Claudine Noblet <clnoblet@gmail.com>

A partir du tutoriel javascool sur les tableaux.

La notion de tableau :

Un tableau constitue la manière la plus efficace pour stocker et accéder aléatoirement à une séquence de données. Chaque donnée est référencée par un index compté à partir de zéro, par exemple, en écrivant:

```
String names = {"Alice", "Bob", "Samia", "Zheng-You"};
```

nous avons créé une collection de quatre jolis prénoms, et la variable noms[0] a pris la valeur "Alice" et, par exemple, noms[2] a pris la valeur "Samia". On peut lire une valeur d'un tableau, par exemple:

```
String him = noms[3];
```

donne la valeur "Zheng-You" à la variable de nom him. On peut aussi changer la valeur d'un tableau, par exemple:

```
noms[2] = "Samiha";
```

changera la 3ème valeur du tableau, celle d'index 2.

La taille du tableau :

La taille[1] d'un tableau est fixée lors de la création et ne peut plus être changé pendant toute la durée de sa vie. La taille d'un tableau est donnée par la variable noms.length (dans notre cas elle vaut 4).

Si on tente d'accéder à une case du tableau qui n'existe pas (par exemple noms[-1] ou noms[4]), alors une erreur est générée lors de l'exécution du programme.

Travail proposé.

1. Manipuler un tableau de valeurs numériques.

Exercice 1 :

- Créer un tableau de 4 nombres entiers ;
- le remplir a avec des nombres tirés au hasard entre 0 et 9 ;
- imprimer toutes les valeurs du tableau.

Exercice 2 :

Concevoir et imprimer le tableau des dix premiers carrés : 0, 1, 4, 9, ...

1. Trouver l'erreur !!

```
int tab[] = new int[5];
tab[0] = 5;
tab[1] = 9;
tab[2] = "trois";
tab[3] = 59;
```

Le morceau de programme ci-contre est-il correct ? Expliquez pourquoi.

Noter aussi que l'élément tab[4] n'a pas été initialisé c'est à dire que personne n'a donné de valeur à cette case du tableau.
Faire un petit code pour vérifier à quelle valeur Java initialise les valeurs des tableaux par défaut. Lancer le code plusieurs fois, pour expérimenter.

2. **Appliquer une fonction à un tableau.**

Créer la fonction:

```
void double_les_elements(int tab[]) {

   ../..

}
```

qui prend un tableau et multiplie par deux tous ses éléments. Ecrire un petit code qui illustre cette fonction.

Nous notons ici que la référence du tableau est passée à la fonction qui peut donc le modifier. Le tableau n'est donc pas recopié avant d'être passé à la fonction.

3. **Tableaux à plusieurs dimensions.**

Ecrite un algorithme qui remplisse ce tableau à deux dimensions:

```
int table[][] = new int[11][11];
```

avec la table de multiplication, c'est à dire de manière à ce que table[i][j] soit le produit de i par j et écrire le résultat sous forme de table.

On voit ici qu'il est très facile de passer à des tableaux de dimensions quelconques.

Aller plus loin: un exemple de jeu avec des tableaux..

Prendre connaissance du petit programme , proposé par des lycéens, qui permet de jouer jeu du pendu et s'en inspirer pour créer un autre jeu de crayon comme le mastermind . !

Remarques.

1. **Tableau et références.**

Dans la mémoire de l'ordinateur, le tableau mémorise juste les références des données ou des objets qui le composent. Les données ne sont donc pas "recopiées dans le tableau".

2. **Créer et détruire les objets en mémoire.**

Il faut créer un tableau avant de pouvoir l'utiliser, mais en Java pas besoin de se soucier de le détruire quand on ne s'en sert plus: ce sera détecté et géré automatiquement. D'autres langages comme le C/C++ obligent à explicitement détruire les objets qui ne servent plus.

3. **D'autres structures de données**

Il existe des "piles", des "tables", etc. de taille variable, mais c'est une autre histoire…