



République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université AMO de Bouira

Faculté des Sciences et des Sciences Appliquées

Département d'Informatique



Compte rendu de TP 5

Méthodes et Technologies d'Implémentation

Spécialité : Génie des Systèmes Informatiques

Modèle Vue Controleur.

Réalisé par

— AZEDDINE MOULAI

— FAROUK LAMRI

2022/2023

0.1 Séance TP :

0.1.1 Introduction

Le but de ce TP est d'apprendre une nouvelle approche qui intervient dans la réalisation des applications, en séparant les données de l'affichage et des actions, c'est l'approche de Modèle Vue Contrôleur.

0.1.2 Méthodologie du travail

Dans cette séance TP, nous avons exécuter un programme donné, qui permet la recherche d'un numéro dans un tableau par son nom, et l'afficher, en premier temps. Puis reformuler le programme en séparant les données de l'affichage. Ajouter une fonctionnalité a ce dernier et enfin changer le modèle de données en un fichier texte puis créer une interface graphique pour notre programme.

0.1.3 Outils utilisés

- **PyCharm.**
- **Language Python.**

0.1.4 Tests et résultats attendus

```

Recherche d'un telephone
Introduire Un nom
Mahdi
Mahdi Ahmed 077878
Mahdi Mounir 0556644

Process finished with exit code 0
|

```

FIGURE 1 – Partie 1.

```

Recherche d'un telephone
Introduire Un nom
Mahdi
La liste des noms trouvés
2 personnes trouvées
Mahdi Ahmed 0778787887
Mahdi Mohamed 0778787887

Process finished with exit code 0

```

FIGURE 2 – Partie 2.

```

Recherche d'un telephone
Introduire Un nom
Mahdi
La liste des noms trouvés
2 personnes trouvées
Mahdi Ahmed 0778787887
Mahdi Mohamed 0778787887
La liste des tous les noms :
Mahdi Ahmed 0778787887
Mahdi Mohamed 0778787887
Katibi Mounir 0778787887
Brahimi Noui 0778787887

Process finished with exit code 0

```

FIGURE 3 – Partie 2 afficher toutes les personnes.

```

Ajout d'une personne
Introduire Un nom
Lamri
Introduire Un prenom
farouk
Introduire Un tel
09876554
La liste des noms trouvés
5 personnes trouvées
Mahdi Ahmed 0778787887
Mahdi Mohamed 0778787887
Katibi Mounir 0778787887
Brahimi Noui 0778787887
Lamri farouk 09876554

```

FIGURE 4 – Partie 3 ajouter des personnes.

Annuaire

Nom: Prenom: Tel:

farouk lamri 987654323456

Search Add Get All

Output

```
[{'nom': 'moulai', 'prenom': '0791950153', 'tel': '0791950153'}, {'nom': 'taha', 'prenom': '0791950153', 'tel': '0791950153'}, {'nom': 'azree', 'prenom': 'azzz', 'tel': 'azzz'}, {'nom': '', 'prenom': '', 'tel': ''}, {'nom': '', 'prenom': '', 'tel': ''}, {'nom': '', 'prenom': '', 'tel': ''}, {'nom': '', 'prenom': '', 'tel': ''}, {'nom': '', 'prenom': '', 'tel': ''}, {'nom': 'farouk', 'prenom': '987654323456', 'tel': '987654323456'}]
```

FIGURE 5 – Interface graphique pour le programme.

.1 Travail à domicile

.1.1 Introduction

Le but de ce travail est d'écrire un programme qui permet d'extraire et exploiter les informations contenues dans le document XML, résultant de TP 1 mais en utilisant la méthodologie MVC.

.1.2 Méthodologie du travail

Ce programme est réalisé déjà dans les TP précédents donc maintenant en va juste le améliorer en utilisant la méthodologie MVC.

.1.3 Outils utilisés

1. *PyCharm*.

.1.4 Tests et résultats attendus

```

+-----+
| La liste des Étudiants: |
+-----+

```

```

-----
niveau: L1
specialite: MI
groupe: g4
matricule: L220121
Nom: ZIANE
Prenom: BILAL
nb_seances: 10
nb_presence: 4
part: 0.5
test1: 6
test2: 1.5
exam: 5
rattrapage: 5.25
CC: 8.0
moyenne: 6.2
moyenne_ratt: 6.35
resulta: exclu
-----

```

```

+-----+
| La liste des statistique: |
+-----+

```

Admis session1	Admis session2	Ajourner	exclus
31	10	9	43

FIGURE 6 – Les resultats du travail a domicile.

