Sintaxis básica de Java

Pedro O. Pérez M., PhD.

Desarrollo de videojuegos con Java
Tecnológico de Monterrey

pperezm@tec.mx

10-2023

Clases

Recordemos que...

- **Nombre del archivo:** El nombre de un archivo fuente debe coincidir exactamente con el nombre de la clase pública con la extensión `.java`. El nombre del archivo puede ser diferente si no tiene ninguna clase pública. Por ejemplo:
- **Nombre de la clase:**
  - La primera letra de la clase debe estar en mayúscula.
  - Si se utilizan varias palabras para formar el nombre de la clase, la primera letra de cada palabra interna debe estar en mayúscula.
- El método `main()` es el principal punto de entrada a un programa Java.

```
public class Example1 {
        public static void main(String args[]) {
                System.out.println("Hello world!!!");
        }
}
```

Métodos

- Todos los nombres de los métodos deben comenzar con una letra minúscula. Si se utilizan varias palabras para formar el nombre del método, entonces cada primera letra de la palabra interna debe estar en mayúscula. Se permiten guiones bajos, pero no se recomiendan.

Identificadores

- Todos los identificadores pueden comenzar con una letra, un símbolo de moneda o un guión bajo (_). Según la convención, una letra debe estar en minúscula para las variables
- El primer carácter de los identificadores puede ir seguido de cualquier combinación de letras, dígitos, símbolos de moneda y guión bajo. No se recomienda el guión bajo para los nombres de variables. Las constantes deben estar en letras mayúsculas.
- Lo más importante es que los identificadores distinguen entre mayúsculas y minúsculas.
- Una palabra reservada no se puede utilizar como identificador.

Modificadores de acceso

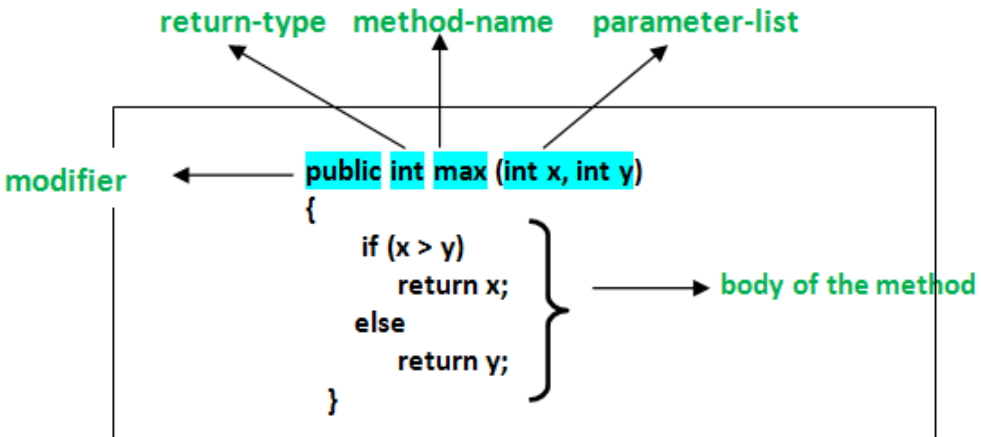
- Estos modificadores controlan el alcance de las clases y los métodos.
  - **Modificadores de acceso:** default, `public`, `protected`, `private`.
  - **Modificadores "sin acceso":** `final`, `abstract`, `static`.

Modificador de acceso	Dentro de la clase	Dentro del paquete	Paquete externo solo por subclase	Paquete exterior
`private`	*Si*	*No*	*No*	*No*
Default	*Si*	*Si*	*No*	*No*
`protected`	*Si*	*Si*	*Si*	*No*
`public`	*Si*	*Si*	*Si*	*Si*

Estructura de una clase en Java

- En general, al declarar una clase en Java se pueden incluir estos componentes en orden:
  - **Modificadores:** Una clase puede ser pública o tener acceso predeterminado.
  - **Nombre de clase:** El nombre de la clase debe comenzar, por convención, con la letra inicial en mayúscula.
  - **Superclase (si existe):** El nombre la superclase, si existe, precedido por la palabra reservada `extends`. Una clase solo puede extender de otra.
  - **Interfaces (si las hay):** Una lista separada por comas de interfaces implementadas por la clase, si las hay, precedidas por la palabra clave `implements`. Una clase puede implementar más de una interfaz.
  - **Cuerpo:** El cuerpo de la clase está rodeado por llaves `{, }`.

