

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

Факультет «Радиотехнический»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчёт по рубежному контролю №1
Вариант №15

Выполнил:
студент группы РТ5-31Б
Мицкевич Влад

Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.

Подпись и дата:

Москва, 2021 г

Описание задания

Рубежный контроль представляет собой разработку программы на языке Python, которая выполняет следующие действия:

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 Е.

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.

Предметная область:

№ варианта	Класс 1	Класс 2
15	Файл	Каталог Файлов

Текст программы

```
class File:
    def __init__(self, id_file, name_file, size_file, file_extension, id_directory):
        self.name_file = name_file
        self.size_file = size_file
        self.file_extension = file_extension
        self.id_file = id_file
        self.id_directory = id_directory
class Directory:
    def __init__(self, name_directory, size_directory, id_directory):
        self.name_directory = name_directory
        self.size_directory = size_directory
        self.id_directory = id_directory
    def __repr__(self):
        return f'{self.size_directory} {self.name_directory} {self.size_directory}'

class DirFile:
    def __init__(self, id_directory, id_file):
        self.id_directory = id_directory
        self.id_file = id_file

files = [File(7, "Lab1", 101, "docx", 1), File(5, "Lab2", 150, "docx", 1), File(120, "Lab3", 98,
"docx", 1),
        File(4, "Math", 700, "pdf", 3), File(27, "Python_Book", 850, "pdf", 3), File(2, "It",
1500, "djvu", 3),
        File(145, "Tester", 160, "py", 7)]

directories = [Directory("Labs", 340, 1), Directory("Film", 0, 45), Directory("BookForLabs",
3050, 3), Directory("Pycharm_programmers", 160, 7)]

one_to_many = [(x,[y for y in files if y.id_directory == x.id_directory]) for x in directories]
many_to_many = [DirFile(1, 7), DirFile(1, 5), DirFile(1, 120), DirFile(1, 5416), DirFile(3, 4),
DirFile(7, 145), DirFile(3, 27)]

def main():
    #«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.
    print("Задание E1-----")
    print("Вывести список директорий, у которых в названии присутствует слово 'Labs', с файлами,
    находящимися внутри данной папки")
    for dir, fil in one_to_many:
        if dir.name_directory.find('Labs') != -1:
            print(dir.name_directory,end= ': ')
            for x in fil:
                print(x.name_file, end = '\t')
            print()
    print("-----")

    #«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
    def func(lst):
        return lst[1]
    lst = list()
    print("\nЗадание E2-----")
    print("Вывести в порядке возрастания средний размер файлов, которые мы рассматриваем в
    выбранной папке")
```

```

for dir, fil in one_to_many:
    sr = 0
    sum = 0
    if len(fil) != 0:
        for x in fil:
            sum += x.size_file
        sr = round(sum / len(fil), 2)
        lst.append([dir.name_directory, sr])
    else:
        lst.append([dir.name_directory, 0])

print(sorted(lst, key=func))
print("-----")
-----")

#«Отдел» и «Сотрудник» связаны соотношением многие - ко - многим. Выведите список всех
сотрудников, у
#которых фамилия начинается с буквы «А», и названия их отделов.
print("\nЗадание ЕЗ-----")
-----")
print("Вывести файлы, у которых первая буква в название - А, дополнительно выводя на консоль
его папку")
#Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех
сотрудников,
# у которых фамилия начинается с буквы «А», и названия их отделов.
for x in many_to_many:
    for y in files:
        if x.id_file == y.id_file:
            if y.name_file[0] == 'L':
                print(y.name_file, end= "\t")
                for z in directories:
                    if z.id_directory == x.id_directory:
                        print(z.name_directory)
print("-----")
-----")

if __name__ == '__main__':
    main()

```

Пример работы программы

```
Задание E1-----
Вывести список директорий, у которых в названии присутствует слово 'Labs', с файлами, находящимися внутри данной папки
Labs: Lab1 Lab2 Lab3
BookForLabs: Math Python_Book It
-----
```

```
Задание E2-----
Вывести в порядке возрастания средний размер файлов, которые мы рассматриваем в выбранной папке
[['Film', 0], ['Labs', 116.33], ['Pycharm_programmers', 160.0], ['BookForLabs', 1016.67]]
-----
```

```
Задание E3-----
Вывести файлы, у которых первая буква в название - A, дополнительно выводя на консоль его папку
Lab1 Labs
Lab2 Labs
Lab3 Labs
-----
```