Object-Oriented Design

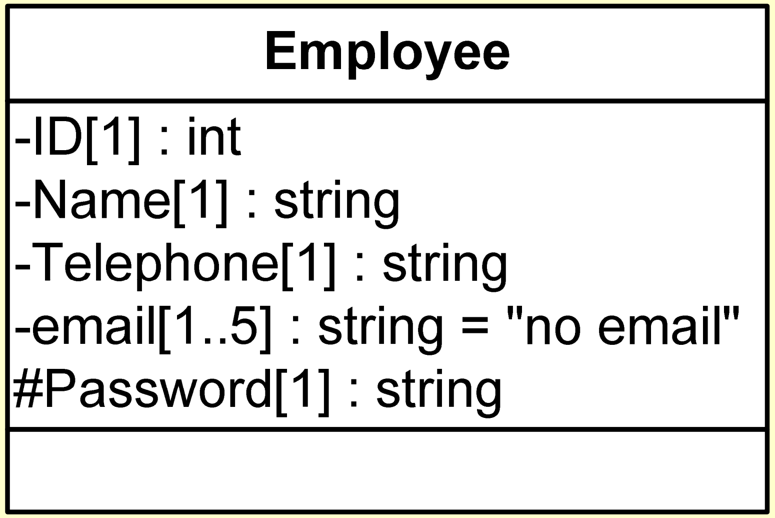
Attribute and method properties

The syntax of attributes

```
visibility name [multiplicity] : type = default
  {property}
```

- visibility (optional): specifies if the attribute is accessible from outside the class
- name (mandatory): the name of the attribute
- multiplicity (optional): specifies the number of values the attribute can store. Default value: 1
- type (optional): specifies the data type of the attribute. Basic data types are: bool, int, float, string.
- default value (optional): specifies the initial value of an attribute
- property (optional): contains additional information about the attribute (more on this later)

The syntax of attributes: example



Visibility (or accessibility)

- Indicates if the attribute can be accessed from outside the class. It can be :
  - public (+) : accessible from outside
  - private (-) : accessible only by the class' methods
  - protected (#) : accessible only by the class' methods and by the class' descendants

Important!

Attributes must (almost) always be

PRIVATE

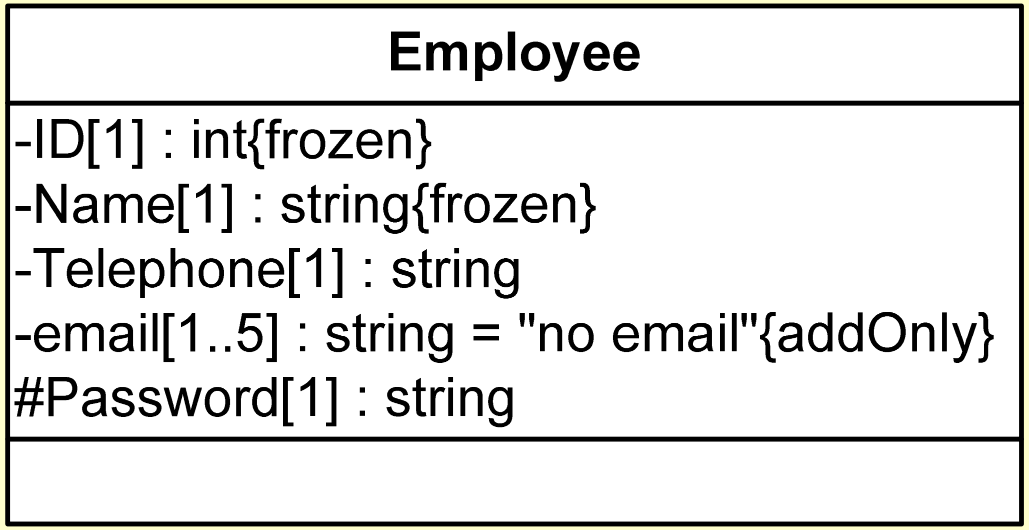
or PROTECTED at most

see: encapsulation

Attribute Properties

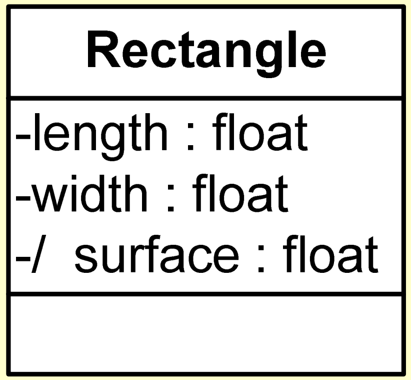
- An attribute can be
  - {changeable}: default, can be read and written
  - {frozen}: a constant
  - {addOnly}: when the attribute is a container (multiplicity > 1). Elements can only be added.

Attribute properties: example



Derived attributes

- Attributes which can be computed from other attributed.
- They are indicated by a /.

