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","Заработок(Р)"]

#print("id "+str(self.Department\_id)+" Сотрудники подразделения "+self.Get\_Name()+": ")

#print("id: Имя: Должность: Заработок(P):")

else:

#print("id "+Department\_id+"Сотрудники подразделения "+self.Get\_Name()+": ")

print("Работников в отделе нет")

for i in self.employeers:

#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.employeers:

medprofit=medprofit+emp.Get\_Profit()

if(len(dep.employeers)!=0): medprofit=medprofit/len(dep.employeers)

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.employeers:

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="Плодов стола вкусилий",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





