Les Tableaux

Thomas Rey

Lycée Marlioz

6 décembre 2016

Sommaire

Un premier exemple

Observer le morceau de programme suivant :

```
nom1 = JOptionPane.showInputDialog(null,"Nom de l'élève 1",
    "Nom", JOptionPane.PLAIN_MESSAGE);
nom2 = JOptionPane.showInputDialog(null,"Nom de l'élève 2",
    "Nom", JOptionPane.PLAIN_MESSAGE);
nom3 = JOptionPane.showInputDialog(null,"Nom de l'élève 3",
    "Nom", JOptionPane.PLAIN_MESSAGE);
...
```

Il serait peut-être intéressant de créer une boucle pour saisir les 13 élèves de spé ISN, oui, mais...

Un premier exemple (suite)

- dans une boucle `for(int i=0, ...)`, l'indice `i` ne peut pas servir à définir un nom de variable ;

Un premier exemple (suite)

- dans une boucle `for(int i=0, ...)`, l'indice `i` ne peut pas servir à définir un nom de variable ;
- par exemple, `nom_i` est une **unique** variable quelque soit la valeur de `i`.

Notion de tableau

Pour créer une variable « **indicée** » on utilise un nouveau type : **le tableau**.
Un tableau est une sorte de suite (comme en mathématiques) sauf :

Notion de tableau

Pour créer une variable « **indicée** » on utilise un nouveau type : **le tableau**.
Un tableau est une sorte de suite (comme en mathématiques) sauf :

- qu'il peut contenir autre chose que des nombres (du texte, des booléens, des caractères, des objets . . .) ;

Notion de tableau

Pour créer une variable « **indicée** » on utilise un nouveau type : **le tableau**.
Un tableau est une sorte de suite (comme en mathématiques) sauf :

- qu'il peut contenir autre chose que des nombres (du texte, des booléens, des caractères, des objets ...) ;
- qu'il contient un nombre **fini** d'éléments.

Une définition

Un **tableau** est une collection de données ordonnées telle que :

- le nombre n de données est fixé (et défini au moment de la création du tableau) ;

Une définition

Un **tableau** est une collection de données ordonnées telle que :

- le nombre n de données est fixé (et défini au moment de la création du tableau) ;
- les données sont repérées par un rang (ou indice) entier compris entre 0 et $n - 1$.

Compléments

Il existe aussi des tableaux **dynamiques** qu'on peut « agrandir » ou « rétrécir » au cours de l'exécution du programme.
Nous n'en parlerons pas ici, sauf à dire leur nom :

Compléments

Il existe aussi des tableaux **dynamiques** qu'on peut « agrandir » ou « rétrécir » au cours de l'exécution du programme.
Nous n'en parlerons pas ici, sauf à dire leur nom :

les `ArrayList`

Libre à vous de vous documenter sur le sujet.

Sommaire

Déclaration en Processing

Pour déclarer un tableau de 4 entiers de type `int`, deux étapes :

- on déclare la variable :

```
int[] montableau;
```

Déclaration en Processing

Pour déclarer un tableau de 4 entiers de type `int`, deux étapes :

- on déclare la variable :

```
int[] montableau;
```

- on alloue l'espace nécessaire :

```
montableau = new int[4]; // OU BIEN :
montableau = {2,4,8,3};
```

Déclaration en Processing (suite)

On peut aussi réaliser les deux opérations en une étape :

```
int[] montableau = new int[4];// OU BIEN
int[] montableau = {2,4,8,3};
```

Déclaration en Processing (suite)

On peut aussi réaliser les deux opérations en une étape :

```
int[] montableau = new int[4];// OU BIEN
int[] montableau = {2,4,8,3};
```

Dans le premier cas, chaque élément du tableau est initialisé à la valeur 0.
Dans le deuxième cas, on a affecté une valeur à chaque élément du tableau.

