**Московский государственный технический** **университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по РК№1

Вариант№8 Г

Выполнил: Проверил:

студент группы ИУ5-34Б преподаватель каф. ИУ5

Кожевников М. А. Гапанюк Ю.Е. Подпись и дата: Подпись и дата:

Москва, 2023 г.

Задание:

1) Необходимо создать два класса данных в соответствии с Вашим вариантом предметной области, которые связаны отношениями один-ко-многим и многие-ко-многим.

Пример классов данных для предметной области Сотрудник-Отдел:

1. Класс «Сотрудник», содержащий поля:
   * ID записи о сотруднике;
   * Фамилия сотрудника;
   * Зарплата (количественный признак);
   * ID записи об отделе. (для реализации связи один-ко-многим)
2. Класс «Отдел», содержащий поля:
   * ID записи об отделе;
   * Наименование отдела.
3. (Для реализации связи многие-ко-многим) Класс «Сотрудники отдела», содержащий поля:
   * ID записи о сотруднике;
   * ID записи об отделе.

2) Необходимо создать списки объектов классов, содержащих тестовые данные (3-5 записей), таким образом, чтобы первичные и вторичные ключи соответствующих записей были связаны по идентификаторам.

3) Необходимо разработать запросы в соответствии с Вашим вариантом. Запросы сформулированы в терминах классов «Сотрудник» и «Отдел», которые используются в примере. Вам нужно перенести эти требования в Ваш вариант предметной области. При разработке запросов необходимо по возможности использовать функциональные возможности языка Python (list/dict comprehensions, функции высших порядков).

Для реализации запроса №2 введите в класс, находящийся на стороне связи «много», произвольный количественный признак, например, «зарплата сотрудника».

Результатом рубежного контроля является документ в формате PDF, который содержит текст программы и результаты ее выполнения.

**Вариант Г.**

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех отделов, у которых название начинается с буквы «А», и список работающих в них сотрудников.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов с максимальной зарплатой сотрудников в каждом отделе, отсортированный по максимальной зарплате.
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по отделам, сортировка по сотрудникам произвольная.



Текст программы:

class Disk:  
 def \_\_init\_\_(self, id, name, volume, speed, pc\_id):  
 self.id = id  
 self.name = name  
 self.volume = volume  
 self.speed = speed  
 self.pc\_id = pc\_id  
  
 def \_\_str\_\_(self):  
 return f"{self.name}, volume: {self.volume}"  
  
  
class PC:  
 def \_\_init\_\_(self, id, owner):  
 self.id = id  
 self.owner = owner  
  
 def \_\_str\_\_(self):  
 return f"{self.owner}"  
  
  
class PCDisks:  
 def \_\_init\_\_(self, pc\_id, disk\_id):  
 self.pc\_id = pc\_id  
 self.disk\_id = disk\_id  
  
 def \_\_str\_\_(self):  
 return f"{self.disk\_id[0]} - {self.pc\_id[0]}"  
  
  
disks = [Disk(1, "HDD1", 2000, 128, 1),  
 Disk(2, "HDD2", 1000, 128, 2),  
 Disk(3, "SSD1", 1000, 550, 1),  
 Disk(4, "SSD2", 2000, 450, 3),  
 Disk(5, "M2", 500, 600, 2)  
 ]  
  
pcs = [PC(1, "user"),  
 PC(2, "admin"),  
 PC(3, "employee")  
 ]  
  
disk\_pc = [PCDisks(1, 3),  
 PCDisks(1, 2),  
 PCDisks(3, 2),  
 PCDisks(5, 1),  
 PCDisks(5, 3),  
 PCDisks(4, 3),  
 PCDisks(2, 2),  
 PCDisks(2, 3),  
 ]  
  
  
def main():  
 otm = [(i, j) for i in disks for j in pcs if i.pc\_id == j.id]  
 otm.sort(key=lambda x: x[1].id)  
 mtm\_tmp = [(i.owner, j.disk\_id) for i in pcs for j in disk\_pc if i.id == j.pc\_id]  
 mtm = [(i[0], str(j)) for i in mtm\_tmp for j in disks if i[1] == j.id]  
 print('Задание 1:')  
 print('Название дисков установленных в ПК, где владелец начинается с \'a\':')  
 ans = []  
 for i in otm:  
 if i[1].owner.startswith('a'):  
 ans.append(i[1].owner)  
 print(f"{i[0].name} установлен в {i[1]}")  
 print('Пк у которых владелец начинается с \'a\':', \*list(set(ans)))  
  
 print('Задание 2:')  
 print('Владельцы компьютеров с максимальными объемами дисков, установленных в них:')  
 dic = {}  
 for i in otm:  
 if i[1].owner in dic:  
 dic[i[1].owner] = min(dic[i[1].owner], i[0].volume)  
 else:  
 dic[i[1].owner] = i[0].volume  
 sorted\_dic = sorted(dic.items(), key=lambda x: x[1])  
 for el in sorted\_dic:  
 print(f"Владелец: {el[0]}, Объем: {el[1]}")  
  
 print('Задание 3:')  
 print('Список всех компьютеров и дисков, отсортированных по компьютерам:')  
 print(sorted(mtm, key=lambda x: x[0]))  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

Результат работы программы:

F:\Python\python.exe F:\Уник\прога\proga\_sem3\RK1\main.py

Задание 1:

Название дисков установленных в ПК, где владелец начинается с 'a':

HDD2 установлен в admin

M2 установлен в admin

Пк у которых владелец начинается с 'a': admin

Задание 2:

Владельцы компьютеров с максимальными объемами дисков, установленных в них:

Владелец: admin, Объем: 500

Владелец: user, Объем: 1000

Владелец: employee, Объем: 2000

Задание 3:

Список всех компьютеров и дисков, отсортированных по компьютерам:

[('admin', 'HDD2, volume: 1000'), ('admin', 'SSD1, volume: 1000'), ('employee', 'HDD2, volume: 1000'), ('user', 'SSD1, volume: 1000'), ('user', 'HDD2, volume: 1000')]

Process finished with exit code 0