

Conceptos fundamentales de Java

7-2: Parámetros y métodos de sobrecarga

Actividades practices

Objetivos de la lección:

- Uso de modificadores de acceso
- Pasar objetos a métodos
- Devolver objetos a partir de métodos
- Uso de métodos con argumentos variables
- Sobrecarga de constructores
- Métodos de sobrecarga
- Escritura de una clase con arreglos, constructores y métodos especificados

Vocabulario:

Identifique el término correspondiente a cada una de las definiciones a continuación.

	Tipo de modificador de acceso. Permite el acceso desde cualquier lugar.
	Se utiliza para asignar valores iniciales a variables de instancias de una clase.
	Forma de invocar un método con una cantidad variable de argumentos.
	Existencia de más de un constructor o método con el mismo nombre pero diferentes argumentos.
	Tipo de modificador de acceso. Permite el acceso sólo desde dentro de la misma clase.
	Un constructor que no tiene parámetros.
	Utilizado para especificar la accesibilidad a variables, métodos y clases.

Inténtelo/resuélvalo:

1. Cree un segmento del código que inicializa una clase pública denominada Fish. Deje a la clase contener una cadena `typeOfFish`, y un número entero para `friendliness`. No establezca valores para esas variables aún. Son variables de instancias y se colocarán dentro de la clase constructores.
2. Cree un constructor público (un método con el mismo nombre que la clase) dentro de la clase Fish (pescado). Este constructor no debería incorporar argumentos. En el constructor, configure `typeOfFish` en "Unknown" y `friendliness` en 3, que asumimos que es el término genérico amistad hacia el pescado.
3. Cree otro constructor público dentro de la clase Fish. Haga a este constructor adoptar una cadena `t` y un número entero `f`. Let `typeOfFish` igual al `t`, y `friendliness` igual a `f`.

4. Explique por qué es posible tener más de un constructor con el mismo nombre y diferentes argumentos.
5. Cree un método dentro de la clase Fish denominado getFriendliness que no incorpora argumentos y devuelve el nivel de amistad del pescado.
6. Escriba un segmento de código que inicialice 2 pescados nuevos según se define a continuación:
 - a. Pescado 1: Nombre – Amber, Tipo – AngelFish (Pez Ángel), Nivel de amistad – 5 (muy amigable)
 - b. Pescado 2: Nombre– James, Tipo – Guppy (Olomina), Nivel de amistad – 3 (neutral)
7. Cree el método nicestFish (el pescado más lindo) que incorpora a dos pescados como parámetros, compara el nivel de amistad de los dos pescados y devuelve el pescado con la mayor amistad. Pruebe este método con el pescado definido en el problema 6. (Escala de amistad: 1 malo, 2 no amigable, 3 neutral, 4 amigable, 5 muy amigable) Consejo: fishName.getFriendliness() le proporciona el número entero del nivel de amistad defishName. Ya ha creado el valor getFriendliness en el problema 5.
8. Modifique el método nicestFish para obtener un método de argumento variable que adopte una cantidad variable de peces y devuelva al pescado más lindo de los pescados proporcionados. Consejo: dentro del método, cree un nuevo pescado denominado temp. Configure el valor temp igual al primer pescado incluido en el método. Utilice un ciclo para ver por todos los pescados introducidos al método y si descubre un pescado más amigable que temp, configure temp igual a ese pescado. Después de finalizar el ciclo, temp debería ser el pescado más amigable. Devolver temp.
9. Pruebe el método nicestFish on los pescados descritos en el problema 6. ¿Qué pescado se devuelve?
10. Determine el mejor modificador de acceso para cada una de las siguientes situaciones:
 - a. La clase Employee (Empleado) registra el nombre, la dirección, el salario y el número de teléfono.
 - b. Un método de agregado dentro de una clase BasicMath que también se utiliza en la clase Álgebra.