 **République Algérienne Démocratique et Populaire**

**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**

**Université Akli Mohand Oulhadj de Bouira**

**Faculté des Sciences et des Sciences Appliquées**

**Département d'Informatique**

**Compte Rendu de MTI**

***Spécialité*** *: GSI*

|  |
| --- |
| **TP n°5**  MVC |

**Réalisé par**

- Akli Meziane

-Ladjal Sarah

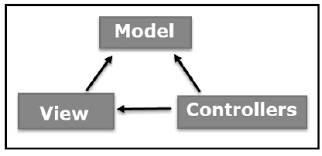
2019/2020

**Compte rendu de TP :**

Introduction

Le Framework Model-View-Controller(MVC) est un modèle architecturel (model de conception) qui sépare une application en trois composants logiques principaux, Model, View, et Controller, d’ou l’abréviation MVC.

Ce model il est décomposé en trois entités :



**La vue** : c’est est une interface homme machine(IHM).

Une page web, les interfaces graphiques des jeux, des applications…

**Le modèle :** il contient les données du programme.

Les algorithmes, la configuration de l’application, la logique.

C’est le cœur de l’application !

**Le contrôleur :** il contrôle les données, il transmet les informations entre la vue et le modèle (lorsque l’utilisateur effectue une action).

Ce model il nous a permet d’ajouter des nouvelles fonctionnalités rapidement et minimiser les bugs afin d’obtenir une application fiable, sans bug et performante.

Méthodologie de travail :

**Partie 1 :**

Dans le TP nous a demandé d’exécuter le code python qui permet de recherchée num dans un tableau par son nom, et l’afficher. Un ulitisé la boucle ***for personne in annuaire:***

**Parti 2 :**

Dans la deuxième partie, on veut reformuler le code précédent en séparant l’aspect affichage de l’aspect donné.

Les avantages de cette méthode : c’est de gagner le temps et faciliter la maintenance, aussi  clarifier la conception et donner une souplesse pour le programme en séparant les taches (Vue Donnes).

L’inconvénient : c’est que ce modèle on ne voit pas ses avantages dans ces petites programmes.

Apres on a ajouté la méthode ***get\_all()*** dans la classe modèle pour afficher pour les personnes .

On l’appelé depuis le contrôleur avec la méthode ***personnes =data\_model.get\_all().***

On a crée une classe de contrôle appelée contrôleur a la place de la fonction contrôleur.

Apres on a ajoutés une nouvelle fonctionnalité a notre système, l’ajout des nouvelles personnes.

On utilisé la fonction ***append()***

La partie qu’on doit ajouter pour afficher toutes les personnes dans l’annuaire

**Parti 3 :**

Dans cette partie on a ajoutées des nouvelles fonctionnalités à notre système :

Des méthodes d’ajout dans toutes les entités de MVC, une méthode d’ajout de nouvelles personnes dans le modèle de données, et dans l’interface View on a ajouté une méthode pour récupérer l’élément ajouté, et dans la dernière entité (contrôleur) on a ajouté une méthode pour rechercher une personne.

**Parti 4 :**

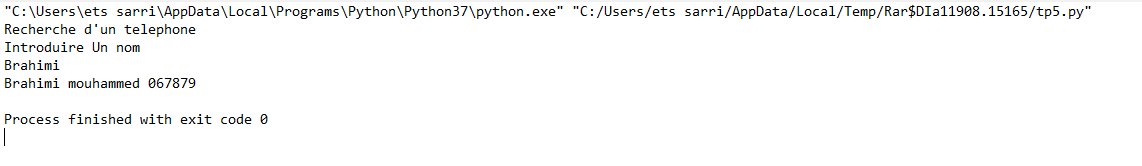
Dans la partie 4, la partie à modifier c’est le modèle et le contrôleur.

La partie a modifié si on veut stocker nos données dans une base de données contrôleur, modèle)

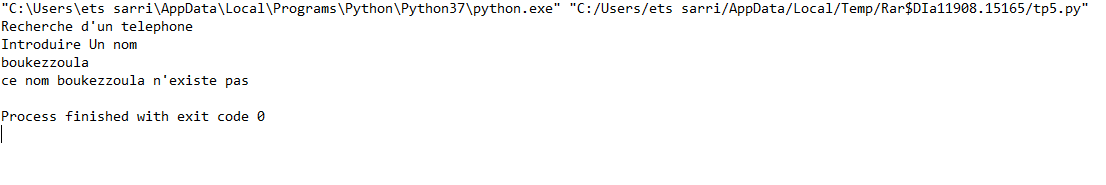
La partie a modifié si on veut créer une interface graphique pour notre programme, ou bien une interface web (contrôleur, Vue).

