```python
class Personne() :
    # # attribut de classe
    nb_eleves = 0
    def __init__(self, nom, prenom,section,matieres = [],notes = {}):
        self.nom = nom
        self.prenom = prenom
        self.section = section
        self.matieres = matieres
        self.notes = notes
        self.__class__.nb_eleves+=1 # modif  attribut de classe

    def set_matieres(self):
        """
        Un dictionnaire de matières constitue les notes
        Permet le choix des spécialités
        Initialise le dictionnaire des notes pour les spécialités vides

        """
        """
        choix des spécialités
        """
        print(f"Choix des matières pour ", self.nom," ",self.prenom )
        specialites = ["maths","physique","NSI","SVT","HLP"]
        for elt in specialites :
            print(" Notes en ",elt)
            reponse = input("repondre oui / non ")
            if reponse  == "oui" :
                self.matieres.append(elt)
            else:
                self.matieres.append(None)

        """
        Initialise les dictionnaires des notes
        """

        for elt in self.matieres :
            if elt != None :
                self.notes[elt] = []


    def get_matieres(self):
        print("les spécialité choisies par ", self.nom," ", self.prenom," sont
:",end = " ")
        for elt in self.matieres :
            if elt != None:
                print(elt," ",end = " ")

        print()
        # print(self.notes)
        return self.matieres

    def set_notes(self):
        for cle in self.notes.keys():
            ajout = "oui"
```

```
            while ajout == "oui" :
                print("Voulez vous ajouter une note en ", cle)
                ajout = input( " oui /non ")
                if ajout  == "oui" :
                    note = float(input("entrer votre note "))
                    self.notes[cle].append(note)

    def get_notes(self) :
        return self.notes

    def get_moyennes(self):
        for cle in self.notes.keys() :
            if cle != None :
                if len(self.notes[cle]) !=0 :
                    som = 0
                    nb_notes  = len(self.notes[cle])
                    for elt in self.notes[cle] :
                        som = som + elt
                    moy  = som/nb_notes
                    print("la moyenne en ",cle," vaut ", moy)


    def __str__(self):
        return(f"{self.nom} {self.prenom} {self.section} {self.matieres}
{self.notes}")


class Eleve(Personne) :

    def __init__(self,nom,prenom,section,matieres,notes,qualite = ""):
        super().__init__(nom,prenom,section,matieres,notes)
        self.qualite  = qualite
    def oui(self) :
        print("oui")

    def regime(self):
        print(self.nom," ",self.prenom, "est - il pentionnaire ?")
        rep = input("oui / non ")
        if rep == "oui":
            self.qualite = "DP"
        else :
            self.qualite = "EXT"

    def __str__(self) :
        return( f" {Personne.__str__(self)} {self.qualite} ")




print(Personne.nb_eleves)
Personne1=Personne("Triboulet","Thomas","tg1")
print(Personne.nb_eleves)
Personne1.get_matieres()
Personne1.set_matieres()
Personne1.get_matieres()
Personne1.set_notes()
```

```
Personne1.get_notes()
Personne1.get_moyennes()

Personne2 =Personne("GROS","Louis","tg2")
Personne2.matieres = ['maths', 'physique', None, None, None]
Personne2.notes = {'maths': [14.0, 18.0], 'physique': [12.5]}
print(Personne2)
Eleve2= Eleve("GROS","Louis","tg2",['maths', 'physique', None, None,
None],{'maths': [14.0, 18.0], 'physique': [12.5]})
Eleve
# print(Eleve2.get_matieres())
# Eleve2.regime()
print(Eleve2.get_notes())
# print(Eleve2)
# print(Eleve2.notes['maths'])
```