Àmbit de les variables

Al cos d'una funció es poden declarar variables, que s'anomenen **variables locals** . Els paràmetres d'una funció es poden considerar com a variables locals.

Les variables locals només es poden utilitzar en la funció on què es declaren, per el que el seu àmbit (scope), és aquesta funció. --> És a dir, les variables pròpies (els paràmetres de la capçalera i les variables creades dins de la funció) no és possible utilitzar-les fora de la funció on s'han declarat.

Podem dir que quan una funció s'executa, es creen les seues variables, s'utilitzen i quan la funció acaba es destrueixen les variables.

De moment, a les funcions només utilitzarem les variables d'àmbit local.

Però, en una classe, **es poden declarar variables globals en qualsevol punt fora del cos d'una funció dins del class**.

Per crear aquestes variables globals farem servir la directiva 'static'. Més endavant, en OO, ja veurem en detall el significat real del static.

Els mètodes d'una classe (el que ara nomenem funcions) poden utilitzar les variables globals.

Una variable local pot tindre el mateix nom que una variable global. En aquest cas la variable local sempre sobreescriu a la variable global. Exemple:

```
public class Ejemplo {
  static int x;  //es una variable global
  static int cubo (int z)
  {
    int x;       //variable local del método cubo
    x = z*z*z;   //x es la variable local, no la global
    return x;
  }
  public static void main (String args[])
  {
    int p;       //variable local del método main
    p = 10;      //correcto: p es local en este ámbito
    z = 100;     //error: z no está en este ámbito
    x = cubo(p); //correcto: x es global, x vale 1000
  }
}
```

Altre exemple:

```
public class Ejemplo {
  static int num = 5;  //es una variable global
  static int cubo (int num)
  {
    int x = num*num*num;       //num es la variable local del método cubo
    return x;
  }
  public static void main (String args[])
  {
    num = cubo(num-1);  // se utiliza la variable global → en num se guarda 64
    System.out.println(num);   // se imprime 64(variable local)
  }
}
```

En canvi en aquest altre exemple, no es sobreescriu la variable global

```
public class Ejemplo {
  static int num = 5;  //es una variable global
  static int cubo (int num)
  {
    int x = num*num*num;       //num es la variable local del método cubo
    return x;
  }
  public static void main (String args[])
  {
    int num = 10;      // es una variable local, sobrescribe la global
    num = cubo(num);  // se utiliza la variable local, en num se guarda 1000
    System.out.println(num);   // se imprime 1000(variable local)
  }
}
```