APRENDERAPROGRAMAR.COM

PALABRAS CLAVE STATIC Y FINAL EN JAVA. VARIABLES DE CLASE O CAMPOS ESTÁTICOS Y CONSTANTES. EJEMPLOS. (CU00673B)

Sección: Cursos
Categoría: Curso “Aprender programación Java desde cero”

Fecha revisión: 2029

Resumen: Entrega nº73 curso Aprender programación Java desde cero.

Autor: Alex Rodríguez

PALABRAS CLAVE STATIC Y FINAL. CONSTANTES EN JAVA.

En los programas que generemos usualmente intervendrán constantes: valores matemáticos como el número Pi, o valores propios de programa que nunca cambian. Si nunca cambian, lo adecuado será declararlos como constantes en lugar de cómo variables. Supongamos que queremos usar una constante como el número Pi y que usamos esta declaración:


```
// Ejemplo aprenderaprogramar.com
public class Calculadora {
    private double PI = 3.1416;
    public void mostrarConstantePi () { System.out.println (PI); }
    … constructor, métodos, … código de la clase …   }
```

Si creas un objeto de tipo Calculadora, comprobarás que puedes invocar el método mostrarConstantePi para que te muestre el valor por pantalla. No obstante, una declaración de este tipo presenta varios problemas. En primer lugar, lo declarado no funciona realmente como constante, sino como variable con un valor inicial. Prueba a establecer *this.PI = 22;* dentro del método y verás que es posible, porque lo declarado es una variable, no una constante. En segundo lugar, cada vez que creamos un objeto de tipo calculadora estamos usando un espacio de memoria para almacenar el valor 3.1416. Así, si tuviéramos diez objetos calculadora, tendríamos diez espacios de memoria ocupados con la misma información, lo cual resulta ineficiente. Para resolver estos problemas, podemos declarar constantes en Java usando esta sintaxis:

```
CaracterPublico/Privado static final TipoDeLaConstante = valorDeLaConstante;
```

En esta declaración intervienen dos palabras clave cuyo significado es importante:

a) **static:** los atributos miembros de una clase pueden ser atributos de clase o atributos de instancia; se dice que son atributos de clase si se usa la palabra clave *static*: en ese caso la variable es única para todas las instancias (objetos) de la clase (ocupa un único lugar en memoria). A veces a las variables de clase se les llama variables estáticas. Si no se usa *static*, el sistema crea un lugar nuevo para esa variable con cada instancia (la variable es diferente para cada objeto). En el caso de una constante no tiene sentido crear un nuevo lugar de memoria por cada objeto de una clase que se cree. Por ello es adecuado el uso de la palabra clave *static*. Cuando usamos “static final” se dice que creamos una constante de clase, un atributo común a todos los objetos de esa clase.

b) **final:** en este contexto indica que una variable es de tipo constante: no admitirá cambios después de su declaración y asignación de valor. *final* determina que un atributo no puede ser sobreescrito o redefinido. O sea: no funcionará como una variable "tradicional", sino como una constante. Toda constante declarada con *final* ha de ser inicializada en el mismo momento de declararla. *final* también se usa como palabra clave en otro contexto: una clase final (*final*) es aquella que no puede tener clases que la hereden. Lo veremos más adelante cuando hablemos sobre herencia.

Cuando se declaran constantes es muy frecuente que los programadores usen letras mayúsculas (como práctica habitual que permite una mayor claridad en el código), aunque no es obligatorio.

Una declaración en cabecera de una clase como *private final double PI = 3.1416;* podríamos interpretarla como una constante de objeto. Cada objeto tendrá su espacio de memoria con un contenido invariable. Una declaración en cabecera de una clase como *private static final double Pi = 3.1416;* la interpretamos como una constante de clase. Existe un único espacio de memoria, compartido por todos los objetos de la clase, que contiene un valor invariable. Veamos ejemplos de uso:

```
// Ejemplo aprenderaprogramar.com
// Sintaxis: CaracterPublico/Privado static final TipoDeLaConstante = valorDeLaConstante;
private static final float PI = 3.1416f;  //Recordar f indica que se trata de un float
private static double PI = 3.1416;
public static final String passwd = "jkl342lagg";
public static final int PRECIO_DEFAULT = 100;
```

Cuando usamos la palabra clave *static* la declaración de la constante ha de realizarse **obligatoriamente en cabecera de la clase,** junto a los campos (debajo de la signatura de clase). Es decir, un atributo de clase hemos de declararlo en cabecera de clase. Si tratamos de incorporarlo en un método obtendremos un error. Por tanto dentro del método main (que es un método de clase al llevar incorporado static en su declaración) no podemos declarar constantes de clase. Por otro lado, *final* sí puede ser usado dentro de métodos y también dentro de un método main. Por ejemplo una declaración como final String psswd = "mt34rsm8" es válida dentro de un método. En resumen: en cabecera de clase usaremos *static final* para definir aquellas variables comunes a todos los objetos de una clase.

Modifica el código de la clase Calculadora que vimos anteriormente para declarar PI como una constante de clase. Luego, intenta modificar el valor de PI usando una invocación como *this.PI =22;*. Comprobarás que se produce un error al compilar ya que los valores de las constantes no son modificables.

EJERCICIO

Define una clase Java denominada Circulo que tenga como atributo de clase (estático) y constante numeroPi, siendo esta constante de tipo double y valor 3.1416. Además la clase tendrá el atributo radio (tipo double) que representa el radio del círculo, y los métodos para obtener y establecer los atributos. También debe disponer de un método para calcular el área del círculo (método tipo funcion areaCirculo que devuelve el área) y la longitud del círculo (método tipo función que devuelve la longitud). Busca información sobre las fórmulas necesarias para crear estos métodos en internet si no las recuerdas. En una clase con el método main, declara el código que cree un objeto círculo, le pida al usuario el radio y le devuelva el área y la longitud del círculo.

¿Es posible crear un método en la clase Circulo para establecer el valor de numeroPi? ¿Por qué?

Puedes comprobar si tu código es correcto consultando en los foros aprenderaprogramar.com.

Próxima entrega: CU00674B

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente:

http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=68&Itemid=188