

```

employee.py
class employee(object):
    """description of class"""
    did=0
    def __init__(self,second_name="",first_name="",patronymic="",pos="",zp=0,d_id=-1):
        self.name=[second_name,first_name,patronymic]
        self.position=pos
        self.profit=zp
        self.id=employee.did
        employee.did+=1
        self.Department_id=d_id
        self.sub_id=[1]
        self.sub_id.pop(0)
    def Set_Sub_Id(self,id):
        self.sub_id.append(id)
    def Get_Name(self):
        nm=""
        for i in self.name:
            nm=nm+" "+i
        return nm
    def Get_Position(self):
        return self.position
    def Get_Profit(self):
        return self.profit

```

Department.py

```

from employee import *
from prettytable import PrettyTable

class Department(object):
    """description of class"""
    did=0
    def __init__(self,ne="Отдел без имени", *emps):
        self.Department_id=Department.did
        Department.did+=1
        self.name=ne
        emp=employee()
        self.employeers=[emp]
        self.employeers.pop(0)
        for i in emps:
            self.employeers.append(i)
            i.Department_id=self.Department_id

    def Set_Name(self,nm):
        self.name=nm

    def Get_Name(self):
        return self.name

    def Add_Employee(self, employee):

        if(employee.Department_id==-1):
            employee.Department_id=self.Department_id
            self.employeers.append(employee)
        else:
            print ("Работник уже привязан, добавим sub-id")
            employee.Set_Sub_Id(self.Department_id)
            self.employeers.append(employee)

    def Get_Employee(self):
        mytable =PrettyTable()
        print(self.Get_Name())
        if (len(self.employeers)!=0):
            mytable.field_names=["id", "Имя", "Должность", "Зарплата(Р)"]

```

```

        #print("id "+str(self.Department_id)+" Сотрудники подразделения
"+self.Get_Name()+": ")
        #print("id:          Имя:          Должность:
Зарплата(P):")
    else:
        #print("id "+Department_id+"Сотрудники подразделения "+self.Get_Name()+": ")
        print("Работников в отделе нет")
        for i in self.employees:
            #print(str(i.id)+" "+i.Get_Name()+" "+i.Get_Position()+"
"+str(i.Get_Profit()))
            mytable.add_row([i.id,i.Get_Name(),i.Get_Position(),i.Get_Profit()])
        print(mytable)

    def Sort_By_Departments_Name(sorter="",*Departments): #Сортировка по наличию слова
sorted в имени отдела
        print ("Сортировка по слову: ",sorter)
        for i in Departments:
            if (any( word in sorter.lower() for word in i.Get_Name().lower().split())):
# Если имя отдела вне зависимости от кодировки начинается со слова Отдел то вывод
i.Get_Employee()

    def Output_By_Median_Profit(*Departments) : #Вывод средней ЗП по
отделу
        print("Средняя зп по отделам")
        for dep in Departments:
            medprofit=0 #Переменная для зарплаты в отделе
            for emp in dep.employees:
                medprofit=medprofit+emp.Get_Profit()
            if(len(dep.employees)!=0): medprofit=medprofit/len(dep.employees)
            medprofit=round(medprofit,2)
            dep.Get_Employee()
            print("Средняя заработная плата по подразделению ",dep.Get_Name()," :
",str(medprofit))
            print("")

    def Output_Employee_In_Departments_Sort_By_Second_Name(check="",*Departments):
mytable =PrettyTable()
        print ("Поиск по букве: ",check)
        mytable.field_names=["Id сотрудника","Сотрудник","Отдел"]
        for dep in Departments:
            for emp in dep.employees:
                if(emp.name[0].lower().startswith(check.lower())):
                    mytable.add_row([emp.id,emp.Get_Name(),dep.Get_Name()])
                    #print ("Сотрудник: ",emp.Get_Name(),"Из отдела:",dep.Get_Name())
        print(mytable)
        return 0

```

PY\_RK\_1.py

#1)«Отдел» и «Сотрудник» связаны соотношением один-ко-многим.  
#Выведите список всех отделов, у которых в названии присутствует слово «отдел», и список работающих в них сотрудников.  
#2)«Отдел» и «Сотрудник» связаны соотношением один-ко-многим.  
# Выведите список отделов со средней зарплатой сотрудников в каждом отделе, отсортированный по средней зарплате.  
# Средняя зарплата должна быть округлена до 2 знака после запятой (отдельной функции вычисления среднего значения в Python нет,  
# нужно использовать комбинацию функций вычисления суммы и количества значений;  
# для округления необходимо использовать функцию  
<https://docs.python.org/3/library/functions.html#round>).  
#3)«Отдел» и «Сотрудник» связаны соотношением многие-ко-многим.  
# Выведите список всех сотрудников, у которых фамилия начинается с буквы «А», и названия их отделов.

```

from employee import *
from Department import *