Annexe A

Code en annexe

A.1 Séance TP Partie 1

```
1 annuaire = [  
2     {'prenom': 'Ahmed', 'nom': 'Mahdi', 'tel': '077878'},  
3     {'prenom': 'Mohamed', 'nom': 'Rabehi', 'tel': '06678'},  
4     {'prenom': 'Mounir', 'nom': 'Mahdi', 'tel': '0556644'},  
5     {'prenom': 'Noui', 'nom': 'Brahimi', 'tel': '067879'},  
6 ]  
7  
8  
9 def main():  
10     # Ce programme permet de recherche le num dans un tableau par son nom et afficher  
11     # lire des donnees à parti du clavier.  
12     print("Recherche d'un telephone")  
13     print("Introduire Un nom")  
14     nom = input()  
15     # nombre d'elements trouves  
16     nb_found = 0  
17     # parcours des noms  
18     for personne in annuaire:  
19         # afficher toutes les personnes qui ont le nom donne  
20         if personne['nom'] == nom:  
21             print(nom, personne['prenom'], personne['tel'])  
22             nb_found += 1  
23     if not nb_found:  
24         print("ce nom %s n'existe pas " % nom)  
25  
26
```

```
27 if __name__ == '__main__':
28     main()
29
```

A.2 Séance TP Partie 2

```
1 class model:
2     def __init__(self):
3         self.annuaire = [
4             {'prenom': 'Ahmed', 'nom': 'Mahdi', 'tel': '0778787887'},
5             {'prenom': 'Mohamed', 'nom': 'Mahdi', 'tel': '0778787887'},
6             {'prenom': 'Mounir', 'nom': 'Katibi', 'tel': '0778787887'},
7             {'prenom': 'Noui', 'nom': 'Brahimi', 'tel': '0778787887'},
8         ]
9
10    def rechercher(self, nom):
11        """ Recherche un tel par nom """
12        # La liste des personnes trouvees
13
14        personnes = []
15        for persn in self.annuaire:
16            # afficher toutes les personnes qui ont le nom donne
17            if persn['nom'] == nom:
18                personnes.append(persn)
19            # retourner une liste de dictionnaires
20        return personnes
21
22    def get_all(self):
23        """ rechercher un tel par nom """
24        # La liste des personnes trouvees
25
26        personnes = []
27        for persn in self.annuaire:
28            # afficher toutes les personnes
29            personnes.append(persn)
30            # un tableau avec des noms des champs
31        return personnes
32
33
34 class view:
35     def __init__(self, ):
```

```

36         pass
37
38     def input(self, ):
39         """ recuperer le nom a rechercher """
40         print("Recherche d'un telephone")
41         print("Introduire Un nom")
42         nom = input()
43         return nom
44
45     def output(self, personnes):
46         """ Afficher les informations d'une liste des personnes """
47         print("La liste des noms trouves")
48         print(" %d personnes trouvees" % len(personnes))
49         for pers in personnes:
50             print(pers['nom'], pers['prenom'], pers['tel'])
51
52     def out_all(self, personnes):
53         """ Afficher les informations d'une liste des personnes """
54         print("La liste des tous les noms :")
55         for pers in personnes:
56             print(pers['nom'], pers['prenom'], pers['tel'])
57
58
59 def controleur():
60     # Ce programme permet de recherche le num dans un tableau par son nom et afficher
61     # lire des donnees à partir du clavier.
62     data_model = model()
63     affichage = view()
64     """ rechercher un nom """
65     nom = affichage.input()
66     personnes = data_model.rechercher(nom)
67     affichage.output(personnes)
68     personnes = data_model.get_all()
69     affichage.out_all(personnes)
70
71
72 if __name__ == '__main__':
73     controleur()
74