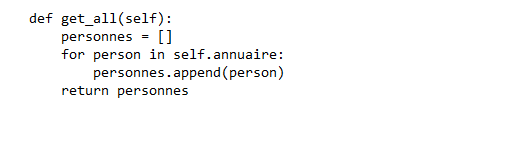
Quels sont les avantages de cette méthode de conception (souplesse de travail en divisant les taches et modifier très rapidement les interfaces ou modèle de stockage).

Teste et résultat



Dans le cas ou le nom n’existe pas :



:

L’affichage :



**Travaille  a domicile** :

Introduction

On a crée un programme en utilisant le modèle MVC, on a crée la classe contrôleur qui instanciée les objets des classes vue et modèle.

Dans la classe modèle on a lu le fichier XML qui contient les résultats en utilisant le XML DOM, on a déclaré des liste et on a ajouté des ces listes les statistiques qu’on a besoin (moyenne max min),

On a crée la méthode qui affiche tout les étudiants et leurs moyennes.

Dans la classe vue on a crée les méthodes input et output et ajout\_input pour faire les entrés-sortis et afficher les statistiques.

Méthodologie de travail

***Self. Moyennes = self.xmldoc.get.ElementsByTagName(‘moyannuel’)*** permet de récupérer les moyennes depuis l’arbre DOM.

***New Ele = self.xmldoc.createElemnt(‘’nom’’)*** ;permet de créer un élément nom pour l’ajouter dans l’arbre.

**Enom=self.xmldoc.createTextNode(nom)** ;permet d’inserer une information dans l’élément.

**groupe.appendChild(ett)** ;permet d’ajouter un étudiant dans le groupe.

**Contro.ajouter ()** permet d’appeler de l’ajout.

Les outils :

On a utilisé dans des sites web comme w3school et openclassroom pour récupérer des fonctions qui nous aide a traité les donnés et aussi l’api pour traiter l’arbre xml, sans utiliser des bibliothèques ou des Framework.