Métodos


return-type
method-name
parameter-list
modifier
public int max (int x, int y)
{
if (x > y)
return x;
else
return y;
}
body of the method

En Java,

- Los argumentos se pasan a los métodos por valor; se hace una copia del tipo de dato primitivo, como `int`, `float`, etcétera, o la referencia al objeto de la clase o un arreglo, y se pasa al método.
- Es posible sobrecargar métodos, es decir, definir dos o más dentro de la misma clase, que compartan nombre y que las declaraciones de sus parámetros sean diferentes; la sobrecarga es una forma de polimorfismo.
- Los métodos abstractos se definen en clases para imponer funcionalidad a las subclases; una clase con un método abstracto es abstracta; estos métodos se definen por la palabra reservada `abstract` y no tienen cuerpo; su implementación se deja a las subclase.

Una clase en java

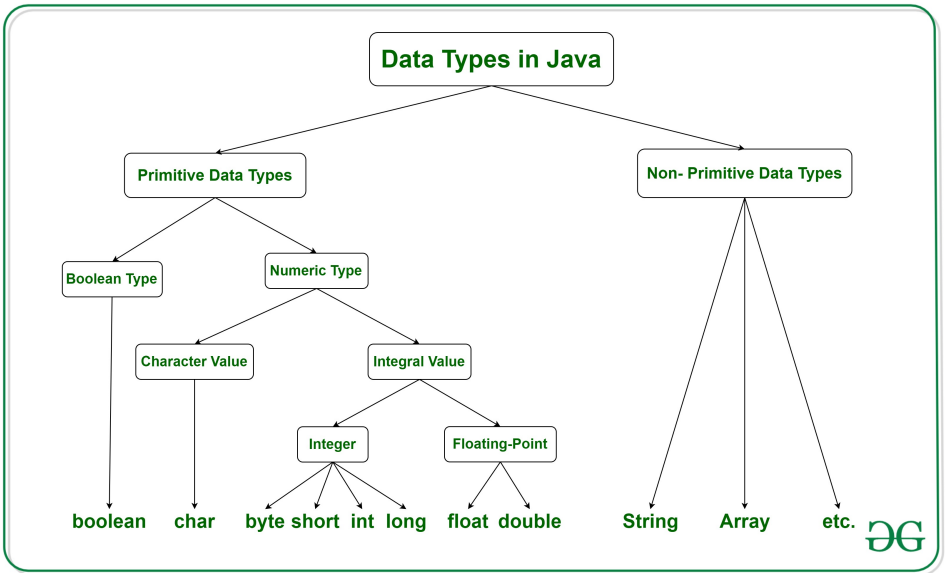
Listing 1: Una clase en Java

```
public class Lamp {
        // stores the value for light
        // true if light is on, false if light is off
        private boolean isOn;

        // method to turn on the light
        public void turnOn() {
                isOn = true;
                System.out.println("Light on? " + isOn);
        }

        // method to turnoff the light
        public void turnOff() {
                isOn = false;
                System.out.println("Light on? " + isOn);
        }
}
```

Tipos de datos en Java



Tipos de datos enteros

Nombre	Bytes	Rango
byte	1	–128 a 127
short	2	-32,768 a 32,767
int	4	–2,147,483,648 a 2,147,483,647
long	8	–9,223,372,036,854,775,808 a 9,223,372,036,854,775,807

Tipos de datos flotantes

Nombre	Bytes	Rango
float	4	1.4e–045 a 3.4e+038
double	8	4.9e–324 a 1.8e+308

Tipo de datos `char` y `boolean`

- **boolean:** El tipo de datos `boolean` representa solo un bit de información, ya sea verdadero o falso. El tamaño del tipo de datos booleano depende de la máquina virtual.
- **char:** El tipo de datos `char` es un único carácter Unicode de 16 bits con un tamaño de 2 bytes (16 bits).

Tipos de datos no primitivos o de referencia

- Los tipos de datos de referencia contendrán una dirección de memoria de valores de variables porque los tipos de referencia no almacenarán el valor de la variable directamente en la memoria.
  - Los objetos Strings se definen como un arreglo de caracteres.
  - Un arreglo es un grupo de variables del mismo tipo a las que se hace referencia con un nombre común.
    - Dado que los arreglos son objetos en Java, podemos encontrar su longitud utilizando la propiedad `length` del objeto.

Estructuras de control

- **Condicionales:** `if`, `if-else`, `if-else` anidados, `switch`.
- **Ciclos:** `for`, `while`, `do-while`.
- **Operadores matemáticos:** +, -, *, /, %, ++, -.
- **Operadores relaciones:** <, <=, >, >=, ==, !=.
- **Operadores lógicos:** && (AND), || (OR), ! (NOT).

Examen Rápido 2
(Sintaxis básica de Java)