le premier élément du tableau a l'indice 0
- tous les objets du tableau sont de même type
- Il faut déterminer le nombre d'élément maximum à la création du tableau

Tableau à une et deux dimensions



An array of 10 elements.

(source image : `https://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html`)

une dimension : ex : int [] tab={1,4,3}

indices	0	1	2
(*elements*)	1	4	3

deux dimensions : un tableau des tableaux : int[][] tab2={{1,4,3},{4,5,6}}

1	4	3
4	5	6

par ligne ou

1	4
4	5
3	6

par colonne

Déclaration et création de tableau

Déclaration : TypeObj[] nomVariable ; (ou TypeObj nomVariable[] ;)

```
//exemple
int[] nombres; //tableau des nombres entiers
Employe[] employes;//tableau des objets classe Employe
```

Création : nomVariable = new TypeObj[nombreMax] ;

```
//exemple
nombres = new int[10];
```

Affectation élément : nomVariable[indiceElem] = ValeurElement

```
//exemple
nombres[0] = 1;
```

Création et Initialisation : nomVariable = { valeursSepareesParVirgules} ;

```
//exemple
int[] nombres = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
```

Tableau deux dimensions : TypeObj[][] nomVariable ;

```
//exemple
int[][] tab2={{1,4,3},{4,5,6}}
```

Manipulation de contenu d'un tableau en Java

- En général, faire attention aux bornes du tableau (Éviter d'avoir à manipuler un indice qui n'existe pas dans le tableau)
- En Java, dans un tableau tous les éléments sont de même type
- indices commencent à 0 et se termine à nombreElements-1
- Nombre d'éléments nomVariable.length
- une dimension : si tab={1,4,3} → tab[1] vaut 4
- deux dimensions : si tab2={{1,4,3},{4,5,6}}
  → tab2[1] vaut {4,5,6}
  → tab2[1][2] vaut 6
  tab2.length → 2 (nombre d'élément ou de sous tableaux).
  → En Java pour connaitre le nombre d'éléments d'un tableau à 2 dimensions, il faut additionner le nombre d'éléments de tous les sous tableaux parce que chaque sous tableau peut avoir un nombre différent d'éléments

Exemple de manipulation de tableau (1)

Accès aux éléments d'un tableau : Tableau.java

```
/**
 * @author Johnny Tsheke @ UQAM -- INF1256
 * 2017-03-27
 */
package s12;


public class Tableau {

	public static void main(String[] args) {
		int[] tab = {1, 4, 3};
		int[][] tab2 = {{1, 4, 3}, {4, 5, 6}};
		System.out.println("Element indice 1 tab = "+ tab[1]);
		System.out.println("Nombre élements dans tab = "+tab.length);
		System.out.println("Element coordonnées (1,2) de tab2 = "+tab2[1][2] );
		System.out.println("Nombre sous tableaux dans tab2 = "+tab2.length);

	}

}
```

Exemple de manipulation de tableau (2)

Affectation et valeurs par défaut à la création : Tableau2.java

```
/**
 * @author Johnny Tsheke @ UQAM -- INF1256
 * 2017-03-27
 */
package s12;


public class Tableau2 {

	public static void main(String[] args) {
		int[] tab = new int[10];//chaque élément initialisé à 0 par défaut
		System.out.println("Nombre élements dans tab = "+tab.length);
		tab[0] = 1;
		tab[1] = 4;
		System.out.println("Element indice 9 tab = "+ tab[9]);
		System.out.println("Nombre élements dans tab = "+tab.length);
		tab[9] = 3;
		System.out.println("Element indice 9 tab = "+ tab[9]);
		System.out.println("Nombre élements dans tab = "+tab.length);

		//tab[10] = 7; //indice 10 est hors tableau


	}

}
```

Conteneur de collection

En Java (`import java.util.*`), une collection est un conteneur d'un groupe d'objets. Il existe plusieurs type de collections :

- dictionnaire ou map (Map), associe une clé à une valeur
- liste (List) : séquence ordonnée d'objet (doublon possible)
- ensemble (Set) : pas de doublon
- queue (Queue), parcours en FIFO (`First In First out`)
- queue à deux sens (Deque – prononcé `deck` – `double-ended-queue`) permet FIFO et LIFO (`Last In First out`)
- ...

