

Séance Exercices 1

Les exercices suivants sont une série de pointeurs sur les connaissances de base en java. Les connaissances de cette première série d'exercices sont fondamentales. Les corrigés sont disponibles, il vous est conseillé de lire le corrigé après chaque question pour pouvoir poursuivre l'exercice. Abordez ceux des exercices qui vous semblent pertinents en sachant que :

- Ex 1 concerne la création de programme et le rôle de la fonction main
- Ex 2 concerne l'initialisation de variables et la construction d'objets
- Ex 3 concerne les chaînes de caractères, le type char et la concaténation.
- Ex 4 concerne les tableaux d'objets.
- Ex 5 concerne la création de fonctions pour un objet
- Ex 6 concerne les structures conditionnelles
- Ex 7 est plus anecdotique, il vise à trouver les équilibres de Nash à partir d'une matrice de scores.
- Ex 8 propose l'implémentation d'un petit jeu

Ex 1

1- On crée un fichier Test.java qui contient seulement :

```
public class Test
```

- Que se passe t'il si on cherche à le compiler ?

2- On crée une classe Test qui est la forme minimale que peut avoir une classe :

```
public class Test
```

```
{  
}
```

L'ordre de compilation pour cette classe est javac Test.java, l'ordre d'exécution est java Test.

- La compilation fonctionne t'elle ?
- Que se passe t'il lorsque on demande à la machine d'exécuter le programme Test en faisant java Test en ligne de commande
? Quelle message renvoie la machine ? Comment interprétez vous ce message ?

3- On modifie la classe Test :

```
public class Test{

    public static void main(String[] args)
    {
    }
}
```

à l'identique, on la compile (javac Test.java) puis on l'exécute : java Test.

- Que constatez vous A l'exécution ?
- Peut-on envisager de faire un programme plus court que celui-ci ?

4- On crée une classe Test qui est de la forme :

```
public class Test{

    public static void main(String[] args)
    {
    System.out.println("Ceci vient du main de Test");
    }
}
```

- Que donne l'exécution de ce code ?
- Question complémentaire : que donnerait la compilation et l'exécution de ce programme :

```
public class Test{

    public static void main(int args)
    {
    System.out.println("Ceci vient du main de Test");
    }
}
```

Ex 2

1- On veut exécuter le code suivant

```
int i;
System.out.println(i);
ActifNegotiable a;
System.out.println(a);
```

- Créez une classe Test contenant une fonction main qui exécute ces lignes de code. La compilation fonctionne t'elle ?

2- Corrigez l'erreur de la question précédente. On veut que `ActifNegotiable` soit une classe avec les attributs suivants :

`double cours;`

`String nom;`

- Créez cette classe de manière à ce qu'elle soit "vue" par la classe `Test.java` (il existe trois solutions, vous devez en connaître au moins deux à ce stade).

3- Une fois la classe `ActifNegotiable` définie, on recompile la classe `Test` et on relance l'exécution des lignes de code de 1-

- L'exécution provoque t'elle encore une erreur ?

4- On modifie le code la fonction `main` :

```
int i=0;
```

```
System.out.println(i);
```

```
ActifNegotiable a=null;
```

```
System.out.println(a);
```

Que se passe t'il à l'exécution de ce code ?

- L'exécution ne provoque plus d'erreur. Comment comprenez vous ce qui apparaît à l'écran ?

5- On rajoute une instruction :

```
int i=0;
```

```
System.out.println(i);
```

```
ActifNegotiable a=null;
```

```
System.out.println(a);
```

```
System.out.println(a.cours);
```

- Quel affichage constate t-on ? Qu'est ce qui vous paraît manquer ? Corrigez le code sans éliminer aucune instruction : vous avez juste le droit d'en modifier une. Expliquez précisément l'affichage qui s'en déduit.

6- On veut maintenant qu'il soit impossible de construire un objet `ActifFinancier` sans lui attribuer une valeur, de sorte que la valeur du `cours` ne soit pas 0 après la construction de l'objet.

7-

```
int i;
```

```
System.out.println(i);
```

```
ActifNegotiable a;
```

```
a=new ActifNegotiable(5);
```

```
System.out.println(a);
```

```
System.out.println(a.cours);
```

```
ActifNegotiable a2=new ActifNegotiable();
```

Que se passe t'il à l'exécution de ce code ? Si finalement on veut pouvoir construire des `ActifNegotiable` en disposant de leur `cours` initial ou sans en disposer, comment faire ?