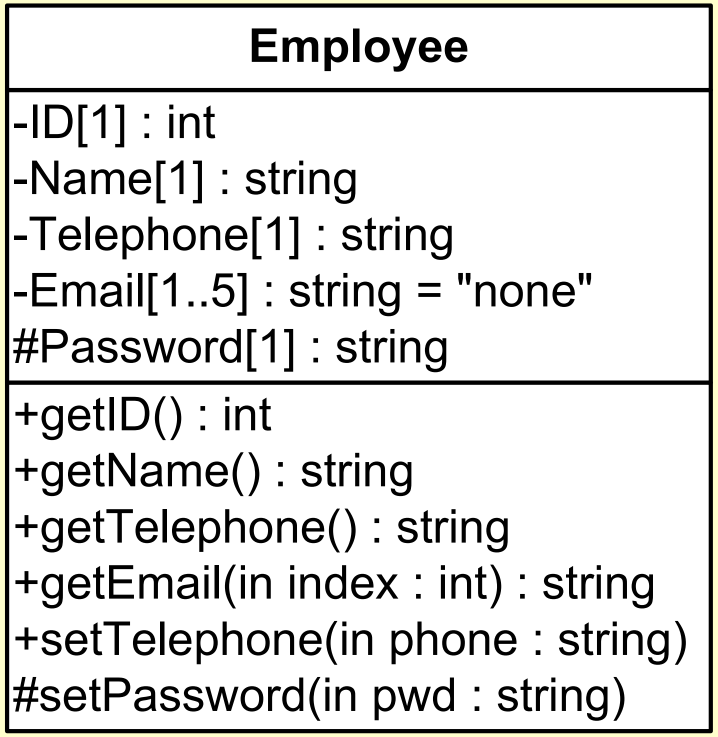


The syntax of methods

```
visibility name (param_list): return_type
```

- visibility (optional): specifies if the method is *callable* from outside the class
- name (mandatory): the name of the method
- param_list (optional): the list of parameters for the method
- return_type (optional): specifies the data type which is returned by the method. Basic data types are: bool, int, float, string.

The syntax of methods

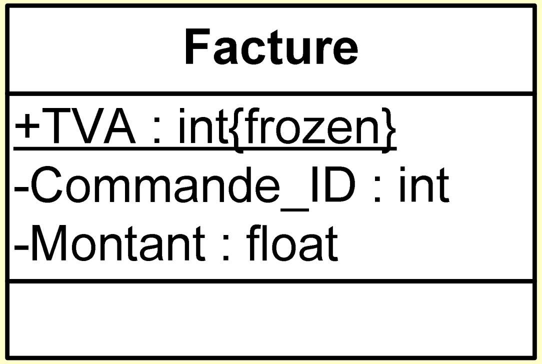


On visibility of methods

- Usually methods are made public
- Private methods are usually used to implement a complex public method without exposing too many internal details

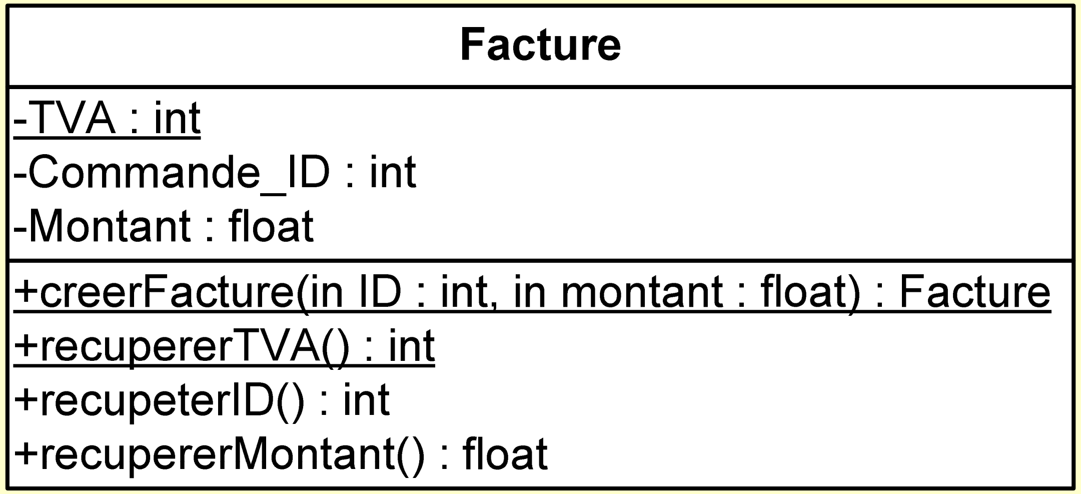
Class (or static) attributes

- A class (or static) attribute is an attribute common to all instances of a class



Class (or static) methods

- A class (or *static*) method is a method which can be invoked without instantiating a class
- Example : methods which generate new classes


Facture
-TVA : int
-Commande_ID : int
-Montant : float
+creerFacture(in ID : int, in montant : float) : Facture
+recupererTVA() : int
+recupeterID() : int
+recupererMontant() : float