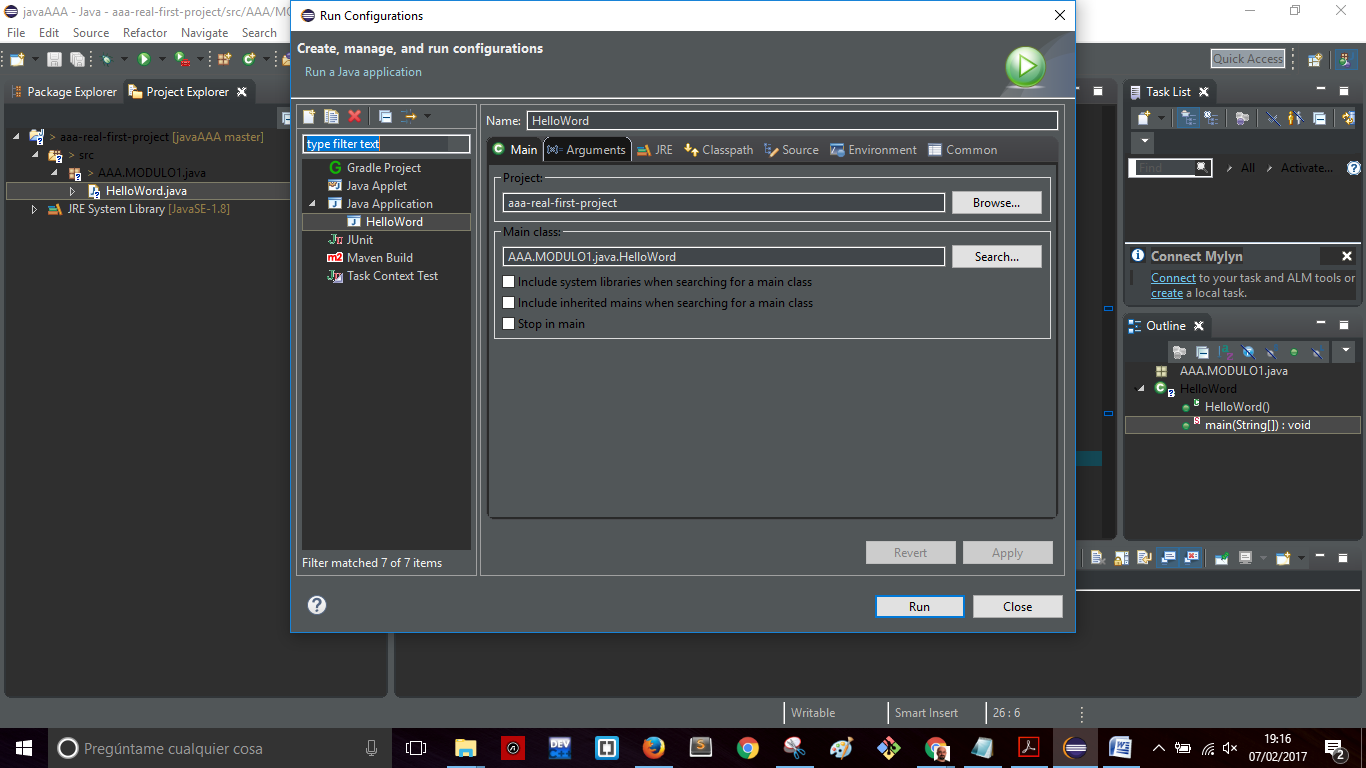


Existen 3 comentarios

Javadoc (para crear documentacion)

Línea

De bloque



Para lanzar con argumentos

# Reglas de Oro

## Solo una clase publica por fuente (por documento.java)

## Comentarios no cuentan

## Si hay una clase pública debe coincidir con el nombre del fuente (case sensitive)

## Sentencia package debe ser siempre la primera del fuente (si no, no compila)

Package

Imports

Class (clases por defecto)

Public class

Modificador “default” (cualquier clase del paquete puede acceder)

Modificador public (puede ser accedida desde otros paquetes con un import )

Modificador final (cortamos la herencia, no puede ser extendida)

Modificador abstract (no se puede crear un modelo, instanciado con un new)

Clase abstracta puede tener métodos con código o sin el código (métodos abstractos)

Los métodos abstractos deben ser rellenados en la clase hija (sino te dice que falta de implementar)

Si la clase hija es también abstracta no está obligada a completar los métodos abstractos del padre.

Modificador interface (clase abstracta estrictamente, todos sus metodos serán abstractos)

son public y abstractos (la clase siempre será public y abstracto)

Se heredada no con extend sino con implement

Interfaz es un contrato, sabes lo que entra y lo que sale ( el interior se implementa por el hijo o el nieto)

No admiten no métodos estaticos

No admiten métodos finales

Pueden extender otros interfaces

Pero no puede implementar los interfaces

# IDENTIFICADORES

COMENZAR con letra, el carácter $, carácter de unión como \_

No pueden empezar con numero

Sin limite en los caracteres

No puedes usar palabras reservadas

Case sensitive

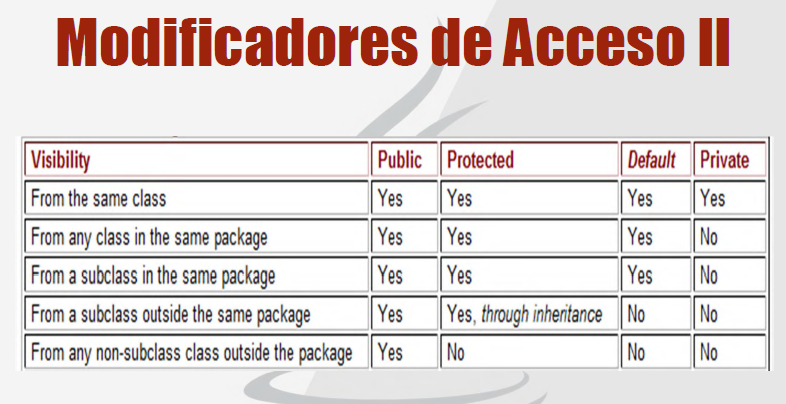
MODIFICADORES de acceso para propiedades

public (si la clase que lo contiene es public cualquiera puede acceder)

prívate (no puede ser accedido por ningún mecanismo externo)

“default” no podemos nada (solo accedido por clases del mismo paquete)

Protected (solo accedido por subclases de la clase)



Modificadores de métodos

Final no puede sobrescribirlo nadie

Final sobre una variable lo declara como constante.

abstract sobre método (método declarado pero no implementado)

si existe un metodo abstract todo la clase será abstract

synchronized a nivel de método (solo puede ser atacado a la vez por un hilo)

no valido para variable ni clases

native método implementado den lenguaje distinto de la plataforma

transient no serializa la variable

volatile (los diferentes hilos sincronizan el ultio valor de una variable

strictfp fuerza a coma flotante IEEE 754 (una expecificacione antigua) de clase y método (no de variable)

# CONSTRUCTORES