



TP N° 1

Configuration de l'environnement de travail

Les TP de programmation orientée objet supposent que votre environnement de travail est configuré pour Java 7 ou Java 8. Sinon, Installez le **JDK** sur votre PC (Windows ou Linux). Pour vérifier les versions de votre JDK et de votre JRE, tapez les commandes suivantes dans un terminal :

javac -version ou java -version.

Pour installer le JDK sur un PC personnel, il vous faudra le télécharger depuis le site d'Oracle pour Windows ou installer le package OpenJDK pour un système Linux. Vous êtes libres d'utiliser votre outil préféré pour l'édition de votre code source. Néanmoins il est plus pratique d'utiliser un outil de développement intégré (IDE). L'IDE conseillé pour ce TP est le **blueJ**.

Organisation du travail

Vous enregistrerez les différents fichiers que vous créerez pour résoudre les exercices, dans un répertoire java/tp (ou poo/tp). L'emplacement exact du répertoire java est à votre convenance. Chaque TP aura son propre répertoire et, sauf exceptions, chaque exercice aura aussi son propre répertoire, ce qui vous permettra de conserver différentes versions d'une même classe. Vous aurez ainsi à créer, par exemple, un fichier **java/tp/tp1/exo1/Bonjour.java**. Si vous n'avez pas terminé un TP, vous devez le terminer seul (ou avec l'aide de vos camarades) pour le prochain TP. Vous pouvez demander de l'aide en demandant ou en envoyant un message électronique à un de vos enseignants si un point précis n'est pas trouvé.

Programme Bonjour !

1. Copiez dans un fichier Bonjour.java le code suivant :

```
public class Bonjour {  
  
    public static void main(String[ ] args) {  
        System.out.println("Bonjour tout le monde !");  
    }  
}
```

2. Depuis un terminal, compilez le fichier à l'aide de la commande :

>javac Bonjour.java

3. Vérifiez qu'un fichier Bonjour.class a été créé, puis exécuter cette classe à l'aide de la commande :

>java Bonjour

4. Refaire la même chose dans l'EDI blueJ

Exemples du cours

1. Recopiez et testez les exemples de code du cours des parties suivantes :

- *Cercle, Carre, Point2D, Vecteur2D et Demarrer*

2. Ajoutez des attributs et des méthodes tels que : couleur, Redimensionner, etc.



Exercice

```
class Moyenne {  
  
    double note;  
    int  nombreDeNotes ;  
  
    void ajouteNote(double note) { note += note; nombreDeNotes += 1; }  
    double calculMoyenne() { return note / nombreDeNotes; }  
}
```

1. Comment faire pour tester cette classe ?
2. Corrigez l'erreur afin qu'elle puisse donner une moyenne correcte.
3. Que se passe-t-il si on calcule la moyenne sans avoir saisi de notes ?
4. Pensez à écrire vos propres constructeurs dans la classe Moyenne ?
5. Créer une classe principale TestMoyenne afin de tester la classe moyenne ?

Gestion simple des entrées/sorties

On a vu que la méthode `System.out.print` permet d'afficher une chaîne de caractère à l'écran. En effet une méthode `System.out.printf` permet d'écrire une chaîne formatée sur la sortie standard, le comportement est analogue à la fonction `printf` du langage C. La classe `Scanner` est une classe utilitaire qui facilite la lecture sur l'entrée standard. Elle dispose de méthode de lecture pour tous les types primitifs et les chaînes de caractères. Cette classe est simple d'usage comme le montre l'exemple suivant :

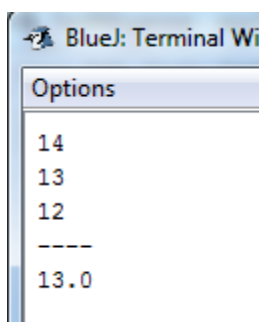
```
Scanner sc = new Scanner(System.in);  
double i = sc.nextDouble();
```

Pour rendre la classe accessible, il faut ajouter la ligne suivante en tout début de fichier.

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

A l'aide de la classe `Scanner`, Réécrivez la classe `TestMoyenne`. Cette classe dispose d'une méthode `main` qui demande à l'utilisateur de rentrer un certain nombre de notes puis affiche la moyenne des notes rentrées.

La classe `Scanner` contient une méthode `hasNextDouble` sans paramètre qui renvoie `true` si la prochaine entrée lue est bien un nombre réel et `false` sinon. Utilisez-la pour sortir d'une boucle `while` en saissant par exemple trois tirets "---".





Solutions TP1

Exo

```
class Moyenne {  
  
    double note;  
    int nombreDeNotes ;  
  
    void ajouteNote(double note) { this.note += note; nombreDeNotes += 1; }  
    double calculMoyenne() { return note / nombreDeNotes; }  
  
}
```

```
import java.util.Scanner;  
  
class TestMoyenne {  
public static void main (String[] args){  
    System.out.println ();  
    Moyenne m =new Moyenne();  
    Scanner sc = new Scanner(System.in);  
    double i;  
    System.out.print ("Saisir une note : " );  
    while (sc.hasNextDouble()) {  
        System.out.print ("Saisir une note : " );  
        i = sc.nextDouble();  
        m.ajouteNote(i);  
    }  
    System.out.println ("Moyenne : " + m.calculMoyenne());  
    }  
}
```