

La documentation de java est disponible en ligne sur le site de **Oracle** à l'url :
<http://docs.oracle.com/javase/7/docs/api/>
La page d'accueil de la documentation donne un certain nombre de pointeurs vers des documents utiles mais ce qui nous intéresse le plus ici est le lien appelé Java Platform API Specification, qui donne accès à la documentation de l'ensemble des classes existantes du jdk. Cette documentation a été générée automatiquement à l'aide de javadoc. Cet outil est capable d'exploiter un format spécial de commentaires dans des sources java, pour générer automatiquement une documentation au format html, consultable à l'aide de n'importe quel navigateur.

Exercice - 1 Prise en main de Eclipse

1. Lancer Eclipse.
2. Sélectionner un répertoire. Ce dossier sera votre répertoire de travail où seront stockés les sources (.java) et les fichiers compilés (.class) de tous vos programmes.
3. Créer un nouveau projet (File > New > Project). Pour cela, utilisez l'option New Project du menu File. Dans la boîte de dialogue qui apparaît, choisissez Java Project. Une fois créée, votre projet doit apparaître dans la vue Package située sur la gauche de la fenêtre. Cliquez sur la flèche pour le parcourir : il doit contenir la **JRE System Library**, ainsi qu'un répertoire de sources **src**.
4. Sélectionner le projet puis créer une nouvelle classe (clic droit, New > Class). Donner un nom à la classe : TestBonjour et vérifier que la case public static void main (String[] args) est bien cochée. Votre première classe est créée!
5. Exécution de programme :

```
public class TestBonjour{  
    // methode main()  
    public static void main(String[] args) {  
        System.out.println("Bonjour on va faire du Java sous Eclipse");  
    }  
}
```

- Pour tester votre programme, cliquez droit sur la classe dont vous voulez exécuter le main() et choisissez **Run As Java Application**.
- Pour fournir des arguments en entrée de votre programme, vous le pouvez en choisissant le mode **Run Configurations...** et régler dans l'onglet Java Application les arguments à fournir à votre programme.

Pour l'installation et la prise en main, on peut consulter l'url :
<http://henri.garreta.perso.luminy.univmed.fr/javaGen/eclipse/index.html>

Exercice - 2 : un peu de statistique

Ecrire un programme Stat, qui lit n notes (entiers dans l'intervalle [0,20]) et qui calcule la moyenne, le minimum , le maximum et l'écart type. les notes seront stockés dans un tableau d'entiers. Voici un exemple d'exécution :

```
Entrer le nombre de notes: 4  
Entrer la note 1: 10  
Entrer la note 2: 12  
Entrer la note 3: 8  
Entrer la note 4: 14  
La moyenne est : 11  
Le maximum est : 14  
Le minimum est : 8  
L'écart type est: 2.23606797749979
```

Pour le calcul de la moyenne et de l'écart type, on utilisera les formules suivantes :

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad \sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2 - \bar{x}^2}$$

```
public class Stat {
    public static int[] notes; //
    // methode main()
    public static void main(String[] args) {
        LireNotes();
        System.out.println("La moyenne est" + moyenne());
        System.out.println("Le minimum est " + min());
        System.out.println("Le maximum est " + max());
        System.out.println("L'ecart type est " + Ecart());
    }
}
```

Exercice - 3 Durée en secondes

On souhaite convertir une durée en seconde et inversement. Une durée est représenté par trois entiers :

- **h** : un entier qui représente les heures, $h \geq 0$,
- **m** : un entier qui représente les minutes, $0 \leq m \leq 59$,
- **s** : un entier qui représente les secondes, $0 \leq s \leq 59$.

Exemple :

2h 10mn 23s = $3600 \times 2 + 60 \times 10 + 23 = 7823$ secondes.

Ecrire la classe TestDuree qui fait la saisie d'une durée et qui la convertit en seconde et inversement. Elle contient en plus de la fonction main(), 2 méthodes public static (fonctions).

