

Unidad Temática 1 - Transición a Java

Recorté algunos extractos del libro que me parecieron interesantes o útiles para tener a mano.

applets y servlets ?

Publicado por leonardo (1 intervención) el 26/09/2016 23:53:31

Hola.

Son muy distintos un applet Java a un servlet.

Un applet java es una pequeña aplicación que se ejecuta al lado del cliente.

Un servlet es un programa que se ejecuta en un contenedor (tomcat) o servidor web. Quiero decir que utiliza el cliente/servidor. Los clientes lo invocan con el protocolo https y el que se encarga de ejecutar la el servlet es el contenedor el cual genera una página dinámica. Actualmente los applets estan quedando atrás por razones de seguridad y tiempo de ejecución. Es mucho más rápido ejecutar una tecnología cliente servidor que un applet.

Los servlet Java van evolucionando en cada nueva versión del javaEE; además existen mucho más tecnologías que pueden hacer uso de los servlets como spring.jsf,jsp,ajax, etc.. se pueden combinar y hacer maravillas.

Saludos.

Valora esta respuesta  2

Comentar

```
char c = (char) (Math.random()*26+'a'); // Generación aleatoria de letras minúsculas
System.out.println("La letra " + c );
switch (c) {
    case 'a': // Se compara con la letra a
    case 'e': // Se compara con la letra e
    case 'i': // Se compara con la letra i
    case 'o': // Se compara con la letra o
    case 'u': // Se compara con la letra u
        System.out.println(" Es una vocal "); break;
    default:
        System.out.println(" Es una consonante ");
}
```



4	Class (static) variables	First the public class variables, then the protected, and then the private.
5	Instance variables	First public, then protected, and then private.
6	Constructors	

4 - Indentation

	Part of Class/Interface Declaration	Notes
7	Methods	These methods should be grouped by functionality rather than by scope or accessibility. For example, a private class method can be in between two public instance methods. The goal is to make reading and understanding the code easier.

4.2 Wrapping Lines

When an expression will not fit on a single line, break it according to these general principles:

- Break after a comma.
- Break before an operator.
- Prefer higher-level breaks to lower-level breaks.
- Align the new line with the beginning of the expression at the same level on the previous line.
- If the above rules lead to confusing code or to code that's squished up against the right margin, just indent 8 spaces instead.

Here are some examples of breaking method calls:

```
function(longExpression1, longExpression2, longExpression3,  
        longExpression4, longExpression5);  
  
var = function1(longExpression1,  
               function2(longExpression2,  
                         longExpression3));
```

utilizados de los enumerados:

```
public enum Demarcacion{PORTERO, DEFENSA, CENTROCAMPISTA, DELANTERO}  
Demarcacion delantero = Demarcacion.DELANTERO; // Instancia de un enum de la clase Demarcación  
delantero.name(); // Devuelve un String con el nombre de la constante (DELANTERO)  
delantero.toString(); // Devuelve un String con el nombre de la constante (DELANTERO)  
delantero.ordinal(); // Devuelve un entero con la posición del enum según está declarada (3).  
delantero.compareTo(Enum otro); // Compara el enum con el parámetro según el orden en el que están declarados lo enum  
Demarcacion.values(); // Devuelve un array que contiene todos los enum
```