Chapitre 6

Les sous-programmes

1. Fonctions Java prédéfinies

Le langage Java propose un ensemble de fonctions prédéfinies, mathématiques ou autres.

Par exemple :

sinus, cosinus, …

1.1 La librairie `Math`

Elle contient un ensemble de fonctions mathématiques prédéfinies.

Le nom de chaque fonction débute toujours par le terme **`Math`**, suivi d'un point, puis du nom de la fonction.

Quelques fonctions

On définit les variables :

```
double resultat, angle, a, b;
```

- Cosinus (en radian) d'un angle : **`Math.cos()`**
  Ex: `resultat = Math.cos(angle);`

- Sinus (en radian) d'un angle : **`Math.sin()`**

- Logarithme : **`Math.log()`**
  Ex: `resultat = Math.log(a);`

- Exponentielle : **`Math.exp()`**
  Ex: `resultat = Math.exp(a);`

- Racine carré d'un nombre : **`Math.sqrt()`**

  Ex: **`resultat = Math.sqrt(angle);`**

- a^b : **`Math.pow()`**

  Ex: **`resultat = Math.pow(a,b);`**

- |a| (valeur absolue de a) : **`Math.abs()`**

  Ex: **`resultat = Math.abs(a);`**

- Tirer un nombre au hasard entre 0 et 1 : **`Math.random()`**

  Ex: **`resultat = Math.random();`**

- Minimum de deux nombres : **`Math.min()`**

  Ex: **`resultat = Math.min(a,b);`**

- Maximum de deux nombres : **`Math.max()`**

2. Construire ses propres fonctions

Pour créer ses propres fonctions, il y a deux étapes :

- **Définir** préalablement sa fonction.
- **Appeler** sa fonction pour qu'elle soit exécutée.

```
public class Cercle
{
  // Fonction principale
  public static void main( String [] arg )
  {
    Scanner clavier = new Scanner(System.in);
    double valeur, résultat;
    System.out.print("Valeur du rayon : ");
    valeur = clavier.nextDouble();
    résultat = périmètre (valeur);
    System.out.println("Perimetre = " + résultat);
  } // fin de main()

  // Fonction périmètre
  public static double périmètre (double rayon)
  {
    double p;
    p = 2 * Math.PI * rayon;
    return p;
  } // fin de périmètre()
} //fin de class Cercle
```

2.3.2 Les différentes formes d'une fonction

- **Fonction avec résultat**

Voir fonction **`périmètre()`**. Le résultat était un double. Ce type doit être annoncé dans l'en-tête :

```
public static double périmètre (double rayon)
```

- **Fonction sans résultat**

Certaines fonctions ne renvoient pas de résultat.

```
public static void affiche (int valeur)
{
    System.out.print(valeur);
}
```

`void` signifie que la fonction **`affiche()`** ne retourne pas de résultat. Dans ce cas, il ne doit pas y avoir de `return` dans le corps de la fonction.

- Fonctions à plusieurs paramètres

Si nous écrivons par exemple une fonction **`max()`** qui permet de déterminer le plus grand de deux entiers, il y a deux paramètres, **`a`** et **`b`** de type entier :

```
public static int max (int a, int b)
{
  if( a > b )
    return a;
  else
    return b;
}
```

Les paramètres sont séparés par une **virgule**. Devant **chaque** paramètre, il faut indiquer son **type**.

- **Fonction sans paramètre**

Une fonction peut ne pas avoir de paramètre.

```
public static void affiche_bonjour()
{
     System.out.print("Bonjour") ;
}
```