```

A.3 Séance TP Partie 3

```
1 class model:
2     def __init__(self):
3         self.annuaire = [
4             {'prenom': 'Ahmed', 'nom': 'Mahdi', 'tel': '0778787887'},
5             {'prenom': 'Mohamed', 'nom': 'Mahdi', 'tel': '0778787887'},
6             {'prenom': 'Mounir', 'nom': 'Katibi', 'tel': '0778787887'},
7             {'prenom': 'Noui', 'nom': 'Brahimi', 'tel': '0778787887'},
8         ]
9
10    def ajouter(self, nom, prenom, tel):
11        """ ajouter une personne """
12        personne = {}
13        personne["nom"] = nom
14        personne["prenom"] = prenom
15        personne["tel"] = tel
16        self.annuaire.append(personne)
17
18    def rechercher(self, nom):
19        """ Recherche un tel par nom """
20        # La liste des personnes trouvees
21
22        personnes = []
23        for persn in self.annuaire:
24            # afficher toutes les personnes qui ont le nom donne
25            if persn['nom'] == nom:
26                personnes.append(persn)
27            # retourner une liste de dictionnaires
28        return personnes
29
30    def get_all(self):
31        """ rechercher un tel par nom """
32        # La liste des personnes trouvees
33
34        personnes = []
35        for persn in self.annuaire:
36            # afficher toutes les personnes
37            personnes.append(persn)
38            # un tableau avec des noms des champs
39        return personnes
40
```

```

41
42 class view:
43     def __init__(self, ):
44         pass
45
46     def input(self, ):
47         """ recuperer le nom a rechercher """
48         print("Recherche d'un telephone")
49         print("Introduire Un nom")
50         nom = input()
51         return nom
52
53     def output(self, personnes):
54         """ Afficher les informations d'une liste des personnes """
55         print("La liste des noms trouves")
56         print(" %d personnes trouvees" % len(personnes))
57         for pers in personnes:
58             print(pers['nom'], pers['prenom'], pers['tel'])
59
60     def out_all(self, personnes):
61         """ Afficher les informations d'une liste des personnes """
62         print("La liste des tous les noms :")
63         for pers in personnes:
64             print(pers['nom'], pers['prenom'], pers['tel'])
65
66     def input_ajout(self, ):
67         """ recuperer le nom a ajouter """
68         print("Ajout d'une personne")
69         print("Introduire Un nom")
70         nom = input()
71         print("Introduire Un prenom")
72         prenom = input()
73         print("Introduire Un tel")
74         tel = input()
75         return nom, prenom, tel
76
77
78 class controleur:
79     def __init__(self, ):
80         """ constructeur du controleur """
81         # initialiser le model de donnees
82         # ~ self.data_model = model()

```

```

83         self.data_model = model()
84         self.affichage = view()
85
86     def rechercher(self):
87         """ rechercher un nom """
88         nom = self.affichage.input()
89         personnes = self.data_model.rechercher(nom)
90         self.affichage.output(personnes)
91
92     def ajouter(self):
93         """ rechercher un nom """
94         nom, prenom, tel = self.affichage.input_ajout()
95         self.data_model.ajouter(nom, prenom, tel)
96         personnes = self.data_model.get_all()
97         self.affichage.output(personnes)
98         tel = input()
99         return (nom, prenom, tel)
100
101
102 if __name__ == '__main__':
103     contro = controleur()
104     contro.ajouter()
105
106

```

A.4 Séance TP Partie 4

```

1  import sys
2
3
4  class model_fichier:
5      """ Classe de model de donnee """
6
7      def __init__(self):
8          self.annuaire = []
9          try:
10             myfile = open("annuaire.txt", "r")
11         except:
12             print("Can't open DataFile")
13             sys.exit()
14         lines = myfile.readlines()

```

```

15         for line in lines:
16             line = line.strip('\n')
17             fields = line.split('\t')
18             personne = {"nom": fields[0], "prenom": fields[1], "tel": fields[2]}
19             self.annuaire.append(personne)
20         myfile.close()
21
22     def __del__(self, ):
23         """ Destructeur de la classe Modele,
24         Il est appele a la fin de programme
25         recharger les donnees dans le fichier """
26         try:
27             myfile = open("annuaire.txt", "w")
28         except:
29             print("Can't open DataFile")
30             sys.exit()
31         for personne in self.annuaire:
32             line = personne['nom'] + '\t' + personne['prenom'] + '\t' + personne['tel'] + "\n"
33             myfile.write(line)
34         myfile.close()
35
36     def rechercher(self, nom):
37         """ Recherche un tel par nom"""
38         # La liste des personnes trouvees
39
40         personnes = []
41         for persn in self.annuaire:
42             # afficher toutes les personnes qui ont le nom donne
43             if persn['nom'] == nom:
44                 personnes.append(persn)
45             # retourner une liste de dictionnaires
46         return personnes
47
48     def get_all(self):
49         """ rechercher un tel par nom"""
50         # La liste des personnes trouvees
51
52         personnes = []
53         for persn in self.annuaire:
54             # afficher toutes les personnes
55             personnes.append(persn)
56         # un tableau avec des noms des champs