Teste et résultat

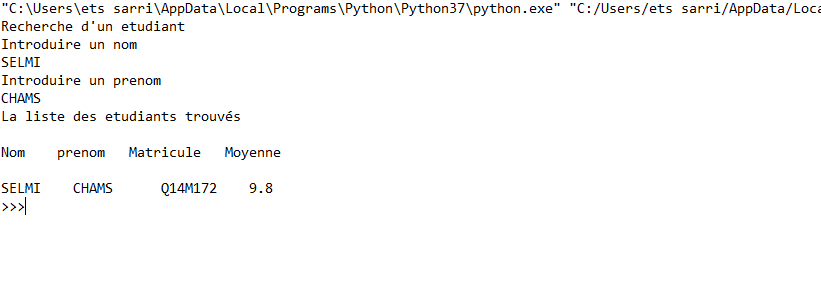


Figure1-Recherche d’un élément donné par l’utilisateur

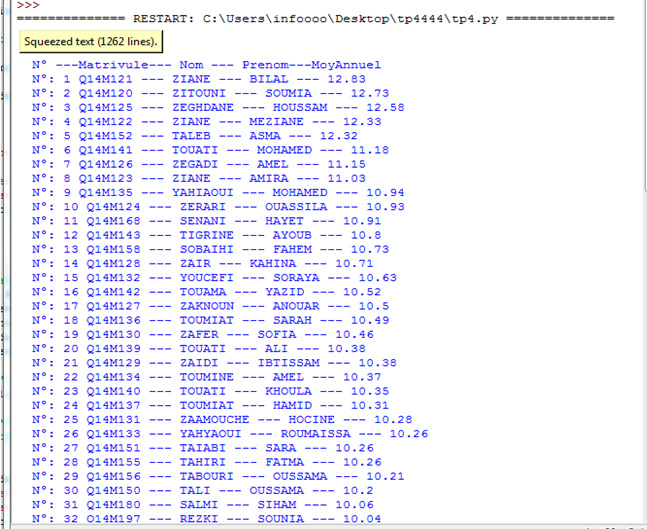


Figure 2-Recherche tout les étudiants

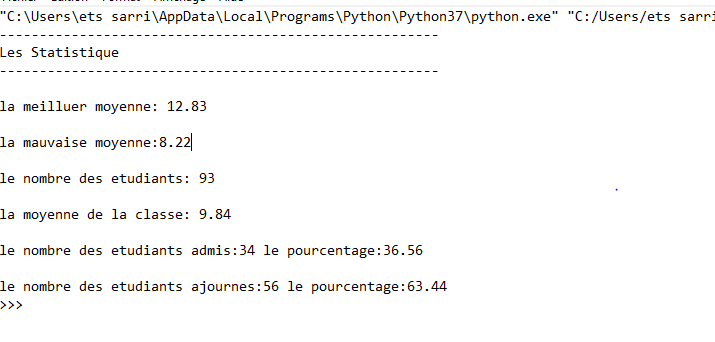


Figure3 : la liste des statistiques

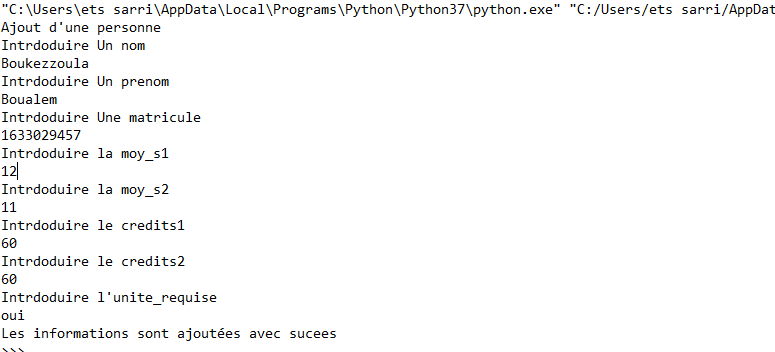
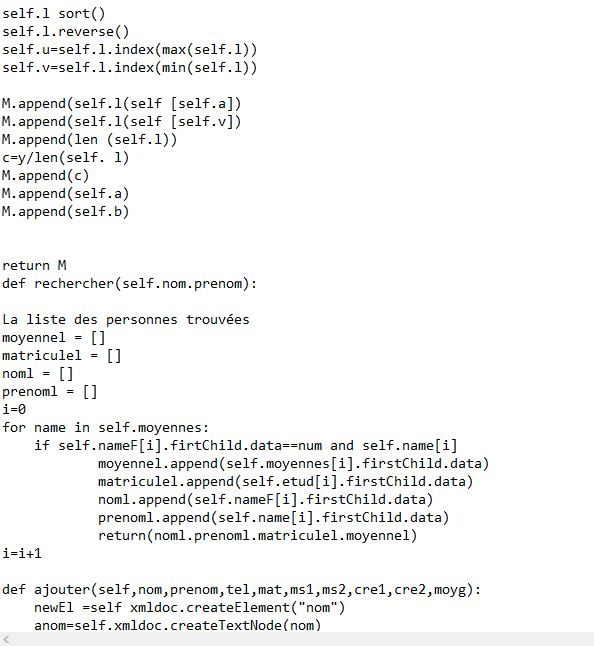


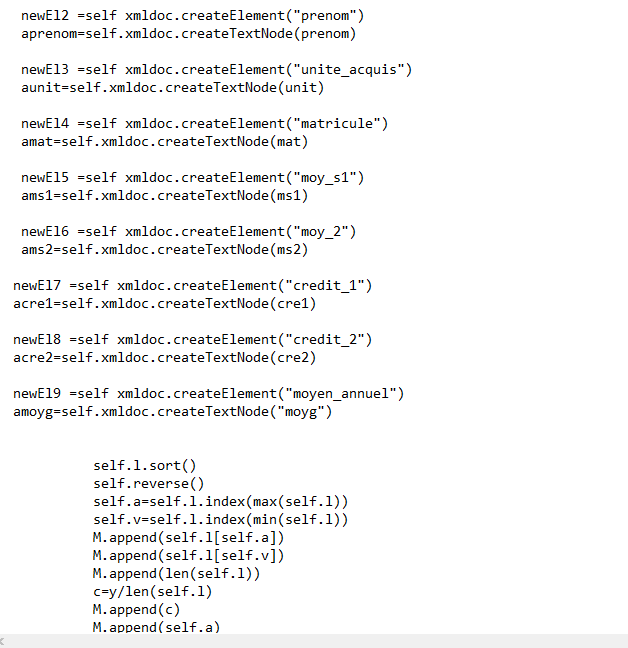
Figure 4 :l’ajout d’un élément dans l’XML

