

UNIVERSITE AFRICAINE DE MANAGEMENT ET TECHNOLOGIE

UATM – Département Génie Electrique



Domaine : Système Informatique et Logiciel (SIL)

Niveau : L1 – Génie Electrique

PROJET DE PROGRAMMATION PYTHON

Thème du projet:

Logiciel de gestion des inscriptions et des
notes en mode console.

Réalisé par :

PEDROS MARCOS Samuel

Encadré par :

Mr DJOHOU Carmel Emeric

Année universitaire : 2024 – 2025

PLAN

I- Introduction

II- Objectifs du projet

III- Structure du programme Python

III.1- La classe *etudiant*

III.2- La fonction *Inscription()*

III.3- La méthode *FICHE_INSCRIPTION()*

III.4- Génération automatique d'une base de données

V- Conclusion

I- INTRODUCTION

Dans le cadre de notre formation en première année de Licence en Génie Électrique à l'UATM GASA-FORMATION, un projet de fin de semestre nous a été confié afin de mettre en pratique les compétences acquises en langages Python. Le projet consiste à développer une **application simple** fonctionnant exclusivement en console, pour la **gestion des inscriptions d'étudiants et le traitement automatique de leurs notes**.

Nous avons choisi une approche orientée objet, tout en respectant la simplicité du fonctionnement par terminal, sans interface graphique, afin de nous concentrer sur la logique, la structure et la gestion des fichiers.

II- OBJECTIFS DU PROJET

Ce projet vise à atteindre les objectifs suivants :

- Concevoir une **classe Python** modélisant un étudiant.
- Permettre à l'utilisateur de **s'inscrire** en saisissant ses informations.
- Permettre la **saisie des notes** obtenues dans plusieurs matières.
- Calculer automatiquement la **moyenne générale**.
- Afficher une **décision du jury** (admis ou non).
- Générer un **bulletin de notes** sous forme de fichier *.txt* nommé automatiquement.
- Créer une base de données (fichier texte personnalisé) pour insérer et conserver les informations des étudiants.

Le programme de ce projet repose sur une structure modulaire, bien organisée. Nous avons :

- Une **classe** *etudiant*, contenant toutes les données personnelles (Nom, Prénoms, Sexe, date de naissance, Lieu de naissance).
- Une méthode *Infos_Etudiant()* qui, si on l'appelle, doit afficher toutes les informations de l'étudiant.
- Une **fonction** *Inscription()* qui permet à l'utilisateur de saisir ses informations via *input()*, puis retourne un objet *E* contenant les informations de l'étudiant.
- Une **méthode** *FICHE_INSCRIPTION()* qui s'occupe de la saisie des notes, du calcul de la moyenne, de la décision du jury et de l'écriture dans un fichier.
- Un **programme principal** qui appelle ces éléments pour exécuter l'application de bout en bout.

III- STRUCTURE DU PROGRAMME PYTHON

III.1- La classe *etudiant*

La classe *etudiant* comporte :

- Cinq attributs privés :

```
class etudiant:
    #Création du constructeur
    def __init__(self, Nom, Prenom, Sexe, date_de_naissance, Lieu_de_naissance):
        self.__Nom = Nom
        self.__Prenom = Prenom
        self.__Sexe = Sexe
        self.__date_de_naissance = date_de_naissance
        self.__Lieu_de_naissance = Lieu_de_naissance
```

Projet Python - UATM

- Des méthodes d'accès (**getters**) et des méthodes de modification (**setters**) pour chacun de ces attributs.

```
18     #Création des getters
19     def get_Nom(self):
20         return self.__Nom
21     def get_Prenom(self):
22         return self.__Prenom
23     def get_Sexe(self):
24         return self.__Sexe
25     def get_date_de_naissance(self):
26         return self.__date_de_naissance
27     def get_Lieu_de_naissance(self):
28         return self.__Lieu_de_naissance
29
30     #Création des setters
31     def set_Nom(self, Nom_etudiant):
32         self.__Nom = Nom_etudiant
33     def set_Prenom(self, Prenom_etudiant):
34         self.__Prenom = Prenom_etudiant
35     def set_Sexe(self, Sexe_etudiant):
36         self.__Sexe = Sexe_etudiant
37     def set_date_de_naissance(self, date_de_naissance_etudiant):
38         self.__date_de_naissance = date_de_naissance_etudiant
39     def set_Lieu_de_naissance(self, Lieu_de_naissance_etudiant):
40         self.__Lieu_de_naissance = Lieu_de_naissance_etudiant
41
```

