

Programación Modular

Programación modular

- La programación modular surge de la evolución de la programación estructurada
- Consiste en crear módulos o partes del programa reutilizables, que se unirán para resolver el problema original
- Se basa en la técnica “divide y vencerás”
- Evita que en los programas exista código repetido

Programación modular en Java

- En Java la programación modular se base en el concepto de método (por ahora serán estáticos)
- Un método es un conjunto de instrucciones de programa que realizan una determinada tarea. Tienen una serie de datos de entrada y pueden devolver un valor de retorno
- Por ahora trabajaremos con métodos serán siempre privados y estáticos (private static....)
- Los métodos se podrán “invocar” o “llamar” desde el main o desde otro método

Método

Un método tiene

- Una lista de parámetros de entrada al método (datos que necesita el método para hacer la tarea específica)
- Un valor de retorno (valor que devuelve el método)
- Puede tener también variables locales al método



Método

- Un método puede tener de 0 a n parámetros de entrada. Los parámetros (formales) tienen
 - Un nombre
 - Un tipo
 - Un valor (que ya le viene dado)
- Un método puede o no tener un valor devuelto
- El método también puede tener variables locales a él.

Método

Sintaxis de un método estático en Java

```
private static TipoDevuelto nombreMetodo  
(listaDeParametros)  
{  
    // Variables locales al metodo  
    // código del método  
    // sentencia return  
}
```

Método

- **TipoDevuelto** es el tipo de dato que devuelve el método (void si no devuelve nada)
- **ListaDeParametros**
(tipo param1, tipo param2, tipo param3)
- La sentencia **return** aparecerá siempre salvo que el método no devuelva nada (sea void)
- Las variables locales sólo son visibles desde₁ el código del método.

Llamada a un método

- La llamada a un método estático que no devuelve nada se realiza con
`NombreClase.nombreMetodo(parametrosReales)`
- La llamada a un método estático que devuelve un valor se realiza con
`variable=NombreClase.nombreMetodo(parametrosReales)`
`//La variable debe ser del tipo devuelto por el método`
- En los parámetros reales no hay que poner el tipo, solo el nombre. Van separados por comas y deben coincidir en tipo y número con los parámetros formales

Llamada a un método

Cuando desde un programa P se hace una llamada o se “invoca” un método M ¿Que ocurre?

1. Se “traspasan” los parámetros reales (los de P) en los parámetros formales (los de M)
2. A continuación se ejecuta el código del método M.
3. El método M termina la última instrucción y devuelve el valor de retorno (opcional) , volviendo el control al programa P
4. El programa P continúa por la siguiente instrucción a la llamada al método.

Llamada a un método

- Si los parámetros son de tipos básicos el paso de parámetros se hace “por valor” (también llamado “por copia”)
- Si los parámetros son objetos (ya lo veremos) el paso de parámetros se hace por referencia.
- Los parámetros reales y los formales deben coincidir en número y en tipo.