- la fonction **duree2seconde** qui convertit une durée en secondes.
- la fonction **Seconde2dureeHMS** qui convertit une durée en secondes en une durée sous forme de chaîne (String). Cette fonction retourne une chaîne de caractère contenant l'horaire. On suppose que la durée ≥ 0 .

Exercice - 4 Tri d'un tableau

Voici un programme qui fait la saisie d'un tableau d'entiers, qui le trie et qui l'affiche. Compléter ce programme en écrivant les fonctions Lire() afficher() et TriTableau(). Cette dernière utilise une fonction echange() qui échange deux éléments d'un tableau.

```
public class TestTriTableau {
    public static void main(String[] args) {
        int [] Tab ; int nb ;
        Scanner sc= new Scanner(System.in);
        System.out.println("Entrer la valeur de n");
        nb=sc.nextInt();
        Tab=Lire(nb);
        afficher(Tab);
        TriTableau(Tab);
        afficher(Tab);
    }
}
```

Exercice - 5 Calcul de π

Ecrire un programme qui calcule la valeur de π , qui utilise le développement en série suivant :

$$\pi \simeq 4 \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \frac{1}{13} \dots \right).$$

Vous devez décider sur le critère de terminaison utilisé dans le calcul (soit le nombre de termes utilisés ou bien l'amplitude d'un terme supplémentaire). Comparer la valeur de π obtenue avec la constante Math.PI de la librairie mathématique en pourcentage.

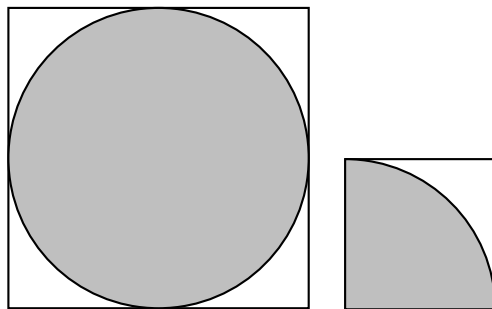
indication : additionner quand le dénominateur $d \% 4 = 1$ et soustraire si $d \% 4 = 3$.

Exercice - 6 Calcul de π et Monte Carlo

Les générateurs de nombres aléatoires sont au coeur de toutes les simulations de Monte Carlo. Pour illustrer l'application d'une méthode d'échantillonnage basée sur des nombres aléatoires dans le but de calculer une propriété macroscopique, prenons l'exemple du calcul de la valeur de π . La Figure représente un cercle unitaire inscrit sur un carré. On pourrait examiner ce problème en terme du cercle complet et du carré, mais il est plus facile de traiter le problème en considérant uniquement le quart, comme le montre la figure ci-dessous. Une personne tente de placer un grand nombre de fois un objet dans le quart du cercle. A chaque tentative, les coordonnées (x, y) de l'objet sont déterminées par deux nombres aléatoires indépendants choisis à partir d'une distribution uniforme entre 0 et 1. La distance r entre chaque objet et l'origine est calculée par la relation : $r = \sqrt{x^2 + y^2}$. Si cette distance est inférieure ou égale à 1, l'objet a atterri sur la zone hachurée et la tentative est comptée, sinon, l'essai est rejeté. Si on note n le nombre d'essais inscrits à l'intérieur du quart de cercle sur N tentatives exécutées, la valeur de π peut être calculée par l'expression :

$$\pi \simeq 4 * \frac{n}{N}$$

En effet si S désigne la surface du disque hachuré de rayon $r = 1$, alors $S = \pi * r^2 = \pi$. La surface du quart de cercle hachurée est donc $\frac{\pi}{4}$.



L'aire du quart de cercle de rayon 1 est égale à $\pi/4$

L'application d'une technique de Monte Carlo fait appel à un générateur de nombres aléatoires pour cet exemple consiste à la génération de N nombres aléatoires compris entre 0 et 1 à l'intérieur d'une distribution uniforme qui seront affectés au couple de coordonnées (x, y) .