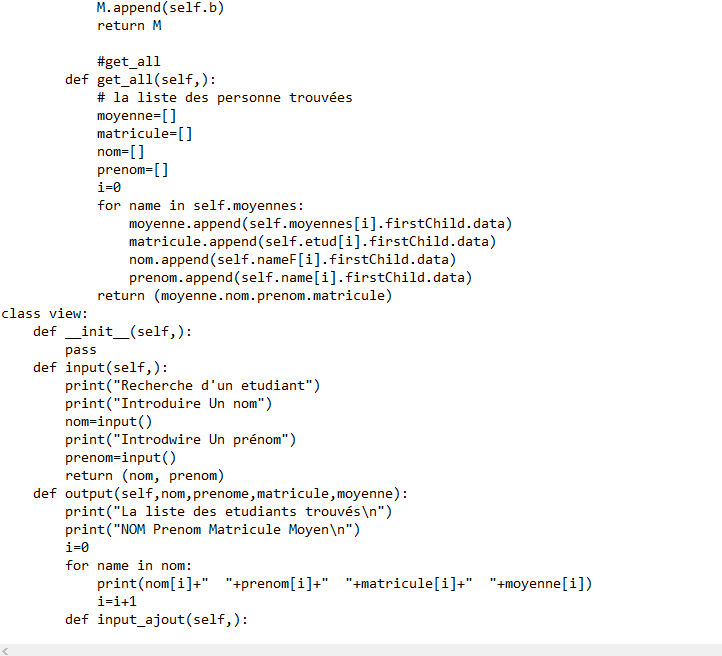
Annexe :

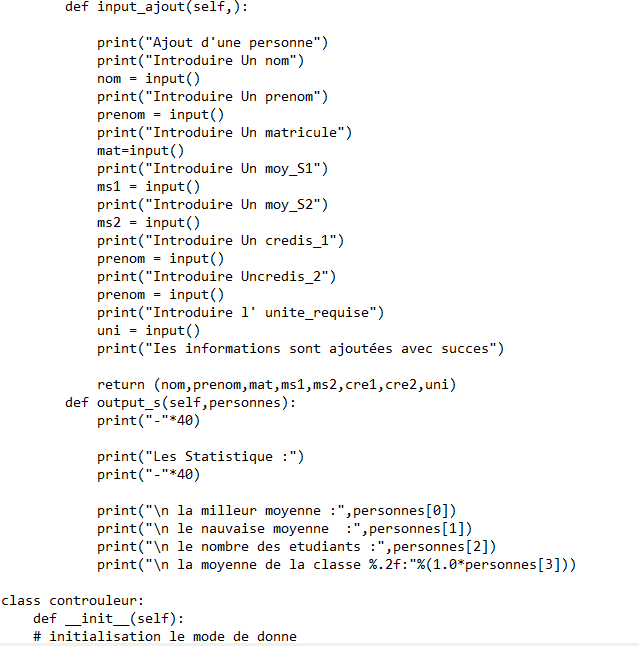
**Parti 1 :**

****

****

****

****

****

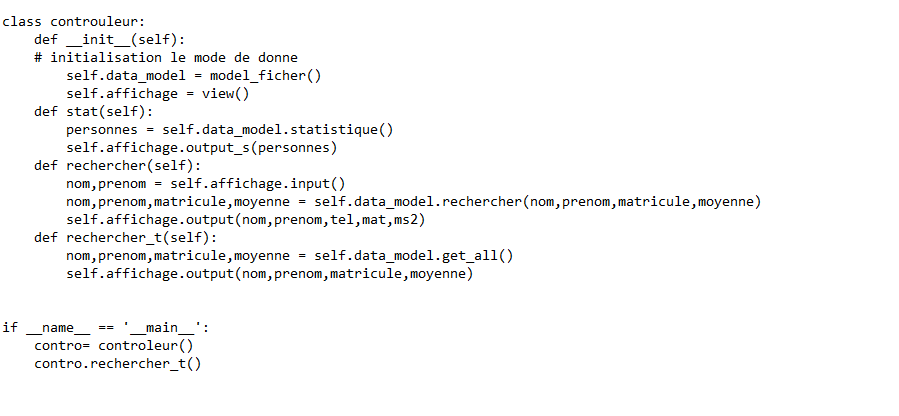
****

Figure – code source MVC