Résumé

Une déclaration de tableau dont on ne connait pas le contenu au départ se fait ainsi :

```
TypeElement nomdutab[] = new TypeElement[n];
// TypeElement à choisir parmi int, double,
   String, ...
```

Résumé

Une déclaration de tableau dont on ne connait pas le contenu au départ se fait ainsi :

```
TypeElement nomdutab[] = new TypeElement[n];
// TypeElement à choisir parmi int, double,
   String, ...
```

ou ainsi :

```
TypeElement[] nomdutab;
nomdutab = new TypeElement[n];
```

Sommaire

Écrire et lire dans un tableau

En maths, pour indiquer que le 5^e terme d'un suite vaut 10 on écrit $u_4 = 10$, en Java si on veut affecter la valeur 10 au 5^e élément d'un tableau `montab`, on écrit :

```
montab[4] = 10;
```

Écrire et lire dans un tableau

En maths, pour indiquer que le 5^e terme d'un suite vaut 10 on écrit $u_4 = 10$, en Java si on veut affecter la valeur 10 au 5^e élément d'un tableau `montab`, on écrit :

```
montab[4] = 10;
```

Exemple : que fait ce « bout » de code ?

```
int montab[] = new int[10];
for(int i = 0; i < 10; ++i) {
  montab[i] = 2*i;
}
```

Réponse

On vérifie avec ce code :

```
for(int i = 0; i < 10; ++i) {
    print(montab[i] + "; ");
}
```

qui affiche :

0; 2; 4; 6; 8; 10; 12; 14; 16; 18;

Autre méthode d'affichage des éléments d'un tableau :

```
for (int val : montab) {// on boucle sur tous
    les éléments du tableau
    print(val+"; ");
}
```

Longueur

Pour retrouver la taille d'un tableau (par exemple dans une fonction ou une procédure) :

```
int longueur = montableau.length;
```

Exemple : que fait cette procédure ?

```
void quefaisje(int[] table) {
  for(int i = 0; i < table.length; i++) {
    print(table[i]+";");
  }
  println();
}
```

Attention !

On donne le code suivant :

```
int[] tab = new int[3];
tab[0] = 0; tab[1] = 1; tab[2] = 2;
int[] tabBis = {0,1,2};
if (tab == tabBis) {
  println("Même tableau");
} else {
  println("Tableaux différents");
}
```

Devinette : que va-t-il afficher ?

Attention !

On donne le code suivant :

```
int[] tab = new int[3];
tab[0] = 0; tab[1] = 1; tab[2] = 2;
int[] tabBis = {0,1,2};
if (tab == tabBis) {
  println("Même tableau");
} else {
  println("Tableaux différents");
}
```

Devinette : que va-t-il afficher ?

```
Tableaux différents
```

Explication

- le code `int[tab]` crée une variable `tab` qui contient une **adresse** de la mémoire où seront stockés les éléments du tableau ;
- le code `int[] tabBis` crée une **autre** variable qui contient une **autre** adresse en mémoire.

Explication

- le code `int[tab]` crée une variable `tab` qui contient une **adresse** de la mémoire où seront stockés les éléments du tableau ;
- le code `int[] tabBis` crée une **autre** variable qui contient une **autre** adresse en mémoire.

On peut s'en convaincre en écrivant :
`System.out.println(tab + "; "+ tabBis);` qui donne un affichage du type :

```
[I@1f01b29; [I@3a8721bd
```

Explication (suite)

Donc les tableaux ont les mêmes contenus mais sont stockés dans des adresses **différentes**.
La ligne `if (tab == tabBis)` teste les **adresses** et non pas les contenus.
En définissant le tableau `tabBis` par :

```
int[] tabBis = tab;
```

les deux tableaux auraient la **même** adresse et le test serait positif.

