

T.P. N°5 en Programmation Orientée Objet en JAVA

Travail dans le package chapitre 2

Exercice 1

Ecrire la classe suivante avec Eclipse.

```
package chapitre2 ;
abstract class Ex1{
int x ;
String z;
abstract int m1();
void m2(){int p;
System.out.println(z);
System.out.println(z.length()); //affiche le nombre de caractères de z
}
}
```

Questions:

- 1) Ajouter une instruction dans la méthode m2() qui crée un objet c de type Ex1. Est-ce que l'instruction compile ? justifier **EX1 c= new EX1;**
..... **l instruction ne compile pas car il n'est pas possible d instancier d'un class abstract .**

Mettre l'instruction en commentaire

- 2) Est-ce que c'est possible d'ajouter le modificateur final à la classe Ex1 ? pourquoi ?
..... **il n est pas possible d'ajouter le modificateur final car abstract signifie que la class doit être hérité**
- 3) Ecrire une classe abstract SubEx3 qui hérite Ex1 et déclare une méthode abstract m3 de type void
..... **package chapitre2;**
..... **abstract class SupEX3 extends EX1 {**
.....
..... **abstract void m3();**
..... **}**
- 4) Ecrire une classe concrète Sub1Ex1 qui hérite deSubEx3 avec une méthode main qui crée une instance de Sub1Ex1 et appelle la méthode m2() ;
.....
.....
.....
.....
.....

a- Que dois faire la méthode sub1Ex1 pour compiler ?
.....

b- Définir la méthode m1 qui doit afficher « ici m1 » et retourne la valeur de x, et la méthode m3() qui affiche « ici m3 »
.....
.....

c- Exécuter la classe Sub1Ex1

Exercice 2

1. Développer une classe Essai qui permet d'inviter l'utilisateur à entrer une chaîne de caractères et de lire la chaîne de caractères entrée par l'utilisateur et l'afficher:

.....

.....

.....

.....

.....

.....

.....

2. Modifier la classe précédente de tel façon l'utilisateur peut entrer plusieurs lignes. La saisie s'arrête si l'utilisateur entre la chaîne « ok »

.....

.....

.....

.....

3. Inviter l'utilisateur pour entrer un entier, saisir l'entier et l'afficher

.....

.....

.....

La classe Scanner

Elle simplifie la lecture de données sur l'entrée standard (clavier) ou dans un fichier.

Pour utiliser la classe Scanner, il faut d'abord l'importer :

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

Ensuite il faut créer un objet de la classe Scanner. Pour pouvoir récupérer ce que va être tapé sur la console, il faut initialiser l'objet Scanner avec l'entrée standard, **System.in**.

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

Pour récupérer les données, il faut faire appel sur l'objet sc aux méthodes décrites ci-dessous.

Ces méthodes parcourent la donnée suivante lue sur l'entrée et la retourne :

`String next()` donnée de la classe String qui forme un mot,

`String nextLine()` donnée de la classe String qui forme une ligne,

`boolean nextBoolean()` donnée booléenne

`int nextInt()` donnée entière de type int,

`double nextDouble()` donnée réelle de type double

Il peut être utile de vérifier le type d'une donnée avant de la lire :

`boolean hasNext()` renvoie `true` s'il y a une donnée à lire
`boolean hasNextLine()` renvoie `true` s'il y a une ligne à lire
`boolean hasNextBoolean()` renvoie `true` s'il y a un booléen à lire
`boolean hasNextInt()` renvoie `true` s'il y a un entier à lire
`boolean hasNextDouble()` renvoie `true` s'il y a un double à lire.