import xml.etree.ElementTree as ET

class student():

    Surname = str()

    Math = int()

    Infa = int()

    RusLang = int()

    def \_\_init\_\_(self, IDS):

        tree = ET.parse('C:/PyProgs/Lab3/tsst.xml') #Парсим

        root = tree.getroot() #Присваеваем Адрес

        STRFORM = "Students/Student[@id='{0}']".format(IDS) #Форматируем строку для поиска

        for student in root.findall(STRFORM): #Ищем

            self.Surname = student.find('Surname').text

            for exam in student.findall("Exams"): #Ищем в подкатологе Exams

                self.Math = int(exam.find('Math').text)

                self.Infa = int(exam.find('Inform').text)

                self.RusLang = int(exam.find('RusLang').text)

    def MidScore(self):

        return((self.Math + self.Infa + self.RusLang)//3)

    def MaxScore(self):

        List = [self.Math, self.Infa, self.RusLang]

        return (max(List))

    def MinScore(self):

        List = [self.Math, self.Infa, self.RusLang]

        return (min(List))

    def Summary(self):

        return int(self.Math + self.Infa + self.RusLang)

Stud1 = student(1)

Stud2 = student(2)

Stud3 = student(3)

Stud0 = [Stud1, Stud2, Stud3]

for i in range(0,2):

    print(Stud0[i].Surname,':')

    print('Средний Балл: ', Stud0[i].MidScore())

    print('Макс. Балл: ', Stud0[i].MaxScore())

    print('Мин. Балл: ', Stud0[i].MinScore())

print('Абитуриенты with 250+ баллов:')

for i in range(0,2):

    if Stud0[i].Summary() > 250:

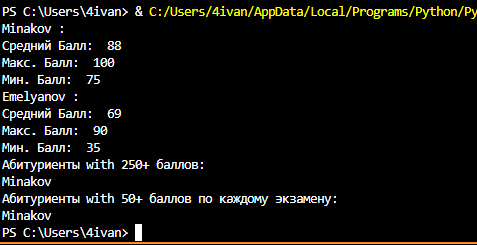
        print(Stud0[i].Surname)

print('Абитуриенты with 50+ баллов по каждому экзамену:')

for i in range(0,2):

    if (Stud0[i].Math > 50) and (Stud0[i].RusLang > 50) and (Stud0[i].Infa > 50):

        print(Stud0[i].Surname)



import xml.etree.ElementTree as ET

class SubjectClass():

    def \_\_init\_\_(self, name, Tasks, TD):

        self.Name = name

        self.Task = int(Tasks)

        self.TaskDone = int(TD)

class student():

    Surname = str()

    Midame = str()

    ID = str()

    SubjectList = []

    def \_\_init\_\_(self, name, surname, midname, ID):

        self.Name = name

        self.Surname = surname

        self.Midame = midname

        self.ID = ID

        self.SubjectList = []

class group():

    def \_\_init\_\_(self, Name):

        self.Name = Name

        self.GroupList = []

tree = ET.parse('C:/PyProgs/Lab3/tsst2.xml')

root = tree.getroot()

SuperList = [group(Groups.get("GNum")) for Groups in root.findall("Groups/Group")]

for GroupObj in SuperList:

    Formattext = "Groups/Group[@GNum='{0}']/Student".format(GroupObj.Name)

    GroupObj.GroupList = [student(Students.get("name"), Students.get("surname"), Students.get("midname"), Students.get("ID")) for Students in root.findall(Formattext)]

    for StudObj in GroupObj.GroupList:

        Formattext2 = "Groups/Group[@GNum='{0}']/Student[@ID='{1}']/Subject".format(GroupObj.Name, StudObj.ID)

        StudObj.SubjectList = [SubjectClass(Subjects.get('SName'), Subjects.find("Tasks").text, Subjects.find("Tasks\_Done").text) for Subjects in root.findall(Formattext2)]

print('2.1')

for GroupObj in SuperList:

    print(GroupObj.Name,': ')

    for Studobj in GroupObj.GroupList:

        print(Studobj.Surname, Studobj.Name, Studobj.Midame)

print('2.2, 2.3, 2.4')

print(SuperList[0].GroupList[2].Surname)

TaskAll = 0

TaskDoneAll = 0

for SubjObj in SuperList[0].GroupList[2].SubjectList:

   PercentSub = SubjObj.TaskDone/SubjObj.Task\*100

   print(SubjObj.Name,': ', SubjObj.Task,'работ сделано', PercentSub,'Процент готовности по предмету')

   TaskAll = TaskAll + SubjObj.Task

   TaskDoneAll = TaskDoneAll + SubjObj.TaskDone

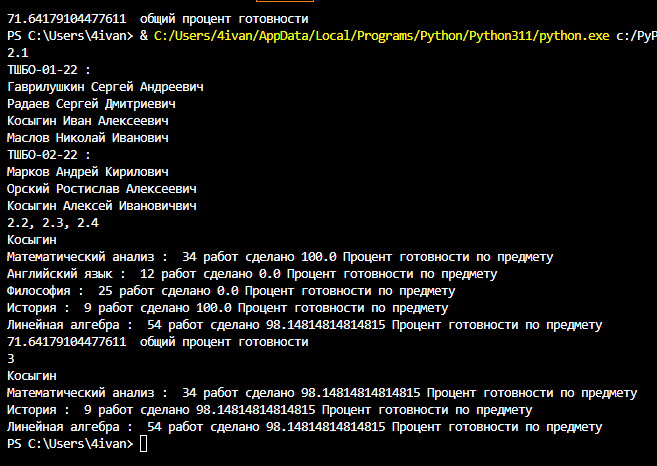
print(TaskDoneAll/TaskAll\*100,' общий процент готовности')

print(3)

print(SuperList[0].GroupList[2].Surname)

for SubjObj in SuperList[0].GroupList[2].SubjectList:

   if SubjObj.TaskDone/SubjObj.Task\*100 > 50 :

    print(SubjObj.Name,': ', SubjObj.Task,'работ сделано', PercentSub,'Процент готовности по предмету')