Remarque

On avait rencontré exactement le même type de problème pour comparer les variables de type `String` :

- le type `String` est un type **évolué** ;

Remarque

On avait rencontré exactement le même type de problème pour comparer les variables de type `String` :

- le type `String` est un type **évolué** ;
- il se comporte comme un tableau de `char` ;

Remarque

On avait rencontré exactement le même type de problème pour comparer les variables de type `String` :

- le type `String` est un type **évolué** ;
- il se comporte comme un tableau de `char` ;
- la condition (`maChaine1 == maChaine2`) teste si les deux ont la même **adresse** ;

Remarque

On avait rencontré exactement le même type de problème pour comparer les variables de type `String` :

- le type `String` est un type **évolué** ;
- il se comporte comme un tableau de `char` ;
- la condition `(maChaine1 == maChaine2)` teste si les deux ont la même **adresse** ;
- la condition `(maChaine1.equals(maChaine2))` teste si les deux chaines ont le même **contenu**.

Sommaire

Un exemple

Comment stocker dans une seule variable les informations numériques suivantes ?

0	1	2	3
1	2	3	4
2	3	4	5

Un exemple (suite)

On va créer un tableau à **deux** indices (un peu à la manière des coordonnées de points dans un repère) :

```
int tabadd[][] = new int[4][3];
for(int y=0; y<3; y++){
  for(int x=0; x<4; x++){
    tabadd[x][y]=x+y;
  }
}
```

Sommaire

On définit un tableau de six joueurs :

```
Joueur [] joueurs = new Joueur[6];// le
    tableau des joueurs
```

On définit un tableau de six joueurs :

```
Joueur [] joueurs = new Joueur[6];// le
    tableau des joueurs
```

On affiche le nom du premier joueur :

```
println(joueurs[0].nom);
```

On définit un tableau de six joueurs :

```
Joueur [] joueurs = new Joueur[6];// le
    tableau des joueurs
```

On affiche le nom du premier joueur :

```
println(joueurs[0].nom);
```

On crédite son compte de 50 € :

```
joueurs[0].gagne(50);
```

Sommaire

Où sont les données ?

Pour remplir un tableau avec des données, on peut :

- les saisir dans le programme (peu interactif) ;

Où sont les données ?

Pour remplir un tableau avec des données, on peut :

- les saisir dans le programme (peu interactif) ;
- les demander à l'utilisateur (fastidieux) ;

Où sont les données ?

Pour remplir un tableau avec des données, on peut :

- les saisir dans le programme (peu interactif) ;
- les demander à l'utilisateur (fastidieux) ;
- les lire dans un fichier texte (pratique !).

Lire dans un fichier connu

On va stocker le fichier dans un tableau de chaînes à définir :

```
String[] contenuFichier;
```

Lire dans un fichier connu

On va stocker le fichier dans un tableau de chaînes à définir :

```
String[] contenuFichier;
```

Si `fichier.txt` est dans le dossier `data` de votre sketch :

```
contenuFichier = loadStrings("fichier.txt");
```

Lire dans un fichier connu

On va stocker le fichier dans un tableau de chaînes à définir :

```
String[] contenuFichier;
```

Si `fichier.txt` est dans le dossier `data` de votre sketch :

```
contenuFichier = loadStrings("fichier.txt");
```

Le tableau contient désormais autant d'entrées qu'il y a de lignes dans `fichier.txt` :

```
int nbLignes = contenuFichier.length;
for (int l = 1; l<nbLignes; ++l) {
  println(contenuFichier[l]);
}
```

