POO

Effet procédurale

Problème :

- maintenabilité - -

POO concept pour construire des programmes modulaire (année 60)

Pouvoir représenter tous les objets de notre environnement (3d), faire des objets

OBJET :

- caractéristique (propriété, attribut)

- action (méthode)

La classe manipule l’objet, définis les caractéristiques

L’objet est indépendant, prend des valeurs

Chaque Class a son programme puis l’importer

Manipulation d’objet leur donner des classes

Class + Nom de la classe :

Constructeur 🡪 def \_\_init\_\_(self) :

Self.nom d’attribut = « l’attribut qu’on donne »

Class Personne :

def \_\_init\_\_(self) :

Self.nom= « DOE »

Self.prenom= « John »

Self = ref vers l’objet courant

Magic méthode = constructeur prédéfinis (Ex : Class)

Nom de la variable = Class()

Personne\_A = Personne()

Rouge = Instanciation

Class Personne :

def \_\_init\_\_(self,nom,prenom) :

Self.nom= nom

Self.prenom= prenom

Personne\_A = Personne(« Doe », « John »)

4 Principes de la POO

* Encapsulation : chaque objet prend des propriétés et des méthodes propres à lui (on peut tout mettre dans un objet)
* Polymorphisme : On peut avoir des méthodes et fonctions qui ont le même nom (ex : boire pour une personne et un chien, les deux classes pourront l’exécuter)
* Abstraction : Rien d’obligatoire de donner trop de détails à l’utilisateur, cela permet à l’utilisateur d’implémenter une logique plus complexe sans comprendre ni même penser à toute la complexité cachée.
* Héritage : On peut créer des objets qui hérite des class mère pour réutiliser les propriétés (ex : Un homme est un humain)