```

```

57         return personnes
58
59     def ajouter(self, nom, prenom, tel):
60         """ ajouter une personne """
61         personne = {}
62         personne["nom"] = nom
63         personne["prenom"] = prenom
64         personne["tel"] = tel
65         self.annuaire.append(personne)
66
67
68     def main():
69         data_model = model_fichier()
70         personnes = data_model.get_all()
71         for person in personnes:
72             print(person)
73
74
75 if __name__ == "__main__":
76     main()
77

```

A.5 Séance TP Partie 5

```

1  from tkinter import Tk, Label, Button, Entry, IntVar, END, W, E, Text, Listbox, OptionMenu
2  from tkinter import StringVar
3  from model_fichier import model_fichier
4
5
6  class view_tk:
7      def __init__(self, model):
8          master = Tk()
9          self.master = master
10         self.model = model
11         master.title("Annuaire")
12         self.entered_number = 0
13         # ~ master.withdraw()
14         master.clipboard_clear()
15         master.clipboard_append('i can has clipboardz?')
16         self.output_label = Label(master, text="Output")
17         self.label = Label(master, text="Nom:")

```

```

18     self.entry = Text(master, height=1, width=50)
19     self.label_prenom = Label(master, text="Prenom:")
20     self.entry_prenom = Text(master, height=1, width=50)
21     self.label_tel = Label(master, text="Tel:")
22     self.entry_tel = Text(master, height=1, width=50)
23     self.output = Text(master, height=10, width=50)
24     self.search_button = Button(master, text="Search", command=lambda:
↳     self.update("search"))
25     self.add_button = Button(master, text="Add", command=lambda: self.update("add"))
26     self.all_button = Button(master, text="Get All", command=lambda: self.update("get_all"))
27     # LAYOUT
28     self.label.grid(row=0, column=0, sticky=W)
29     self.label_prenom.grid(row=0, column=1, sticky=W)
30     self.label_tel.grid(row=0, column=2, sticky=W)
31     # input nom
32     self.entry.grid(row=1, column=0, columnspan=10, sticky=W + E)
33     # prenom et tel
34     self.entry_prenom.grid(row=1, column=1, columnspan=10, sticky=W + E)
35     self.entry_tel.grid(row=1, column=2, columnspan=2, sticky=W + E)
36     # command
37     self.search_button.grid(row=2, column=0)
38     self.add_button.grid(row=2, column=1)
39     self.all_button.grid(row=2, column=2)
40     # Output
41     self.output_label.grid(row=3, column=0, sticky=E)
42     # output
43     self.output.grid(row=4, column=0, columnspan=5, sticky=W + E)
44
45     def update(self, method):
46         nom = self.entry.get("1.0", END).strip()
47         prenom = self.entry_prenom.get("1.0", END)
48         tel = self.entry_tel.get("1.0", END)
49         # clean ends of words
50         nom = nom.strip()
51         prenom = prenom.strip()
52         tel = tel.strip()
53         if method == "search":
54             result = self.search(nom)
55         elif method == "add":
56             result = self.add(nom, prenom, tel)
57         elif method == "get_all":
58             result = self.get_all()

```

```

59         else: # reset
60             print("method", method)
61             result = nom
62             self.output.insert(END, str(result))
63
64     def add(self, nom, prenom, tel):
65         return self.model.ajouter(nom, prenom, tel)
66
67     def search(self, nom):
68         result = self.model.rechercher(nom)
69         if result:
70             return result
71         else:
72             return "%s" % nom not found
73
74     def get_all(self):
75         return self.model.get_all()
76
77     def run(self):
78         self.master.mainloop()
79
80
81 def controleur3():
82     data_model = model_fichier()
83     affichage = view_tk(data_model)
84     affichage.run()
85
86
87 if __name__ == '__main__':
88     controleur3()
89

```

A.6 Travail à domicile

```

1 import xml.dom.minidom as minidom
2 import sys
3
4
5 class model:
6
7     def __init__(self):

