

16. modificateur d'accès

final

- variable locale final = constante
- attribut/variable d'instance final = constante
- méthode d'instance final = méthode non surchargeable et non redéfinissable
- classe final = ne peut avoir de sous-classe

static

- attribut static = variable de classe = existe en un seul exemplaire quelque soit le nombre d'instanciation (ou sans instanciation)
- méthode static = méthode de classe = méthode "sans envoi de message" appliquée à la classe et non à un objet.
- code static = static { instructions } initialise les variables static d'une classe

public

- public = la variable ou la méthode (ou le constructeur) ou la classe est accessible partout
- variable d'instance : rarement public sauf si final.

private

- les variables ou les méthodes (ou constructeurs) ne sont accessibles que par les membres internes de la classe

protected

- protected = les variables ou les méthodes (ou constructeurs) ne sont accessibles que par les membres internes de la classe, de ses sous-classes et des classes du même package.
- classe protected = accessible que par ses sous-classes et des classes du même package.

private protected

- les variables ou les méthodes ne sont accessibles que par les membres internes de la classe, de ses sous-classes.

pas de modificateur d'accès :

- acces package = les variables ou les méthodes (ou constructeurs) ou classes ne sont accessibles que par les membres internes de la classe, et des classes du même package.

abstract :

- une méthode abstraite ne contient pas de corps, mais doit être implémentée dans les sous-classes,
- une classe est abstraite si elle contient au moins une méthode abstraite ; elle ne peut pas être instanciée, mais ses sous-classes non abstraites le peuvent.
- une interface est par défaut abstraite.

autre modificateurs :

- native : méthode écrite en C dans un fichier externe
- synchronized : variable à accès en exclusion mutuelle pour les threads
- transient
- volatile