

МГТУ им. Н.Э. Баумана

Отчёт по рубежному контролю №1
по курсу «Базовые компоненты и интернет-технологии»
Вариант 17.

Руководитель
Гапанюк Ю.Е.
28.10.2022

Студент группы ИУ5-34Б
Стукалов И.Д.
28.10.2022

2022 г.

Полученное задание:

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

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

Необходимо разработать запросы в соответствии с Вашим вариантом.

Предметная область: класс_1 – Дирижер, класс_2 – Оркестр, вариант запросов: Г.

Запросы:

1. «Оркестр» и «Дирижер» связаны соотношением один-ко-многим.
Выведите список всех оркестров, у которых название начинается с буквы «А», и список работающих в них дирижеров.
2. «Оркестр» и «Дирижер» связаны соотношением один-ко-многим.
Выведите список оркестров с максимальной зарплатой дирижеров в каждом оркестре, отсортированный по максимальной зарплате.
3. «Оркестр» и «Дирижер» связаны соотношением многие-ко-многим.
Выведите список всех связанных дирижеров и оркестров, отсортированный по оркестрам, сортировка по дирижерам произвольная.

Текст программы:

```
from operator import itemgetter

class Director:
    """Дирижер"""

    def __init__(self, id, name, salary, orch_id):
        self.id = id
        self.name = name
        self.salary = salary
        self.orch_id = orch_id

class Orchestra:
    """Оркестр"""

    def __init__(self, id, title):
        self.id = id
        self.title = title

class DirectorOrchestra:
```

```

"""
'Дирижеры оркестров' для реализации
связи многие-ко-многим
"""

def __init__(self, dir_id, orch_id):
    self.dir_id = dir_id
    self.orch_id = orch_id

# Дирижеры
directors = [
    Director(1, 'Бан', 150000, 1),
    Director(2, 'Вагнер', 2800, 2),
    Director(3, 'Деймур', 100000, 2),
    Director(4, 'Ким Чен Ын', 86400, 3),
    Director(5, 'Поляков', 44444, 4)
]

# Оркестры
orchestras = [
    Orchestra(1, 'Банановая республика'),
    Orchestra(2, 'AGSPD'),
    Orchestra(3, 'BTS'),
    Orchestra(4, 'Дрилл Авиация')
]

directors_orchestras = [
    DirectorOrchestra(1, 1),
    DirectorOrchestra(2, 2),
    DirectorOrchestra(3, 3),
    DirectorOrchestra(4, 4),
    DirectorOrchestra(5, 1),

    DirectorOrchestra(5, 1),
    DirectorOrchestra(2, 3),
    DirectorOrchestra(3, 2),
    DirectorOrchestra(4, 1),
    DirectorOrchestra(5, 4)
]

def main():
    # Соединение данных один-ко-многим
    one_to_many = [(dir.name, dir.salary, orch.title)
                    for orch in orchestras
                    for dir in directors
                    if dir.orch_id == orch.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(orch.title, dir.orch_id, dir.dir_id)
                          for orch in orchestras
                          for dir in directors_orchestras
                          if orch.id == dir.orch_id]

    many_to_many = [(dir.name, dir.salary, orchestra_name)
                     for orchestra_name, orch_id, dir_id in many_to_many_temp
                     for dir in directors if dir.id == dir_id]

    print('\nЗадание 71')
    task_1 = {}

```

```

for orch in orchestras:
    if orch.title[0] == 'A':
        d_emps = list(filter(lambda i: i[2] == orch.title, one_to_many))
        d_emps_names = [x for x, _, _ in d_emps]
        task_1[orch.title] = d_emps_names

print(task_1)

print('\nЗадание Г2')
task_2_unsorted = []
for orch in orchestras:
    orch_Directors = list(filter(lambda i: i[2] == orch.title,
one_to_many))
    if len(orch_Directors) > 0:
        orch_sals = [sal for _, sal, _ in orch_Directors]
        orch_sals_sum = max(orch_sals)
        task_2_unsorted.append((orch.title, orch_sals_sum))

task_2 = sorted(task_2_unsorted, key=itemgetter(1), reverse=True)
print(task_2)

print('\nЗадание Г3')
task_3 = sorted(many_to_many, key=itemgetter(2))
for i in task_3:
    print(i)

if __name__ == '__main__':
    main()

```

Результаты выполнения:

```

Задание Г1
{'AGSPD': ['Вагнер', 'Деймур']}

Задание Г2
[('Банановая республика', 150000), ('AGSPD', 100000), ('BTS', 86400), ('Дрилл Авиация', 44444)]

Задание Г3
('Вагнер', 2800, 'AGSPD')
('Деймур', 100000, 'AGSPD')
('Деймур', 100000, 'BTS')
('Вагнер', 2800, 'BTS')
('Бан', 150000, 'Банановая республика')
('Поляков', 44444, 'Банановая республика')
('Поляков', 44444, 'Банановая республика')
('Ким Чен Ын', 86400, 'Банановая республика')
('Ким Чен Ын', 86400, 'Дрилл Авиация')
('Поляков', 44444, 'Дрилл Авиация')

```