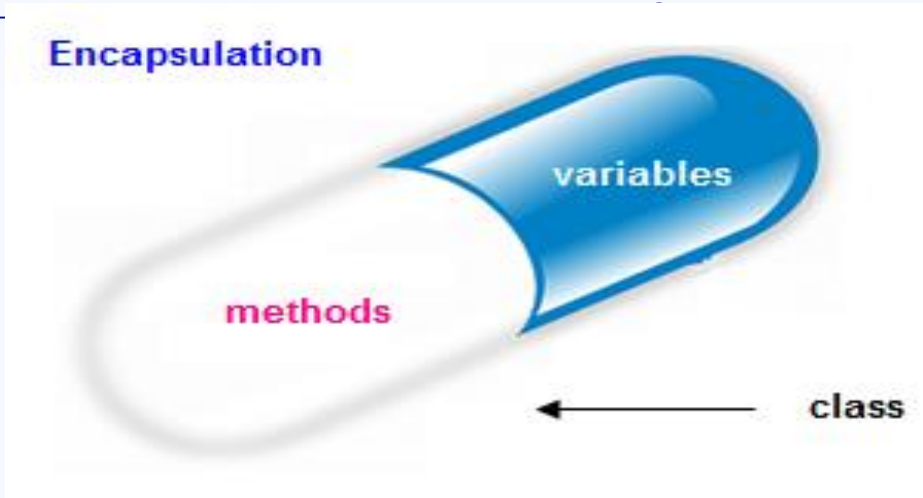
Encapsulation & packaging

2023-2024

One of the essential techniques of object-oriented programming is **encapsulation**. its interest is to protect as much as possible the code present in a class from the outside



01

Encapsulation

Encapsulation consists of creating a sort of black box containing internally a protected mechanism and externally a set of commands which will allow it to be manipulated. This set of commands is made in such a way that it will be impossible to alter the protected mechanism in the event of misuse.

For each attribute or method, it is possible to define adegree of visibility from outside the class

By default, ***attributes*** should be hidden. Their values should only be visible and modifiable through methods. However, it is acceptable to leave visible the ***constants***

The attributes must a priori be private. If necessary, ***public access methods*** and ***alteration methods*** are defined and allow access to attributes to be controlled.

Intermediate methods that are not intended to be used outside the class should be hidden.

02

Modifiers or encapsulation levels

- All attributes and methods of a class can be used in the body of the class itself using their simple names.
- However, the Java language allows you to specify access restrictions to members of a class (attributes and methods) outside of that class (called a definition class or declaration class).
- Member access control is carried out using different keywords (reserved words called ***modifiers***) which can precede the declaration of attributes and methods.

Modifier	Description of access rights on the member
Public	Access is possible by, or in, any other class
Private	restrict access to only the class in which it is declared
protected	restrict access to derived classes (subclasses)
none	Accessible to all classes of the same package

The charter of good OO programming recommends against the useof “protected”. Besides, Python does not have this level of encapsulation

Visibility of members (attributes/methods) according to their modifiers and encapsulation level

Modifier	Class	package	subclass	Other class
Private	visible			
None	visible	visible		
Protected	visible	visible	visible	
Public	visible	visible	visible	visible

02

access methods/ alteration methods

- ***access methods*** provide information relating to the state of an object, that is to say the values of certain of its attributes (generally private), without modifying them;
- ***alteration methods*** (mutator) which modify the state of an object, therefore the values of certain of its attributes.
- We often encounter the use of names of the form getXXXX (***getters***) for access methods and setXXXX (***setters***) for alteration methods

```
public double getLongueur() {
return longueur;
}
public void setLongueur(double longueur) {
this.longueur = longueur;
}
public double getLargeur() {
return largeur;
}
public void setLargeur(double largeur) {
this.largeur = largeur;
}
```

03

Encapsulation advantages

- Hiding implementation details (which can be changed without consequences for external users)
- Protection of the class against incorrect uses (consistency checks in methods)
- API simplification (internal fields and methods are hidden).