- Une méthode *Infos_Etudiant()* qui affiche toutes les informations de l'étudiant.

```
#Création de la méthode Infos_Etudiant()
def Infos_Etudiant(self):
    print("\nNom: ", self.get_Nom())
    print("Prenom: ", self.get_Prenom())
    print("Sexe: ", self.get_Sexe())
    print("Date de naissance: ", self.get_date_de_naissance())
    print("Lieu de naissance: ", self.get_Lieu_de_naissance())
```

- Une méthode *FICHE_INSCRIPTION()* pour :
 - recueillir les matières et les notes.
 - calculer la moyenne.
 - afficher la décision du jury.
 - enregistrer toutes les données dans un fichier texte personnalisé.

III.2- La fonction *Inscription()*

Cette fonction créée hors de la classe *etudiant* demande à l'utilisateur :

- Son nom et prénom,
- Son sexe avec vérification (m ou M pour les hommes et f ou F pour les femmes),
- Sa date de naissance (au format << JJ/MM/AAAA >>),
- Son lieu de naissance.

Projet Python - UATM

Et retourne un objet *E* qui sera utilisé pour la saisie des notes.

```
#Création de la méthode Inscription()
def Inscription():
    print("***** INSCRIPTION *****")
    Nom=input("Entrez votre nom: ")
    Prenom=input("Entrez votre prenom: ")
    Sexe=input("Entrez votre sexe: ")
    liste=["m", "M", "f", "F"]
    while Sexe not in liste:
        Sexe=input("Entrez votre sexe(M ou m pour Les hommes et F ou f pour Les femmes): ")
    date_de_naissance=input("Entrez votre date de naissance(JJ/MM/AAAA): ")
    Lieu_de_naissance=input("Entrez votre lieu de naissance: ")

    #Création de l'objet E
    E = etudiant(Nom, Prenom, Sexe, date_de_naissance, Lieu_de_naissance)
    return E
```

Elle s'exécute comme suit avec un exemple :

```
***** INSCRIPTION *****
Entrez votre nom: PEDROS MARCOS
Entrez votre prenom: Samuel
Entrez votre sexe: M
Entrez votre date de naissance(JJ/MM/AAAA): 18/06/1999
Entrez votre lieu de naissance: Brazil - Rio

Nom: PEDROS MARCOS
Prenom: Samuel
Sexe: M
Date de naissance: 18/06/1999
Lieu de naissance: Brazil - Rio
```

III.3- La méthode *FICHE_INSCRIPTION()*

Cette méthode :

- Demande le nombre de matières.
- Pour chaque matière, demande :
 - le nom de la matière
 - la note obtenue (entre 0 et 20)

```
50 #Création de la méthode FICHE_INSCRIPTION()
51 def FICHE_INSCRIPTION(self):
52     nbr_mat=int(input("\nDonnez le nombre de matières composées: "))
53     dictionnaire = {}
54     for i in range(nbr_mat):
55         if i == 0:
56             mat=str(input("Donnez le libellé d'une matière: "))
57         else:
58             mat=str(input("Donnez le libellé d'une autre matière: "))
59             note=float(input("Donnez la note obtenue: "))
60             if note < 0 or note > 20:
61                 note=float(input("La note doit être compris entre 0 et 20: "))
62             dictionnaire[mat]=note
63
64     print("\n***** INFOS ETUDIANT *****")
65     self.Infos_Etudiant()
66
67     print("\n***** NOTE OBTENUE / MATIERE *****")
68     for key in dictionnaire:
69         print(f"{key}:", dictionnaire[key],"/20")
70
```

