

EXPLICACIONES P6-6.

1- Public class Foo:

Esta clase tiene una variable estática llamada x y un método void go() no estático.

El valor de esta variable estática será el mismo para todos los objetos de la clase.

Sí que compila porque un método no estático, sí que puede acceder a una variable estática. Si fuese al revés no.

2- Public class Foo2:

Esta clase tiene una variable de instancia y un método estático.

El método estático de la clase no puede ver el estado de la variable de instancia. No compila porque la variable de instancia int x; debería ser estática.

3- Public class Foo3:

Esta clase tiene una variable final int x; y un método void go() no estático.

No compila porque la variable final int x; no está inicializada. La inicializo con un valor para que compile.

La variable final una vez sea inicializada ya no cambiará su valor. Con esta variable utilizamos barra baja _ para separar dos palabras .

4- Public class Foo4:

Esta clase tiene una variable estática final int x con un valor de 12, y un método público void go();

Sí que compila porque la variable estática final, sí que está inicializada, y el método void go() no estático, sí que tiene acceso a la variable estática.

5- Public class Foo5:

Esta clase tiene la misma variable estática final int x con un valor de 12, y un método público void go (final int x); en el cual se le pasa por parámetro la variable estática final x.

Sí que compila porque la variable estática final, sí que está inicializada, y el método void go() no estático, sí que tiene acceso a la variable estática.

6- Public class Foo6:

Esta clase tiene una variable int x con un valor de 12, y un método público estático void go(final int x), y se le pasa por parámetro la variable final int x, la cual no ha sido declarada como final si no como entero a secas.

No compila porque el método estático está intentando acceder al valor de la variable de instancia y da error porque no tiene acceso. Solicita declarar la variable x como estática.