

TP2 : LES VARIABLES ET OPERATIONS DE LECTURE

OBJECTIFS

- Savoir déclarer une variable.
- Savoir utiliser le bon type de variable en fonction du problème.
- Effectuer des saisies utilisateurs.
- Comprendre le mécanisme du cast.

Pour commencer :

- Créer un nouveau Projet nommé TP2.
- Ajouter un fichier Java Main Class pour chaque Exercice (rappel : Run-File pour exécuter le fichier)

EXERCICES

EXERCICE 1 :

On suppose les variables du tableau. Elles sont toutes de type int. Dans un fichier **Exercice1.Java** tenter de déclarer ses variables. Indiqué dans le tableau si elles sont valides ou pas, et si non valide en expliquer la raison.

	Valide (oui / non) ? Si non Pourquoi ?
age	
!Ex	
R79JH3	
Revenu98	
_invalide	
élu	
long	
3v	

EXERCICE 2 :

Donner le type et le nom de la variable destinée à recevoir les informations suivantes, ainsi que la déclaration de cette variable. On choisira le type le mieux adapté.

Description	Déclaration en langage Java
Votre age.	
La surface d'un terrain de football.	
Le nombre de galaxie dans l'univers.	
Le nombre d'habitant de la ville de Paris.	
Le nombre d'habitant dans le monde entier.	
Le nombre d'étoile dans l'univers.	

EXERCICE 3 : LA CLASSE SCANNER

Cette classe permet de lire les entrées clavier, aussi appelé saisie utilisateur.

Pour l'utiliser il faut commencer par importer cette classe dans la zone appropriée.

```
import java.util.Scanner;
```

Ensuite il faut déclarer objet qui sera de type Scanner. Un seul objet suffit pour toutes les saisies, donc une seule déclaration :

```
Scanner sc = new Scanner(System.in);
```

Dans l'exemple l'objet créé se nomme « sc ». A partir de cet objet il va être possible de faire des saisies de valeurs selon le type de donnée voulue. Cette classe se trouve dans le package « **java.util** ».

Pour cela lire la documentation Java (JavaDoc) sur le sujet.

<https://docs.oracle.com/javase/8/docs/api/java/util/Scanner.html#method.summary>

Exemple d'utilisation de la méthode `nextInt()` : tester ce code dans votre programme.

```
Scanner sc = new Scanner(System.in);  
int a;  
a = sc.nextInt(); // a prend la valeur de qui a été saisie au clavier.
```

A faire :

Coder un programme permettant d'effectuer la saisie des différents types de variable dans cet ordre précis : byte, short, int, long, float, double, boolean. Pour chaque type de donnée trouver la méthode `next...()` adapté à la saisie.

A faire :

Continuer le programme précédent en ajoutant la possibilité de saisir un message texte. Il y a 2 méthodes le permettant. Trouver les et expliquer leurs différences après les avoir testées.

A faire : Reprendre l'exemple suivant :

```
Scanner sc = new Scanner(System.in);  
int a;  
a = sc.nextInt(); // a prend la valeur de qui a été saisie au clavier.
```

Que se passe-t-il si l'utilisateur saisie les informations suivantes ? Conclure en expliquant le principe de fonctionnement du Scanner.

- 12.5
- Bonjour
- J'ai 20 ans
- 20 ans c'est mon âge
- 12 + 3
- 2147483647
- 2147483646

A faire : Reprendre l'exemple suivant :

```
Scanner sc = new Scanner(System.in);  
float a;  
a = sc.nextFloat();
```

Tester la saisie des nombres suivant. Conclure sur le format de saisie des nombres réelles.

- 20
- 12.5
- 2e6
- 12,5
- 12,5 ans
- J'ai 20,5 ans
- 20,5 c'est mon âge

A faire : Reprendre l'exercice précédent. Trouver une solution pour que le séparateur des nombres à virgule ne soit plus la « , » mais le point. Lire la documentation suivante pour trouver la réponse:

[https://docs.oracle.com/javase/7/docs/api/java/util/Scanner.html#useLocale\(java.util.Locale\)](https://docs.oracle.com/javase/7/docs/api/java/util/Scanner.html#useLocale(java.util.Locale))

EXERCICE 4 – LE MECANISME DU CAST.

Dans cet exercice on gère des notes. Pour cela on va utiliser le type « byte » pour chaque note. En fait on va gérer 5 notes (n_1 , ..., n_5).

1. Ecrire un programme permettant de saisir les 5 notes puis les afficher. Tester votre programme.
2. Maintenant nous allons faire la somme des 5 notes précédemment saisies. Le résultat sera affecté à une variable « somme » de type byte également. Corriger l'erreur provoqué avec le mécanisme du cast.
3. Afficher maintenant la moyenne des 5 notes. Pour cela ajouter une variable « moyenne » de type « float ». Vérifier que le résultat affiché correspond bien à un réel. Si ce n'est pas le cas trouver au moins 2 solutions pour que cela soit le cas.
4. Trouver une solution pour que la moyenne soit arrondie à l'entier inférieur.
5. Trouver une solution pour que la moyenne soit arrondie à l'entier supérieur.

EXERCICE 5 – PROBLEME EN BETON

La cimenterie Lafarge souhaite disposer d'une application permettant de calculer le volume de béton nécessaire pour une dalle en béton armé dosé à 350kg/m³ (350 kg de ciment pour 1 m³ de béton) pour des maisons individuelles, ainsi qu'un outil d'aide au calcul de la facture.

L'épaisseur minimal de la dalle est de 15 cm pouvant aller jusqu'à 35 cm. Ensuite le volume dépend bien sûr de la surface plancher de l'habitation.

Afin de préparer au mieux le mélange béton l'application indiquera la quantité en tonne de ciment nécessaire.

Dernière précision, la société propose uniquement des livraisons de camions de 9m³ (ni plus, ni moins). Il faudra donc adapter le volume en fonction de ce critère à l'entier supérieur. L'application indiquera donc le nombre de camions nécessaires.

Le prix du m³ de béton est actuellement de 91€ HT, le prix du transport et de 140€ par livraison. L'application indiquera donc le tarif HT et TTC du devis.

Pour résoudre ce problème nous allons procéder de la façon suivante :

- Lister les données nécessaires. Celles qui sont en entrées, et celles qui sont des résultats.
- Trouver les règles de calcul qui vont permettre de transformer nos entrées en sortie.
- Ecrire l'algorithme sur papier
- Le coder en JAVA.
- Tester le programme avec différentes entrées, vérifier le résultat.