Le procédural

Slide 3

Les méthodes

Une méthode permet d'écrire **un bloc d'instruction** particulier.

On fait appel à une méthode avec son nom afin d'exécuter ce bloc d'instruction.

Il n'y a pas de limite dans le nombre d'appels.

Une procédure

Une procédure

Une procédure est une méthode qui ne renvoie pas de valeur.
Elle fait simplement un traitement.

nomDeLaMéthode + (paramètres)

```
PROCEDURE maProcedure( )
  DEBUT_PROCEDURE
  AFFICHER “J’exécute ma procédure”
  FIN_PROCEDURE
```

Une procédure

Une procédure est une méthode qui ne renvoie pas de valeur.
Elle fait simplement un traitement.

nomDeLaMéthode + (paramètres)

```
PROCEDURE maProcedure( )
  DEBUT_PROCEDURE
  AFFICHER “J’exécute ma procédure”
  FIN_PROCEDURE
```

On l’appelle par son nom : maProcedure()

Une procédure

Déclaration d'une procédure avec paramètres :

```
PROCEDURE double( a )
  DEBUT_PROCEDURE
  AFFICHER a * 2
  FIN_PROCEDURE
```

Utilisation de ma procédure double avec une valeur passée en paramètre :

```
double(12)
```

Affiche : 24

Une fonction

Une fonction

Une fonction renvoie une valeur d'un certain type.

Présence obligatoire de **RENVOYER** en fin de fonction.

```
FONCTION maFonction()
  DEBUT_FONCTION
  RENVOYER …
  FIN_FONCTION
```

Une fonction

Déclaration de ma fonction double :

```
FONCTION double( a )
  DEBUT_FONCTION
  RENVOYER a * 2
  FIN_FONCTION
```

Utilisation de ma fonction double :

```
DEBUT_ALGORITHME
  result PREND_LA_VALEUR double(12)
  AFFICHER result
FIN_ALGORITHME
```

En java

Une procédure :

```
public static void maProcedure() {
    // …
}
```

Une fonction qui retourne un entier :

```
public static int maFonction() {
    // …
    return 4;
}
```

En java

Une procédure qui affiche le double du paramètre :

```
public static void maProcedureDouble(int value) {
    int result = value * 2;
    System.out.println(result);
}
```

Une fonction qui retourne le double du paramètre :

```
public static int maFonctionDouble(int value) {
    int result = value * 2;
    return result;
}
```

La Récursivité

La récursivité

Une fonction récursive est **une fonction qui s'appelle elle-même.**

Elle doit impérativement posséder **un cas d'arrêt.**

Exemple :

```
if (n == 0)
    return 1;
```

La récursivité

Exemple d'une méthode récursive qui affiche bonjour n fois :

```
PROCEDURE afficheBonjour( n )
  DEBUT_PROCEDURE
  SI (n > 0) ALORS
  DEBUT_SI
     AFFICHER "Bonjour"
     afficheBonjour( n - 1 )
  FIN_SI
  FIN_PROCEDURE
```

```
DEBUT_ALGORITHME
  afficheBonjour(4)
FIN_ALGORITHME
```

La récursivité

La récursivité n’est en aucun cas une mauvaise pratique.

La récursivité

La récursivité n’est en aucun cas une mauvaise pratique.

A partir du moment où on l’utilise correctement.

C’est à dire que l’on traite **le gros problème** en rappelant le même algorithme de manière récursive sur une sous partie afin de gérer **ce plus petit problème de la même manière**.