Projet Python - UATM

Et s'exécute comme suit avec un exemple de 3 matières :

```
Donnez le nombre de matières composées: 3
Donnez le libellé d'une matière: Merise
Donnez la note obtenue: 18
Donnez le libellé d'une autre matière: TPEP Web
Donnez la note obtenue: 19
Donnez le libellé d'une autre matière: Langage C
Donnez la note obtenue: 11

***** INFOS ETUDIANT *****

Nom: PEDROS MARCOS
Prenom: Samuel
Sexe: M
Date de naissance: 18/06/1999
Lieu de naissance: Brazil - Rio

***** NOTE OBTENUE / MATIERE *****
Merise: 18.0 /20
TPEP Web: 19.0 /20
Langage C: 11.0 /20
```

- Calcule la moyenne générale :

```
71         #Calcul de la moyenne
72         if nbr_mat == 1:
73             moy = note
74         else:
75             somme=0
76             liste=list(dictionnaire.values())
77             for x in liste:
78                 somme += x
79             moy = (somme)/nbr_mat
80             print("\n***** MOYENNE *****")
81             print(f"Moyenne obtenue: {moy} /20")
82
```

S'exécute comme suit avec l'exemple précédent :

```
***** MOYENNE *****
Moyenne obtenue: 16.0 /20
```

- Détermine si l'étudiant passe au niveau supérieur :
 - Admis si 80% des matières ont une note ≥ 12 .

```
83         print("\n***** DECISION DU JURY *****")
84         if nbr_mat == 1:
85             if moy >= 12:
86                 decision = "Passe au niveau supérieur."
87             else:
88                 decision = "Ne passe pas au niveau supérieur."
89         else:
90             list_e=[]
91             for y in liste:
92                 if y >= 12:
93                     list_e.append(y)
94             nbr_bonnes_notes=len(list_e)
95             pourcentage=nbr_mat*0.8
96             if nbr_bonnes_notes >= pourcentage:
97                 decision = "Passe au niveau supérieur."
98             else:
99                 decision = "Ne passe pas au niveau supérieur."
100             print(decision)
```

Projet Python - UATM

S'exécute comme suit avec l'exemple du haut :

```
***** DECISION DU JURY *****  
Ne passe pas au niveau supérieur.
```

NB. Si le nombre de matières composées était > 5 et l'étudiant avait eu une seule note < 12 , il irait en classe supérieur.

- Affiche en commentaire les matières à améliorer (note < 12) :

```
101  
102     print("\n***** COMMENTAIRE *****")  
103     lyste=[]  
104     for key in dictionnaire:  
105         if dictionnaire[key] < 12:  
106             lyste.append(key)  
107             commentaire = f"Doit s'améliorer en {key}."  
108     if lyste == []:  
109         commentaire = "Félicitations. Vous avez fait un bon travail. Continuer sur cet  
110     print(commentaire)  
111
```


Ça donne :

```
***** COMMENTAIRE *****  
Doit s'améliorer en Langage C.
```

- Génère le fichier *Nom_Prenom.txt* contenant toutes les informations.

III.4- Création d'un fichier texte

Le fichier *.txt* est créé automatiquement avec le nom et le prénom de l'étudiant et contient :