```

```

8         self.DATA_FILE = 'suivi_etudiant.xml'
9     try:
10         self.xmldoc = minidom.parse(self.DATA_FILE)
11     except:
12         print("Can't Open the file")
13         sys.exit()
14
15     def get_value(self, element):
16         values = self.xmldoc.getElementsByTagName(element)
17         value_list = []
18         for value in values:
19             try:
20                 val = float(value.firstChild.data)
21             except:
22                 val = " "
23             value_list.append(val)
24         return value_list
25
26     def add_ele(self, etudiants, doc_root, obj_list, ele_name):
27         for etudiant, ele in zip(etudiants, obj_list):
28             element = self.xmldoc.createElement(ele_name)
29             element.appendChild(self.xmldoc.createTextNode(str(ele)))
30             etudiant.appendChild(element)
31             doc_root.appendChild(etudiant)
32         return doc_root
33
34     def update_xmldoc(self, calcul_list, name_list):
35         etudiants = self.xmldoc.getElementsByTagName("etudiant")
36         doc_root = self.xmldoc.createElement("suivietudiant")
37         for calc, name in zip(calcul_list, name_list):
38             self.add_ele(etudiants, doc_root, calc, name)
39         self.xmldoc = doc_root
40
41
42     class view:
43         def __init__(self):
44             pass
45
46         def affichier_liste_etudiants(self, xmldoc):
47             etudiants = xmldoc.getElementsByTagName("etudiant")
48
49             print("\t", "+-----+")

```



```

50     print("\t", "| La liste des Étudiants:      |")
51     print("\t", "+-----+")
52     print()
53     for etudiant in etudiants:
54         print("-"*40)
55         print("\t\tniveau: ", etudiant.getElementsByTagName("niveau")[0].firstChild.data)
56         print("\t\tspecialite: ",
57             ↳ etudiant.getElementsByTagName("specialite")[0].firstChild.data)
58         print("\t\tgroupe: ", etudiant.getElementsByTagName("groupe")[0].firstChild.data)
59         print("\t\tmatricule: ",
60             ↳ etudiant.getElementsByTagName("matricule")[0].firstChild.data)
61         print("\t\tNom: ", etudiant.getElementsByTagName("nom")[0].firstChild.data)
62         print("\t\tPrenom: ", etudiant.getElementsByTagName("prenom")[0].firstChild.data)
63         print("\t\tnb_seances: ",
64             ↳ etudiant.getElementsByTagName("nb_seances")[0].firstChild.data)
65         print("\t\tnb_presence: ",
66             ↳ etudiant.getElementsByTagName("nb_presence")[0].firstChild.data)
67         print("\t\tpart: ", etudiant.getElementsByTagName("part")[0].firstChild.data)
68         print("\t\ttest1: ", etudiant.getElementsByTagName("test1")[0].firstChild.data)
69         print("\t\ttest2: ", etudiant.getElementsByTagName("test2")[0].firstChild.data)
70         print("\t\texam: ", etudiant.getElementsByTagName("exam")[0].firstChild.data)
71         print("\t\ttrattapage: ",
72             ↳ etudiant.getElementsByTagName("rattapage")[0].firstChild.data)
73         print("\t\tCC: ", etudiant.getElementsByTagName("CC")[0].firstChild.data)
74         print("\t\tmoyenne: ", etudiant.getElementsByTagName("moyenne")[0].firstChild.data)
75         print("\t\tmoyenne_ratt: ",
76             ↳ etudiant.getElementsByTagName("moyenne_ratt")[0].firstChild.data)
77         print("\t\tresulta: ", etudiant.getElementsByTagName("resultat")[0].firstChild.data)
78         print("-"*40)
79
80     def affichier_les_statistiques(self, xmldoc):
81         ads1 = xmldoc.getElementsByTagName("damis_S1")[0]
82         ads2 = xmldoc.getElementsByTagName("damis_S2")[0]
83         ajr = xmldoc.getElementsByTagName("Ajournner")[0]
84         ex1 = xmldoc.getElementsByTagName("exclu")[0]
85         print("\t" * 8, "+-----+")
86         print("\t" * 8, "| La liste des statistique:  |")
87         print("\t" * 8, "+-----+")
88         print()
89         print("\t\t", "+" + "-----+" * 4)