```

```

def main():

```

```

dep= Department ("Отдел закупок")
#print(dep.Get_Name())
Vasya= employee("Зайцев", "Василий", pos="Фруктов стол вкусный", zp=300000)
Egor= employee("Лохматенко", "Егор", pos="Глава отдела", zp=122000)
Ivan= employee("Пончиков", "Иван", pos="Закупщик", zp=30000)
dep.Add_Employee(Vasya)
dep.Add_Employee(Egor)
dep.Add_Employee(Ivan)
dep.Get_Employee()

dep2= Department("Отдел продаж",
employee(first_name="Иван", second_name="Романов", pos="Закупщик", zp=52145),
employee(first_name="Пётр", second_name="Пономарёв", pos="Перекупщик", zp=15321),
employee(first_name="Иван ", pos="Продажщик", zp=33400))
dep2.Add_Employee(Ivan) # попытка перепривязать работника
dep2.Get_Employee()
dep3= Department("Бухгалтерия",
employee(first_name="Пётр", second_name="Иванов", pos="Глав. Бух. ", zp=64305),
employee(first_name="Пётр", second_name="Иванов", pos="Бухгалтер", zp=19321),
employee(first_name="Иван ", second_name="Петров", pos="Продажщик", zp=33400))
dep3.Get_Employee()
print("")
Department.Sort_By_Departments_Name("Отдел", dep, dep2, dep3)
print("")
Department.Output_By_Median_Profit(dep, dep2)
print("")
Department.Output_Employee_In_Departments_Sort_By_Second_Name("П", dep, dep2, dep3)

if(__name__=="__main__"):
    main()

```

## Результаты

C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python37\_64\python.exe

### Отдел закупок

id	Имя	Должность	Зарплата(Р)
1	Зайцев Василий	Плодов стола вкусилий	300000
2	Лохматенко Егор	Глава отдела	122000
3	Пончиков Иван	Закупщик	30000

Работник уже привязан

### Отдел продаж

id	Имя	Должность	Зарплата(Р)
4	Романов Иван	Закупщик	52145
5	Пономарёв Пётр	Перекупщик	15321
6	Иван	Продажщик	33400

### Бухгалтерия

id	Имя	Должность	Зарплата(Р)
8	Иванов Пётр	Глав.Бух.	64305
9	Иванов Пётр	Бухгалтер	19321
10	Петров Иван	Продажщик	33400

C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python37\_64\python.exe

Сортировка по слову: Отдел

Отдел закупок

id	Имя	Должность	Зарплата(Р)
1	Зайцев Василий	Плодов стола вкусилий	300000
2	Лохматенко Егор	Глава отдела	122000
3	Пончиков Иван	Закупщик	30000

Отдел продаж

id	Имя	Должность	Зарплата(Р)
4	Романов Иван	Закупщик	52145
5	Пономарёв Пётр	Перекупщик	15321
6	Иван	Продажщик	33400

Средняя зп по отделам

Отдел закупок

id	Имя	Должность	Зарплата(Р)
1	Зайцев Василий	Плодов стола вкусилий	300000
2	Лохматенко Егор	Глава отдела	122000
3	Пончиков Иван	Закупщик	30000

Средняя заработная плата по подразделению Отдел закупок : 150666.67

C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python37\_64\python.exe

+-----+-----+-----+-----+  
Средняя заработная плата по подразделению Отдел закупок : 150666.67

Отдел продаж

id	Имя	Должность	Зарплата(Р)
4	Романов Иван	Закупщик	52145
5	Пономарёв Пётр	Перекупщик	15321
6	Иван	Продажник	33400

+-----+-----+-----+-----+  
Средняя заработная плата по подразделению Отдел продаж : 33622.0

Поиск по букве: П

Id сотрудника	Сотрудник	Отдел
3	Пончиков Иван	Отдел закупок
5	Пономарёв Пётр	Отдел продаж
10	Петров Иван	Бухгалтерия

+-----+-----+-----+-----+  
Press any key to continue . . .