

Tema 5: Métodos

- Declaración
- Llamada
- Tipos de parámetros
 - Valor
 - Referencia
- Variables de instancia
- Tipos de acceso

09/10/2017

1

Método

- Valor de retorno: valor devuelto por el método como resultado de su llamada. Es único.
- Parámetros: Variables adicionales que necesita el método para poder realizar las instrucciones en él indicadas.

09/10/2017

3

Método

- **Método:** Un método es un bloque de sentencias que se puede ejecutar repetidas veces invocando su nombre. Los métodos pueden tener cero, uno o varios parámetros y devolver o no un resultado de cierto tipo.
 - Cuerpo: Contiene las instrucciones que se ejecutan cuando se realiza la llamada al método.
 - Nombre: Identificador para la nomenclatura del método.

09/10/2017

2

Sintaxis de declaración de un método

```
<acceso> <tipoDevuelto> <nombreMétodo>(<lista de parámetros>)
{
  <cuerpo>
}
```

Dónde:

- acceso: indica si el método se puede ejecutar desde otra clase distinta, y puede ser public, private, protected, o el tipo por defecto, si no se indica nada (private).
- tipoDevuelto: Indica el tipo de datos que devuelve el método. Puede ser cualquier tipo de datos primitivo: int, float, double, etc o no básico (Clase). Si el método no devuelve valor, se indica mediante la palabra reservada **void**.

09/10/2017

4

Sintaxis de declaración de método

- nombreMetodo: identificador del método. Debe seguir las reglas de los identificadores y ser suficientemente descriptivos de la acción que realizan.
- listaParámetros: lista con el formato: tipoDato para1, tipo Dato para2,..... Donde se especifica el tipo de dato y el nombre para cada uno de los parámetros del método. Consideraciones:
 - Para más de un parámetro: Se separan por ,
 - No existen parámetros: Es obligatorio indicar ()
- cuerpo: contiene las instrucciones que realiza el método.

09/10/2017

5

Llamada a métodos

<objeto>.<nombreMétodo>(<listaDeArgumentos>)

- objeto: indica la variable de tipo clase en la que se encuentra el método al que se quiere acceder.
- nombreMétodo: identificador del método que se quiere acceder.
- listaDeArgumentos: Lista de parámetros necesarios para el método.

09/10/2017

7

Valor de retorno

- Para conseguir que un método devuelva un valor indicado, es necesario incluir la sentencia:


```
return <expresión>;
```
- El tipo de datos devuelto por la expresión y el tipo de datos especificado como valor de retorno del método han de ser iguales
- Método de tipo void:
 - Incluir la palabra return; sin ninguna expresión.
 - Omitir la palabra return y que el método finalice al llegar a la llave de final de método.

09/10/2017

6

Tipos de parámetros

- Todos los parámetros de los métodos se pasan por valor. Cuando se realiza la llamada a un método, los parámetros formales (parámetros indicados en la declaración) reservan un espacio en memoria independiente y reciben los valores de los parámetros reales
 - Valores primitivos: Se copia un valor del parámetro y su valor no se ve afectado en el método.

```
public void sumar(int n1, int n2, ref int res)
{
    res = n1 + n2;
}
```

09/10/2017

8

Tipos de parámetros

- Valores de tipo referencia: Su valor se puede modificar dentro del método. Se emplea la palabra.

```
public static void sumar(int n1, Persona p)
{
    n1 = 0;
    p.CambiarEdad(16);
}
```

09/10/2017

9

Tipos de acceso

Modificador	La misma clase	Mismo paquete	Subclase	Otro paquete
private	Sí	No	No	No
default	Sí	Sí	No	No
protected	Sí	Sí	Sí/No	No
public	Sí	Sí	Sí	Sí

09/10/2017

11

Variable de instancia

- Dentro de un método de una clase, se puede acceder a los datos del objeto actual. Este tipo de acceso se llama al valor implícito del objeto y se utiliza la palabra reservada *this*.
- Dentro de un método de una clase, se puede acceder a los datos del objeto padre, siempre que haya herencia. Este tipo de acceso se realiza con la palabra reservada *super*.

09/10/2017

10