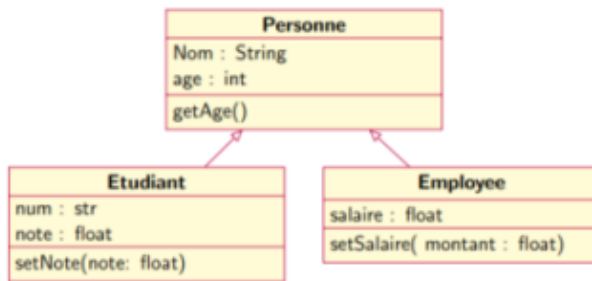


# Python Inheritance and polymorphism

## I. Inheritance

Inheritance allows us to define a class that inherits all the methods and properties from another class. The following figure shows a parent class (Personne) and two child classes (Etudiant and Employee).



**Parent class** is the class being inherited from, also called base class.

**Child class** is the class that inherits from another class, also called derived class. The child class inherits all the members (attributes and methods) of its parent class. It may have additional members.

### Example:

#### 1. Create a Parent Class

```
class Person:

    def __init__(self,nom,prenom):

        self.nom=nom

        self.prenom=prenom

    def print(self):
```

```
print(self.nom)
print(self.prenom)
```

2. Use the Person class to create an object, and then execute the print method.

3. Create a child class

```
class Student(Person):
    def __init__(self, nom, prenom, bac):
        Person.__init__(self, nom, prenom)
        self.bac = bac
    def print(self):
        Person.print(self)
        print(self.bac)
```

4. Use the Student class to create an object, and then execute the print method:

5. Give the attributes of a Person object.
6. Give the methods of a Person object.
7. Give the attributes of a Student object.
8. Give the methods of a Student object.
9. Add a method which modifies the bac attribute.
10. Define the concept of inheritance

## II. Polymorphism

The word polymorphism means having many forms. In programming, polymorphism means same function name being used for different types. Let us illustrate this concept through the following example.

1. Write a parent class Shape defined by coordinates x and y. Write its constructor and the print method.

2. Write a class named Square which inherits from the shape class. It has an additional attribute, called length. Redefine the print method. Define the perimeter method.
3. Write a class named rectangle which inherits from the shape class. It has two additional attributes, called length and width. Redefine the print method. Define the perimeter method.
4. Write a class named rectangle which inherits from the shape class. It has two additional attributes, called length and width. Redefine the print method. Define the perimeter method.
5. Execute and understand. Then show the interest of the polymorphism concept through this example.

```
shapes=[]

c1=square(1,3,56)

c2=square(-3,-3,100)

r1=rectangle(3,6,10,10)

r2=rectangle(-3,-6,10,10)

shapes.append(c1)

shapes.append(r1)

shapes.append(c2)

shapes.append(r2)

for s in shapes:

    s.print()

    print('The perimeter=',s.perimeter())

    print('-----')
```