

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

Факультет «Радиотехнический»

Кафедра РТ5 «Системы обработки информации и управления»

Отчет по рубежному контролю №1

По курсу «Парадигмы и конструкции языков программирования»

Вариант Е4

Выполнил:

студент группы РТ5-31Б:

Гладышев А.К.

Подпись и дата:

Проверил:

преподаватель кафедры ИУ5

Гапанюк Ю.Е.

Подпись и дата:

Москва 2024

Предметная область: Дисплейный класс, компьютеры

Задание 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.

Задание 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

Задание 3:

«Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех сотрудников, у которых фамилия начинается с буквы «А», и названия их отделов

Текст программы:

```
class DispClass:
```

```
    def __init__(self, id_dispClass: int, audit_numb: int, admin_name: str, cmp_nub: int):
```

```
        self._id = id_dispClass
```

```
        self._numb = audit_numb
```

```
        self._admin = admin_name
```

```
        self._num_comp = cmp_nub
```

```
    @property
```

```
    def get_id(self) -> int:
```

```
        return self._id
```

```
    @property
```

```
    def get_aud_num(self) -> int:
```

```
        return self._numb
```

```
    @property
```

```
def get_admin_name(self) -> str:  
    return self._admin
```

```
@property
```

```
def get_comp_numb(self) -> int:  
    return self._num_comp
```

```
class PC:
```

```
def __init__(self, self_id: int, comp_class_id: int, disk_size: int, motherboard: str):  
    self._id = self_id  
    self._auditId= comp_class_id  
    self._disk_size = disk_size  
    self._mother_board = motherboard
```

```
@property
```

```
def get_id(self) -> int:  
    return self._id
```

```
@property
```

```
def get_adudit(self) -> int:  
    return self._auditId
```

```
@property
```

```
def get_disk_size(self) -> int:  
    return self._disk_size
```

```
@property
```

```
def get_mother_board(self) -> str:  
    return self._mother_board
```

```
class AudPC():
```

```
def __init__(self,id_aud:int, id_pc: int):
```

```
self._idaud = id_aud
```

```
self._idpc = id_pc
```

```
@property
```

```
def get_aud_id(self) -> int:
```

```
    return self._idaud
```

```
@property
```

```
def get_pc_id(self) -> int:
```

```
    return self._idpc
```

```
def task1(auditories: list[DispClass], computers: list[PC]):
```

```
    print("Задание 1")
```

```
    print("Имя рук.\tсерийный номер ПК\tматплата")
```

```
    answer = [(a,c) for a in auditories for c in computers if a.get_id == c.get_adudit and  
a.get_admin_name == "Nikita"]
```

```
    for (a,c) in answer:
```

```
        print(f'{a.get_admin_name} {a.get_id}\t{c.get_id}\t\t{c.get_mother_board}')
```

```
    print()
```

```
def task2(auditories: list[DispClass], computers: list[PC]):
```

```
    print("Задание 2")
```

```
    avg_space = []
```

```
    for otd in auditories:
```

```
        avg_sum = 0
```

```
        for comp in computers:
```

```
            if otd.get_id == comp.get_adudit:
```

```
                avg_sum += comp.get_disk_size
```

```
    avg_sum = avg_sum / otd.get_comp_numb
```

```

        avg_space.append((round(avg_sum,2),otd.get_id,))
avg_space.sort(reverse=True)
print("ID отдела\tСредний объем памяти компьютера в классе")
for a in range(len(avg_space)):
    print('{}\t\t{}'.format(avg_space[a][1],avg_space[a][0]))
print()

def task3(audit: list[DispClass], pc: list[PC], aud_pc: list[AudPC]):
    print("Задание 3")
    pc_with_a = []
    for otd in auditories:
        for comp in computers:
            if otd.get_id == comp.get_adudit and comp.get_mother_board.startswith('A'):
                pc_with_a.append((comp.get_id,otd.get_id,comp.get_mother_board,otd.get_admin_name))

    print("ID PC\tID аудитории\тназвание матплаты\tзав.Аудитроии")
    for i in aud_pc:
        if pc[i.get_pc_id].get_mother_board.startswith('A'):
            pc_with_a.append((pc[i.get_pc_id].get_id,i.get_aud_id,pc[i.get_pc_id].get_mother_board,
            audit[i.get_aud_id].get_admin_name))

    for a in range(len(pc_with_a)):
        print('{}\t{}\t\t{}\t\t\t{}'.format(pc_with_a[a][0],pc_with_a[a][1],pc_with_a[a][2],pc_with_a[a][
        3]))

auditories = [
    DispClass(0, 203, "Nikita", 3),
    DispClass(1,204,"Vladimir",3),
    DispClass(2,301,"Nikita",1)
]

```

```
computers = [  
    PC(0,0,500,"Asus"),  
    PC(1,0,100,"Gigiabyte"),  
    PC(2,0,1750, "Astra"),  
    PC(3,1,250,"Aorus"),  
    PC(4,1,1000,"Acer"),  
    PC(5,1,2000,"Netak"),  
    PC(6,2,300,"zotack")  
  
]
```

```
auditor_Comp = [  
    AudPC(0,4),  
    AudPC(1,2),  
    AudPC(0,6)  
  
]
```

```
task1(auditories,computers)  
task2(auditories,computers)  
task3(auditories,computers,auditor_Comp)
```

Результат выполнения:

Задание 1

Имя рук.	серийный номер ПК	матплата
Nikita 0	0	Asus
Nikita 0	1	Gigiabyte
Nikita 0	2	Astra
Nikita 2	6	zotack

Задание 2

ID отдела	Средний объем памяти компьютера в классе
1	1083.33
0	783.33
2	300.0

Задание 3

ID PC	ID аудитории	название матплаты	зав.Аудитории
0	0	Asus	Nikita
2	0	Astra	Nikita
3	1	Aorus	Vladimir
4	1	Acer	Vladimir
4	0	Acer	Nikita
2	1	Astra	Vladimir

PS: C:\cringe(ov)\ehtest\LabsPТКУАР\BK>