

Universidad Francisco Marroquín

Facultad de Ciencias Económicas

Ingeniería en Computer Science

Profesor: Etson Guerrero

Auxiliar: Yosef Maldonado

Estructura de Métodos en Java

Sophia Gamarro

Carlos Alvarado

Cruz del Cid

Mario Psiquiy

Javier Mazariegos

Métodos

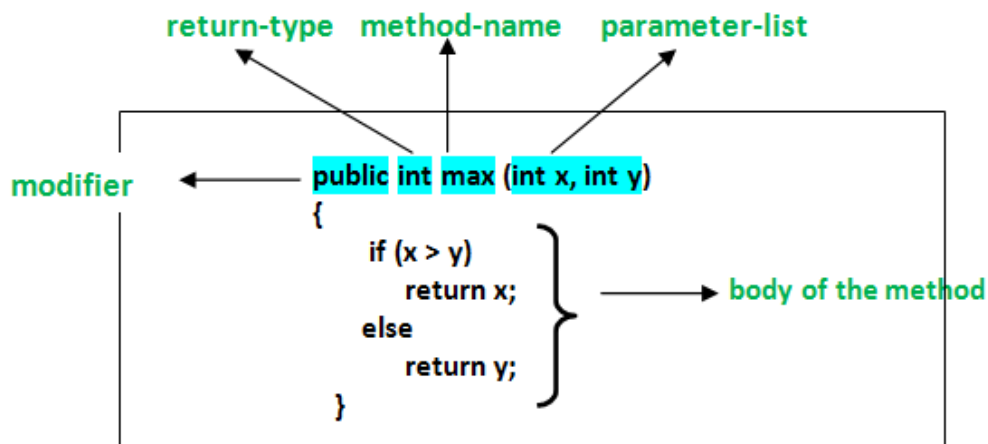
Un método en Java es un conjunto de instrucciones definidas dentro de una clase que realizan una determinada tarea, la cual pueden ser invocada mediante un nombre. Los métodos permiten reutilizar el código sin volver a escribirlo por lo que son considerados ahorradores de tiempo. En Java, cada método debe ser parte de alguna clase, a diferencia de otros lenguajes como C, C++ y Python.

Generalmente, la declaración de un método consiste en seis componentes:

1. Especificadores: Define el tipo de acceso del método, es decir, desde donde se puede acceder a él en su aplicación. En Java, hay 4 tipos de especificadores de acceso.
 - Public: accesible en todas las clases en la aplicación.
 - Protected: accesible dentro de la clase en la que está definido y en su (s) subclase (s)
 - Private: accesible solo dentro de la clase en la que está definido.
 - Default (declarado / definido sin usar ningún modificador): accesible dentro de la misma clase y paquete dentro del cual su clase está definida.
2. Tipo de valor devuelto: tipo de datos del valor devuelto por el método o 'void' si no devuelve un valor.
3. Nombre del método: nombre único que se utiliza para definir el nombre de un método. Debe corresponder a la funcionalidad del método. Es decir, si se está creando un método para restar dos números, el nombre del método debe ser resta(). Un método se invoca por su nombre.
4. Lista de parámetros: Se define una lista separada por comas de los parámetros (también llamados 'argumentos') de entrada, precedida por su tipo de datos, dentro del paréntesis cerrado. Si no hay parámetros, debe usar paréntesis vacíos ().

5. Lista de excepciones: indica las excepciones que puede generar y manipular el método.
6. Cuerpo del método: Es parte de la declaración del método. Contiene todas las acciones a realizar. Está encerrado dentro de un par de llaves

Los componentes mencionados pueden observarse en la siguiente imagen.



Variables

Una variable proporciona un almacenamiento con nombre que los programas pueden manipular. Cada variable en Java tiene un tipo específico, que determina el tamaño y el diseño de la memoria de la variable, el rango de valores que se pueden almacenar dentro de esa memoria, y el conjunto de operaciones que se pueden aplicar a la variable.

En Java existen cuatro tipos de variables:

1. Variables de instancia (campos no estáticos): variables que están definidas dentro de un objeto. Los objetos mantienen su estado interno en campos no estáticos. Los campos no estáticos también se denominan variables de instancia, porque pertenecen

a instancias (objetos) de una clase. Las variables de instancia suelen llevar un modificador de visibilidad (public, private, protected) definiéndose.

2. Variables de clase (campos estáticos): son aquellas variables que están precedidas del modificador 'static'. Esto indica que solo hay una instancia de dicha variable. Es decir, aunque se tengan N objetos de la clase, la variable estática solo se instancia una vez.
3. Variables locales: variables temporales cuyo ámbito de visibilidad es el método sobre el que están definidas. No pueden ser accedidas desde otra parte del código. Se les distingue de las variables de instancia ya que estas no llevan modificadores de visibilidad delante.
4. Parámetros: son las variables recibidas como parámetros de los métodos.

Declaración de variables en Java

Para crear una variable, es necesario especificar el tipo de dato y asignarle un valor:

tipo variable = valor;

Donde 'tipo' es uno de los tipos de datos de Java (como int o String) y 'variable' es el nombre de la variable (como x o name). El signo igual se utiliza para asignar valores a la variable.

Los tipos de datos se dividen en dos grupos:

1. Tipos de datos primitivos: incluye byte, short, int, long, float, double, boolean y char
2. Tipos de datos no primitivos, como Strings, Arrays y Clases.

Bibliografía

Tutorials Point. Java-Methods. Recuperado de
https://www.tutorialspoint.com/java/java_methods.htm

ProgramarYa. Métodos en Java, funciones y procedimientos. Recuperado de
[https://www.programarya.com/Cursos/Java/Funciones#:~:text=Crear%20un%20m%C3%A9todo%20en%20Java&text=EL%20primer%20componente%20corresponde%20a,ambas\)%2C%20tambi%C3%A9n%20es%20opcional.](https://www.programarya.com/Cursos/Java/Funciones#:~:text=Crear%20un%20m%C3%A9todo%20en%20Java&text=EL%20primer%20componente%20corresponde%20a,ambas)%2C%20tambi%C3%A9n%20es%20opcional.)

TutorialsPoint. Java-Variables. Recuperado de
https://www.tutorialspoint.com/java/java_variable_types.htm