

Introducción a Java

Constantes

Si una variable se declara constante, es de tipo primitivo o es una cadena y se conoce su valor, el compilador sustituye el nombre de la variable por el valor de esta al compilar

Constructores

Se usan para crear nuevos objetos

Es una buena práctica asignar un valor inicial a los atributos para asegurar que el objeto tiene un estado válido, esto se hace dentro del constructor

El nombre del constructor debe ser igual al nombre de la clase

A diferencia de un método, un constructor no especifica el tipo del valor de retorno

Declaración de un constructor

```
Ejemplo(T prm1, T prm2, ..., T prmN){  
    //Secuencia de instrucciones  
}
```

`Ejemplo` es el nombre de la clase

`T prm1, T prm2, ..., T prmN`
es una lista de cero o más variables
llamadas parámetros

Constructores

No es obligatorio declarar constructores para una clase

Si no se declara un constructor, el compilador automáticamente creara uno sin parámetros

Invocando constructores

Para crear un nuevo objeto se invoca a un constructor con el operador **new**

```
new Ejemplo(prm1, prm2, ..., prmN)
```

```
param1, param2, ..., paramN
```

Es una lista de cero o más variables o valores, se les llama argumentos

El número y orden de los tipos tiene que coincidir con alguna declaración de constructor de la clase

Crear un nuevo objeto

El nuevo objeto puede ser asignado a una variable

```
Ejemplo ejemplo = new Ejemplo(...)
```

Operador . (punto)

El operador punto se usa para acceder o modificar el valor de un atributo de una instancia:

```
objeto.atributo
```

```
objeto.atributo = literal
```

```
objeto.atributo = variable
```

```
objeto.atributo = otroObjeto.atributo
```


Métodos

Un **método** es una secuencia de instrucciones usada para modelar el comportamiento del objeto

Tienen la capacidad de:

Modificar el estado del objeto

Regresar información (valor de retorno)

Recibir nueva información (parámetros)

Declaración de métodos

```
T metodo(T prm1, T prm2, ..., T prmN){  
    //Secuencia de instrucciones  
}
```

T es el tipo del valor de retorno

metodo es el nombre del método

T prm1, T prm2, ..., T prmN
es una lista de cero o más parámetros

Nombrado de métodos

[JCC] Por convención los nombres de métodos deben ser verbos escritos en minúsculas o un verbo escrito en minúscula seguido de sustantivos o adjetivos sin espacios con la primer letra escrita en mayúscula

`run()`

`isEmpty()`

`runApp()`

`getAllNames()`

Parámetros

En la declaración de un método o un constructor se declara el número y tipo de parámetros

```
T metodo(T prm1, T prm2, ...T prm1){  
    ...  
}
```

Los parámetros son usados en el cuerpo del método y serán reemplazados por valores en tiempo de ejecución

Parámetros vs argumentos

Los parámetros se refiere a las variables de la declaración de un método

```
void metodo(int m1, byte m2) {  
    ...  
}
```

Los argumentos son los valores que se le dan a un método cuando este es invocado

```
objeto.metodo(2, 0b010);
```

Valor de retorno

Para retornar un valor se usa la sentencia
`return`

El método terminará su ejecución y regresará el valor a la expresión donde fue invocado

```
T metodo(T prm1, ..., T prmN,){  
    ...  
    return expresion;  
}
```

Valor de retorno

Si el método no retorna nignun valor, el tipo de retorno en la declaración del método debe ser **void**

```
void metodo(T param1, ...){  
    ...  
}
```

Invocando métodos

Un método está asociado a una clase de objetos, para invocar el método se usa el operador . (punto)

```
objeto.ejemplo(prm1, prm2, prmN)
```

```
objeto.ejemplo()
```


Métodos de clase

No es necesario instanciar un objeto para hacer uso del método

Se declaran con el modificador `static`

```
static T metodo(T param1, ...) {  
    ...  
}
```

Dentro del método, no se pueden usar variables de instancia

Invocar a un método de clase

```
Clase.metodo(param1, param2, ... paramN);
```

```
objeto.metodo(param1, param2, ... paramN);
```

Métodos de acceso

Se usan para conocer el valor de los atributos del objeto.

Por convención el nombre del método comienza con **get**

Métodos modificadores

Se usan para modificar el valor de los atributos del objeto

Por convención el nombre del método comienza con **set**