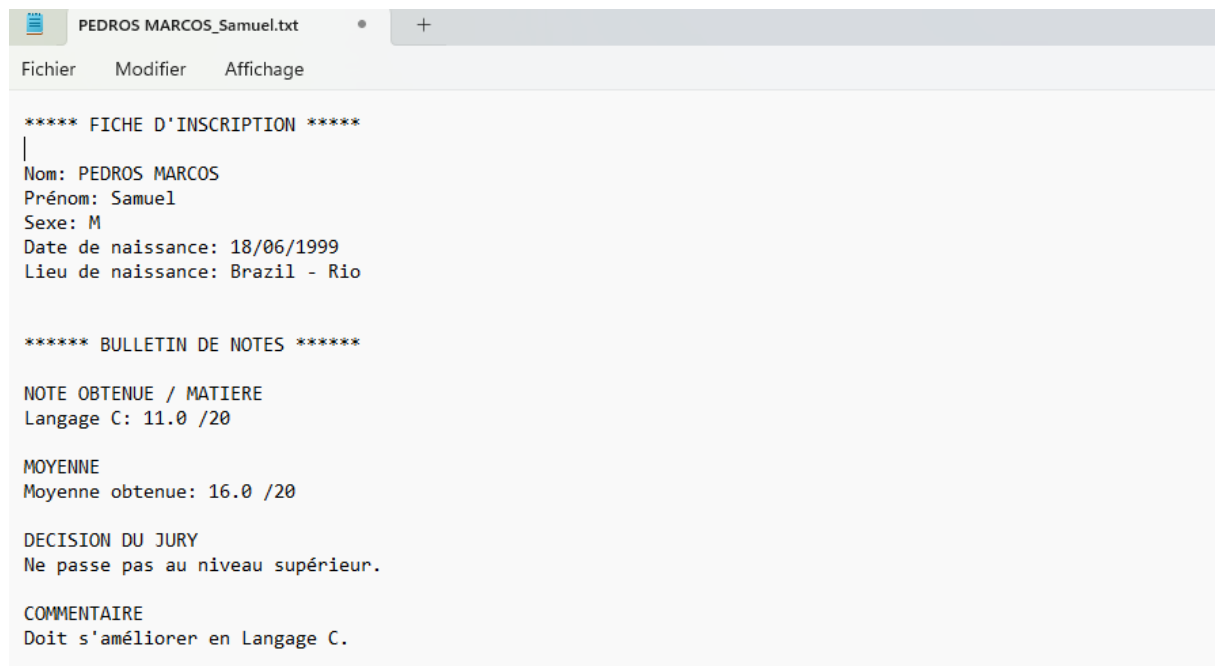
 PEDROS MARCOS_Samuel	14/05/2025 03:18	Document texte	1 Ko
--	------------------	----------------	------

- Les informations personnelles,
- Les matières et les notes,
- La moyenne,
- La décision du jury,
- Et si besoin, les matières à améliorer.

```
111  
112     #Création de la base de données  
113     with open(self.get_Nom()+"_"+self.get_Prenom()+".txt", "a") as f:  
114         f.write("***** FICHE D'INSCRIPTION *****")  
115         f.write("\n\nNom: "+self.get_Nom())  
116         f.write("\nPrénoms: "+self.get_Prenom())  
117         f.write("\nSexe: "+self.get_Sexe())  
118         f.write("\nDate de naissance: "+self.get_date_de_naissance())  
119         f.write("\nLieu de naissance: "+self.get_Lieu_de_naissance())  
120         f.write("\n\n***** BULLETIN DE NOTES *****")  
121         f.write("\n\nNOTE OBTENUE / MATIERE")  
122         f.write(f"\n\n{key}: {dictionnaire[key]} /20")  
123         f.write("\n\nMOYENNE")  
124         f.write(f"\nMoyenne obtenue: {moy} /20")  
125         f.write("\n\nDECISION DU JURY")  
126         f.write(f"\n\n{decision}")  
127         f.write("\n\nCOMMENTAIRE")  
128         f.write(f"\n\n{commentaire}")  
129
```


Projet Python - UATM

Ensuite les informations s'afficheront dans le fichier comme suit :



The screenshot shows a text editor window with a single tab titled 'PEDROS MARCOS_Samuel.txt'. The menu bar includes 'Fichier', 'Modifier', and 'Affichage'. The text content is as follows:

```
***** FICHE D'INSCRIPTION *****
|
Nom: PEDROS MARCOS
Prénom: Samuel
Sexe: M
Date de naissance: 18/06/1999
Lieu de naissance: Brazil - Rio

***** BULLETIN DE NOTES *****

NOTE OBTENUE / MATIERE
Langage C: 11.0 /20

MOYENNE
Moyenne obtenue: 16.0 /20

DECISION DU JURY
Ne passe pas au niveau supérieur.

COMMENTAIRE
Doit s'améliorer en Langage C.
```

IV- Conclusions

Ce projet nous a permis de maîtriser la programmation orientée objet, l'utilisation des fonctions et des méthodes, les structures conditionnelles et les boucles, la gestion des saisies utilisateurs et la validation des données et la création de fichier texte, utiles pour la gestion des données.

A l'avenir, ce programme pourrait évoluer vers une version avec interface graphique (Tkinter) et une version connectée à une base de données (SQLite) pour une meilleure gestion des données.