

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

Отчёт по рубежному контролю №1 по курсу «Разработка интернет
приложений».

Вариант А4.

Выполнил:
Анцифров Н. С.
студент группы ИУ5-51Б

Проверил:
Гапанюк Ю. Е.

Подпись и дата:

Подпись и дата:

Москва, 2021 г.

Содержание:

1. Задание рубежного контроля.
2. Описание объектов классов.
3. Текст программы.
4. Результат выполнения программы.

1. Задание рубежного контроля.

Необходимо разработать программу на языке Python, которая отвечает следующим условиям:

1) В программе должно быть два класса данных в соответствии с вариантом предметной области, которые связаны отношениями один-ко-многим и многие-ко-многим.

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

3) В программе должны быть реализованы запросы в соответствии с вариантом. При разработке запросов необходимо по возможности использовать функциональные возможности языка Python (list/dict comprehensions и функции высших порядков).

Для реализации запроса №2 нужно ввести в класс, находящийся на стороне связи «много», произвольный количественный признак.

Вариант задания:

Вариант предметной области – вариант 4.

№ варианта	Класс 1	Класс 2
4	Компьютер	Дисплейный класс

Вариант запросов – вариант А.

1. «Дисплейный класс» и «Компьютер» связаны соотношением один-ко-многим. Выведите список всех связанных компьютеров и дисплейных классов, отсортированный по дисплейным классам по возрастанию, сортировка по компьютерам произвольная.

2. «Дисплейный класс» и «Компьютер» связаны соотношением один-ко-многим. Выведите список дисплейных классов с суммарным количеством компьютеров в каждом дисплейном классе, отсортированный по количеству компьютеров по убыванию.

3. «Дисплейный класс» и «Компьютер» связаны соотношением многие-ко-многим. Выведите список всех дисплейных классов, у которых в номере есть цифра «1», и список находящихся в них компьютеров (их серийные номера).

2. Описание объектов классов.

Класс «Компьютер» включает в себя следующие объекты:

- id – идентификатор,
- name – наименование производителя ПК,
- ser – серийный номер конкретной единицы,
- class_id – ID дисплейного класса, где располагается компьютер.

Класс «Дисплейный класс» включает в себя следующие объекты:

- id – идентификатор,
- numb – номер помещения дисплейного класса.

3. Текст программы.

```
from operator import itemgetter
class Comp:
    """Компьютер"""
    def __init__(self, id, name, ser, class_id):
        self.id = id
        self.name = name
        self.ser = ser
        self.class_id = class_id
class DispCl:
    """Дисплейный класс"""
    def __init__(self, id, numb):
        self.id = id
        self.numb = numb
class CompDisp:
    """
    'Компьютеры в дисплейных классах' для реализации
    связи многие-ко-многим
    """
    def __init__(self, class_id, comp_id):
        self.class_id = class_id
        self.comp_id = comp_id

# Дисплейные классы
classes = [
    DispCl(1, '404'),
    DispCl(2, '113'),
    DispCl(3, '41'),
    DispCl(4, '362'),
    DispCl(5, '511'),
]

# Компьютеры
comps = [
    Comp(1, 'HP', '1AAA01', 1),
    Comp(2, 'DELL', '2BBB02', 1),
    Comp(3, 'DELL', '3CCC03', 4),
    Comp(4, 'TOSHIBA', '4DDD04', 2),
    Comp(5, 'ASUS', '5EEE05', 5),
]
```

```

# Компьютеры и дисплейные классы (для "многих ко многим")
compsindisps = [
    CompDisp(1,1),
    CompDisp(1,2),
    CompDisp(2,4),
    CompDisp(4,3),
    CompDisp(5,5),
]

def main():
    """Основная функция"""

    # Соединение данных один-ко-многим
    one_to_many = [(comp.name, comp.ser, cl.numb)
                    for comp in comps
                    for cl in classes
                    if comp.class_id==cl.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(cl.numb, ccl.class_id, ccl.comp_id)
                          for cl in classes
                          for ccl in compsindisps
                          if cl.id==ccl.class_id]

    many_to_many = [(comp.ser, numb)
                     for numb, comp_id, class_id in many_to_many_temp
                     for comp in comps
                     if (comp.id==comp_id)]

    print('Задание 1')
    res_1 = sorted(one_to_many, key=itemgetter(2))
    print(res_1)

    print('\nЗадание 2')
    res_2_unsorted = []
    # Перебираем все дисплейные классы
    for c in classes:
        # Количество компьютеров в классах
        kol = sum ([comp.id for comp in comps if (comp.class_id == c.id)])
        res_2_unsorted.append((c.numb, kol))

    # Сортировка по возрастанию
    res_2 = sorted(res_2_unsorted, key=itemgetter(1), reverse=True)
    print(res_2)

    print('\nЗадание 3')
    res_3 = {}
    # Перебираем все дисплейные классы
    for c in classes:
        if '1' in c.numb:
            # Список классов с цифрой "1"
            listcl = list(filter(lambda a: a[1] == c.numb, many_to_many))
            # Только серийные номера компьютеров
            lisclcomp = [x for x,_ in listcl]
            res_3[c.numb] = lisclcomp

    print(res_3)

if __name__ == '__main__':
    main()

```

4. Результат выполнения программы.

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ ТЕРМИНАЛ КОНСОЛЬ ОТЛАДКИ

```
PS C:\Users\antsi\OneDrive\Документы\МГТУ\5 семестр\РИП> c::; cd 'c:\Users\antsi\OneDrive\Документы\МГТУ\5 семестр\РИП'; & 'C:\Users\antsi\AppData\Local\Programs\Python\Python39\python.exe' 'c:\Users\antsi\.vscode\extensions\ms-python.python-2021.10.1365161279\pythonFiles\lib\python\debugpy\launcher' '65167' '--' 'c:\Users\antsi\OneDrive\Документы\МГТУ\5 семестр\РИП\rk1.py'
```

Задание 1

```
[('TOSHIBA', '4DD04', '113'), ('DELL', '3CCC03', '362'), ('HP', '1AAA01', '404'), ('DELL', '2BBB02', '404'), ('ASUS', '5EEE05', '511')]
```

Задание 2

```
[('511', 5), ('113', 4), ('404', 3), ('362', 3), ('41', 0)]
```

Задание 3

```
{'113': ['2BBB02'], '41': [], '511': ['5EEE05']}
```

```
PS C:\Users\antsi\OneDrive\Документы\МГТУ\5 семестр\РИП> █
```