⇒ Un itérateur Itarator sur une collection est une énumération des éléments de la collection. Le lecteur intéressé pourrait consulter : `https://docs.oracle.com/javase/tutorial/collections/interfaces/index.html` ou `https://docs.oracle.com/javase/8/docs/api/java/util/Collection.html`

Ensemble Set

- Groupe d'éléments (collection) sans doublon
- Implémentation

  HashSet , utilise une table de hachage, rapide (ordre non garantie)

  TreeSet , stockage en structure arbre

  LinkedHashSet , comme une liste, élément dans l'ordre d'insertion
- Déclaration : `Set<TypeElem> nomVar = new ClasseImpl<TypeElem>();`

  `ex: Set<String> ens = new HashSet<String>();`
- méthodes : `add(elem), clear(), contains(elem), isEmpty(), iterator(), remove(elem), size(), toArray(), forEach(),...`

⇒`http://docs.oracle.com/javase/8/docs/api/java/util/Set.html`

Exemple de manipulation d'un ensemble (set)

Manipulation Ensemble : Ensemble.java

```
/**
 * @author Johnny Tsheke @ UQAM -- INF1256
 * 2017-03-27
 */
package s12;

import java.util.*;
public class Ensemble {

        public static void main(String[] args) {
                Set<String> ens = new HashSet<String>();
        ens.add("Marcel");
        ens.add("Tremblay");
        ens.add("David");
        System.out.println("ens a " + ens.size() +" eléments");
        ens.add("Marcel");//doublon mais garde une seule occurence
        System.out.println("ens a " + ens.size() +" eléments");
        ens.add("Charles");
        System.out.println("ens a " + ens.size() +" eléments");

        for(String nom:ens){
                System.out.println(nom);
        }
        ens.forEach(elem->System.out.println("impression avec foreach: "+elem));
      String[] tab = new String[ens.size()];//initialisation tab
                  ens.toArray(tab);//ATT: toArray() sans arg. retourne un tab  Objects
                  System.out.println("deuxieme nom est: "+ tab[1]);
        }
}
```

Ensemble Map

- Associe une clé à une valeur
- Pas de doublon de clé
- Impl : HashMap, TreeMap, LinkedMap,
- Ex : Déclaration :

```
Map<Integer,String> map = new HashMap<Integer,String>();
```

- méthodes : `clear`, `containsKey(K)`,`containsValue(V)`, `entrySet()`, `forEach()`, `get(K)`, `isEmpty`, `keySet()`,`put(K,V)`, `putIfAbsent(K,V)`, `replace(K,V)`, `remove(K)`, `size()`, `values()`, ...

⇒`http://docs.oracle.com/javase/8/docs/api/java/util/Map.html`

Exemple de manipulation de Map (HashMap)

Manipulation deMap : ExempleMap.java

```
/**
 * @author Johnny Tsheke @ UQAM -- INF1256
 * 2017-03-27
 */
package s12;

import java.util.*;
public class ExempleMap {
        public static void main(String[] args) {
                Map<Integer,String> map = new HashMap<Integer,String>();
        map.put(12,"Marcel");
        map.put(45,"Tremblay");
        map.put(27,"David");
        System.out.println("ens a " + map.size() +" eléments");
        map.put(12,"Marc");//cled deja utilise, remplace valeur
        System.out.println("ens a " + map.size() +" eléments");
        map.put(200,"Charles");
        System.out.println("ens a " + map.size() +" eléments");
        for(String nom:map.values()){
                System.out.println("Avec for et .values() : "+nom);
        }

        map.forEach((k,v)->System.out.println("Avec forEach: Clé= "+k+" valeur= "+v));
        for(Iterator<Integer> it= map.keySet().iterator();it.hasNext();){
                Integer cle = it.next();
                String valeur = map.get(cle);
                System.out.println("Avec iterateur: Cle= "+cle+" valeur= "+valeur);
        }
        }
}
```

Imbrication des boucles for

- Souvent utilisée pour parcourir plusieurs séquences telle qu'un élément d'une séquence S1 correspond à une autre séquence S2
- Exemple : parcourir un tableau à 2 dimensions, parcourir les éléments de chaque ligne d'un fichier texte, etc.

$\Rightarrow$ voir le cours 5 pour les détails sur l'instruction for

Exemple des boucles `for` imbriquées

parcours d'un tableau à 2 dimensions avec des boucles for imbriquées : tableau3.java

```
/**
 * @author Johnny Tsheke @ UQAM -- INF1256
 * 2017-03-27
 */
package s12;


public class Tableau3 {

	public static void main(String[] args) {
		int[] tab = {1, 4, 3};
		int[][] tab2 = {{1, 4, 3}, {4, 5, 6}};
		int sommeTab=0;
		int sommeTab2 = 0;
		for (int i=0;i<tab.length;i++){//for simple
			sommeTab = sommeTab +tab[i];
		}
		for(int i = 0; i<tab2.length;i++){//for imbriqués
			for(int j=0;j<tab2[i].length;j++){
				sommeTab2 = sommeTab2 + tab2[i][j];
			}
		}

		System.out.println("Somme des éléments tab = "+ sommeTab);
		System.out.println("Somme des éléments tab2 = "+ sommeTab2);

	}

}
```