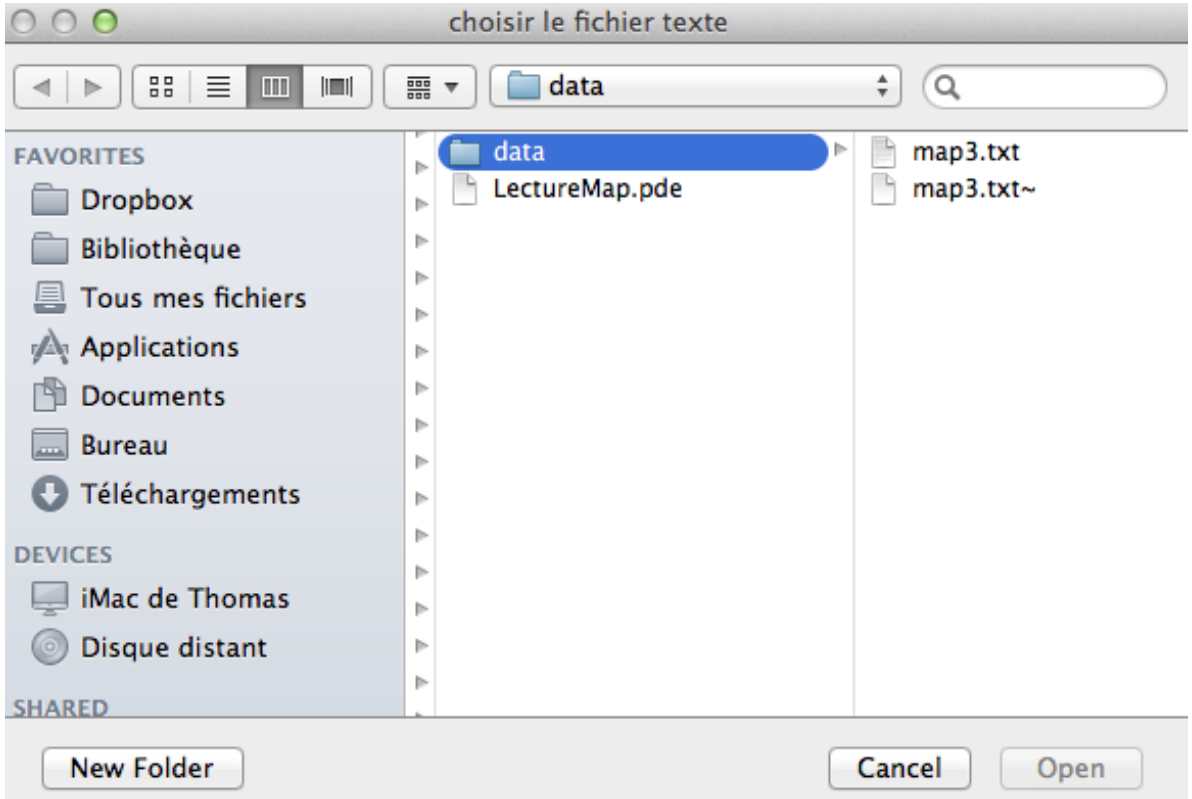
Choisir un fichier

On peut aussi demander à l'utilisateur de sélectionner le fichier dans une boîte de dialogue :

```
String[] contenuFichier;
selectInput("choisir le fichier texte ", "
    choixFait");
```

`choixFait` est alors le nom d'une méthode qui sera lancée dès que l'utilisateur a cliqué sur OK.

Boite de dialogue de choix de fichier



La méthode choixFait

Cette méthode contient au minimum :

```
void choixFait(File fichierChoisi) {
  if (fichierChoisi == null) {//pas de
      fichier
    println("Erreur : pas de fichier ou
        illisible");
  } else {
    monTexte = loadStrings(fichierChoisi.
        getAbsoluteFile());
  }
}
```

Elle peut évidemment se nommer autrement.

Écrire dans un fichier

La méthode est comparable à l'écriture dans la console :

- On crée le fichier :

```
PrintWriter output;
output.createWriter("records.txt");
```

Écrire dans un fichier

La méthode est comparable à l'écriture dans la console :

- On crée le fichier :

```
PrintWriter output;
output.createWriter("records.txt");
```

- On écrit : `output.println("Thomas:10000");`

Écrire dans un fichier

La méthode est comparable à l'écriture dans la console :

- On crée le fichier :

```
PrintWriter output;
output.createWriter("records.txt");
```

- On écrit : `output.println("Thomas:10000");`
- On ferme :

```
output.flush();
output.close();
```