Información sobre clase atributos y métodos

Modificadores de Clase

class	Indica que es una clase	class ClaseEjemplo{..}
public class	Indica que puede ser accesible desde clases que se encuentren en el mismo paquete que esta clase y también desde clases que se encuentren en paquetes distintos, realizando un import. Solo puede haber una clase publica en un fichero java.	public class ClaseEJemplo{...}
abstract class	Indica que no se pueden crear instancias de esta clase con el operador new, Puede contener métodos sin implementar de tipo abstract Otras clases pueden heredar de ella implementanda todos los metodos que no lo estén. En Caso de que la clase hija sea abstracta, puede implementar solo algunos métodos	abstract class ClaseEjemplo{..}
final class	Indica que no puede tener clases hijas o subclases. Es decir. ninguna clase puede heredar de ella	final class ClaseEjemplo{...}
sin modificador class	Indica que puede ser accesible solo desde clases que se encuentren en el mismo paguete que esta clase.	class ClaseEjemplo

Modificadores de acceso de Métodos y Atributos

public	Accesible desde cualquier lugar	public int atributoEjemplo
sin_modificador	Accesible solo desde cualquier clase que se encuentre en el mismo paquete	int atributoEjernplo
protected	Accesible desde las clases hijas que se encuentren en cualquier paquete. O accesible desde de cualquier clase que se encuentre dentro del mismo paquete, sea o no su hija.	protected int atributoEjemplo
private	Accesible solo dentro de la clase en la que fue declarado	private int atributoEjemplo
static	No se necesita una instancia de la clase para poder acceder/modifcar un atributo o ejecutar un método. Se accede al atributo o método a través del nombre de la clase. Un método static solo puede acceder a atributos y métodos static. en caso de que estos estén declarados en la misma clase. Se puede acceder a atributos y métodos no static a través de objetos. El valor de un atributo static es compartido por todos los objetos, y si un objeto lo modifica, los demas objetos perciben la modificación. A estos atributos se les suele llamar atributos de Clase.	public static atributoEjemplo
final	No permiten reescrituras ni sobrecargas.Se suele utilizar si se desea declarar una constante. También en el caso de que se quiera evitar que modifiquen un método definido en la clase padre en una clase hija. (Evitar override)	public static final CONSTANTE_EJEMPLO;

synchronized	Evita que dos hilos puedan acceder al mismo metodo a la vez, para evitar problemas de acceso concurrente. Dejaria al segundo hilo en espera hasta que terminase el primero. (No se usa en atributos)	public synchronized int metodo{...}
volatile	Tiene el el mismo electo que synchronized, pero este se aplica solo a atributos.	public volatile int atributoEjemplo
transient	Evita que un atributo sea serializado, al serializar el objeto que lo contiene.	private transient int AtributoEjemplo
native	native es un modificador utilizado cuando un determinado método está escrito en un lenguaje distinto a Java. normalmente C. C++ o ensamblador para mejorar el rendimiento. La forma más común de implementar estos métodos es utilizar JNI (Java Native Interface). (No aplicable a atributos)	private native int metodo();

Tabla de accesibilidad

	La misma clase	Otra clase del mismo paquete	Subclase de Otro paquete	Otra clase de Otro paquete
public	X	X	X	X
protected	X	X	X	
sin_modificador	X	X		
private	X			

Crear objetos

Crear un Objeto de una clase
NombreClase nombreObjeto = new NombreClase();

Crear Objeto de clase interna estática
ClaseExterna.ClaseInterna nombreObjeto = new ClaseExterna.ClaseInterna()

Crear objetos de clase interna no estática
ClaseExterna nombreObjetoExterna = new ClaseExterna();
ClaseInterna nombreObjetoInterno = nombreObjeto.new ClaseInterna();

Crear objeto de clase anónima:

```
Clasepadre nombreObjeto = new ClasePadre {
        métodoDeClasePadre(){
                Cambios en el método
        }
};
nombreObjeto.metodoDeClasePadre() //Llamada al método de la clase anonima.
```

Métodos y atributos estáticos

Crear un método o atributo estático
public static int nombreAtributo;
public static void nombreMetodo();

Acceder a un método estático.
NombreClase.nornbreMetodo(); //No se necesita crear un Objeto con new para poder usarlo.

Acceder a un atributo estätico.
NombreClase.nornbreAtributo //NO se necesita crear un objeto new para poder usarlo.