```

```

85     print("\t\t | Admis session1 | Admis session2 | Ajourner | exclus
      ↪ |")
86     print("\t\t", "+" + "-----+" * 4)
87     print("\t\t | \t", ads1.firstChild.data, "\t | \t\t", ads2.firstChild.data, "\t
      ↪ | \t\t",
88         ajr.firstChild.data, "\t\t | \t\t", exl.firstChild.data, "\t |")
89     print("\t\t", "+" + "-----+" * 4)
90
91
92 class controller:
93     def __init__(self):
94         self.data_model = model()
95         self.affichage = view()
96
97     def calculate_cc(self):
98         nb_seances_list = self.data_model.get_value("nb_seances")
99         nb_presence_list = self.data_model.get_value("nb_presence")
100        part_list = self.data_model.get_value("part")
101        test1_list = self.data_model.get_value("test1")
102        test2_list = self.data_model.get_value("test2")
103
104        nb_abs_liste = []
105        for nb_seances, nb_presence in zip(nb_seances_list, nb_presence_list):
106            nb_abs = nb_seances - nb_presence
107            if nb_abs >= 3:
108                note_abs = 0
109            else:
110                note_abs = 3 - nb_abs
111            nb_abs_liste.append(note_abs)
112        cc_notes_liste = []
113        for abs, part, test1, test2 in zip(nb_abs_liste, part_list, test1_list, test2_list):
114            cc_note = abs + part + test1 + test2
115            cc_notes_liste.append(cc_note)
116        return cc_notes_liste, nb_abs_liste
117
118    def calculate_moy(self, cc_notes_listes):
119        exam_notes_list = self.data_model.get_value("exam")
120        moy_liste = []
121        for cc, exam in zip(cc_notes_listes, exam_notes_list):
122            moy = float(format((cc * 0.4) + (exam * 0.6), ".2f"))
123            moy_liste.append(moy)
124        return moy_liste

```

```

125
126 def calculate_ratt(self, cc_notes_liste, moy_notes_liste):
127     ratt_notes_list = self.data_model.get_value("rattrapage")
128     moy_ratt_liste = []
129     for cc, moy, ratt in zip(cc_notes_liste, moy_notes_liste, ratt_notes_list):
130         if ratt == "":
131             moy_ratt_liste.append("")
132         elif ratt != "" and moy < 10:
133             moy_ratt = float(format((cc * 0.4) + (ratt * 0.6), ".2f"))
134             moy_ratt_liste.append(moy_ratt)
135     return moy_ratt_liste
136
137 def calculate_result_and_stat(self, nb_abs_liste, moy_notees_liste, ratt_notes_liste):
138     global resulta
139     resulta_liste = []
140     s1_cpt, s2_cpt, aj_cpt, ex_cpt = 0, 0, 0, 0
141     for abs, moy, ratt in zip(nb_abs_liste, moy_notees_liste, ratt_notes_liste):
142         if abs == 0:
143             resulta_liste.append("exclu")
144             ex_cpt += 1
145         elif moy >= 10:
146             resulta_liste.append("Admis S1")
147             s1_cpt += 1
148         elif ratt != "" and ratt >= 10:
149             resulta_liste.append("Admis S2")
150             s2_cpt += 1
151         else:
152             resulta_liste.append("Ajourner")
153             aj_cpt += 1
154     statistiques = [s1_cpt, s2_cpt, aj_cpt, ex_cpt]
155     statistiques_names = ["damis_S1", "damis_S2", "Ajourner", "exclu"]
156     resulta = self.data_model.xmlDoc.createElement("resulta")
157     for stat, ele in zip(statistiques, statistiques_names):
158         element = self.data_model.xmlDoc.createElement(ele)
159         element.appendChild(self.data_model.xmlDoc.createTextNode(str(stat)))
160         resulta.appendChild(element)
161
162     return resulta_liste, resulta
163
164 def run(self):
165     cc_notes_liste, nb_abs_liste = self.calculate_cc()
166     moy_notes_liste = self.calculate_moy(cc_notes_liste)

```

```
167         ratt_notes_liste = self.calculate_ratt(cc_notes_liste, moy_notes_liste)
168         resulta_liste, statistiques = self.calculate_result_and_stat(nb_abs_liste,
    ↪     moy_notes_liste,
169                                     ratt_notes_liste)
170
171         calcul_list = [cc_notes_liste, moy_notes_liste, ratt_notes_liste, resulta_liste]
172         name_list = ["CC", "moyenne", "moyenne_ratt", "resultat"]
173         self.data_model.update_xmldoc(calcul_list, name_list)
174         self.affichage.affichier_liste_etudiants(self.data_model.xmldoc)
175         self.affichage.affichier_les_statistiques(statistiques)
176
177
178 if __name__ == '__main__':
179     ctrl = controller()
180     ctrl.run()
181
182
```