Ex 3

0- Quelques éléments préliminaires. String est une classe. Elle s'instancie par un constructeur qui prend une chaîne en entrée :

```
String str=new String("Coucou");
```

Pour cette classe, une autre forme d'instanciation est possible et équivalente :

```
String str="Coucou";
```

Quand on utilise `System.out.println("");`, on utilise en fait la fonction `System.out.println(String str);`. Donc

```
String str="Coucou";  
System.out.println(str)
```

donne le même affichage que

```
System.out.println("Coucou");
```

1- Quelle différence faites-vous entre "c", 'c' et c ?

2- Dans quelles conditions le test `v=="c1"` ne provoquera pas d'erreur à la compilation ? Qu'en est-il des tests `v=='c1'` et `v==c1` ?

3- Soit le code suivant, qu'affiche t'il ? Pourquoi ?

```
char c='a';  
char x='a';  
boolean bo=(c==x);  
System.out.println(bo);  
String str="President";  
String str2="President"  
boolean b=(str==str2);  
System.out.println(b);
```

Dans le code qui précède combien d'objets String ont été créés ?

4- Soit le code suivant, que produit-il ?

```
String str="";          int i=7;  
str+='P';              str+="r";   str+='e';   str+="s";   str+="i";   str+="d";  
str+='e';              str+="n";   str+='t'; str+=i;  
System.out.println(str);
```

Quelle différence faites vous entre `String str1=""`; et `String str2=" "`;

5- Concaténation de deux String :

```
String str1="Les sanglots longs ";  
String str2="des violons de l'automne";  
String str3=str1+str2;  
boolean b1=(str3==str1);  
boolean b2=(str3==str2);  
System.out.println(b1+" "+b2+" "+str3);
```

Quel affichage produit ce code ?

6- Quelle différence faites vous entre "2.55" et 2.55 ? Comment passer de l'un à l'autre ?

7- Création d'une classe de chaînes URL, soit une classe de chaîne contenant seulement des String commençant par "http://". A la construction des objets, de cette classe, il faut s'assurer de bien avoir une telle chaîne.

Ex 4

Que donne le code suivant :

```
int[] i=new int[100];
System.out.println(i[5]);
A[] a=new A[100];
System.out.println(a[5]);
a[6]=new A();
```

En utilisant une classe minimale A visible par la classe contenant le main :

```
class A{ }
```

Ex 5

On veut créer une classe Utilitaire qui propose une fonction permettant d'obtenir le max entre deux variables de type int, entre deux variables de type double, entre deux variables char (la machine dispose d'un ordre total sur les char, on peut ainsi faire des comparaisons de la forme 'c'>'n', 'n'>'t'). Combien de fonctions faut il implémenter ? Proposez une fonction qui de deux String renvoie la chaîne maximale ou la chaîne null si jamais les deux chaînes ont le même contenu. Proposez une méthode qui renvoie la valeur max au sein d'un tableau de String.

Ex 6

On considère l'instruction **if(a && b)**

- Est-ce que la machine doit toujours tester la condition logique a et la condition logique b pour évaluer la condition logique (a && b) ?
- Est-ce que la machine teste ou non les deux conditions systématiquement ? Imaginez une procédure pour le tester.
- Soit une boucle for(int i=0;i<5;i++), on a des instructions, une condition, des instructions : for(instruction;condition;instruction). Imaginez un moyen d'afficher un message à chaque fois que la condition est testée.

Ex 7

Imaginez une fonction qui prend en entrée un tableau double[][] tab. tab est une matrice de jeu telle que tab[i][j] est le score du joueur 1 lorsque le joueur 1 joue i et le joueur 2 joue j. Affichez les équilibres de Nash de la matrice.

Exercice complet jeu

On veut créer un programme tel qu'à son exécution, il crée un objet Joueur, que la création de cet objet entraîne de demander à l'utilisateur son nom. Le programme poursuit en tirant un entier au hasard. Il demande ensuite à l'utilisateur de rentrer une valeur entière. Le programme indique si la valeur saisie est plus grande ou moins grande que la valeur tirée au hasard et lui demande de rentrer une nouvelle valeur il s'arrête si l'utilisateur a deviné la bonne solution.

- 1) créez une classe DevineNombre qui contient la fonction main, on ne met pas encore d'instruction pour cette fonction
- 2) Créez une classe Joueur telle que la création d'un Joueur entraîne de lui demander son nom et que la classe Joueur enregistre ce nom. On utilise la classe Lire : expliquer Lire.S
- 3) Notez les opérations que doit faire la fonction main sans vous préoccuper de comment les faire
- 4) Implémentez les opérations de la fonction main, en sachant que pour lire un entier, en sachant que pour tirer un entier entre 0 et n-1, il faut :
 - importer la classe Random : `import java.util.Random;`
 - instancier la classe Random : `Random alea=new Random();`
 - tirez un nombre au hasard dans [0:n-1] : `int i=alea.nextInt(n);`

En sachant également

- 5) Répertoriez les classes utilisées dans le programme.