

Рубежный контроль №1 по WEB-разработке

Студент: Трофимов Е. Е.

Группа: СМ10-61Б

Вариант 22

22	Библиотека	Язык программирования
----	------------	--------------------------

Условия рубежного контроля №1 по курсу “Разработка WEB приложений”

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.

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.

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 указанные далее запросы. 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).

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по отделам, сортировка по сотрудникам произвольная.

2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов с суммарной зарплатой сотрудников в каждом отделе, отсортированный по суммарной зарплате.

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.

Решение

Код:

```
# используется для сортировки
from operator import itemgetter

class Lib:
    def __init__(self, id, name, purpose, amount_of_functions,
lang_id):
        self.id = id
        self.name = name
        self.purpose=purpose #направленность библиотеки
        self.amount_of_functions = amount_of_functions #число
функций в библиотеке
        self.lang_id = lang_id #идентификатор языка
```

```

class Lang:
    def __init__(self, id, name):
        self.id = id
        self.name = name

#Языки
langs = [
    Lang(1, 'Python'),
    Lang(2, 'C'),
    Lang(3, 'Pascal'),
]

#Библиотеки
libs = [
    Lib(1, 'math', 'математические функции', 44, [1, 2, 3]),
    Lib(2, 'stdio.h', 'ввод-вывод данных', 47, [2, 0, 0]),
    Lib(3, 'itertools', 'итераторы', 18, [1, 0, 0]),
    Lib(4, 'Dos', 'взаимодействие с операционной системой', 25,
[3, 0, 0]),
]

def main():
    # Соединение данных один-ко-многим
    one_to_many = [(e.name, e.amount_of_functions, d.name)
        for d in langs
        for e in libs
        for f in range(3)
        if e.lang_id[f]==d.id]

    print('Задание 1')
    sort1 = sorted(one_to_many, key=itemgetter(2))
    print(sort1)

    print('\nЗадание 2')
    unsorted1 = []
    # Перебираем все отделы
    for d in langs:
        # Список библиотек языка
        d_libs = list(filter(lambda i: i[2]==d.name,
one_to_many))
        # Если есть библиотеки
        if len(d_libs) > 0:
            # мощности множеств функций каждой библиотеки языка
            d_amount = [x for _,x,_ in d_libs]
            # Суммарная мощность библиотек языка
            d_amounts_sum = sum(d_amount)
            unsorted1.append((d.name, d_amounts_sum))
    # Сортировка по числу функций (по возрастанию)
    sorted1 = sorted(unsorted1, key=itemgetter(1))
    print(sorted1)

main()

```

Результат выполнения:

Задание 1

```
[('math', 44, 'C'), ('stdio.h', 47, 'C'), ('math', 44, 'Pascal'),  
('Dos', 25, 'Pascal'), ('math', 44, 'Python'), ('itertools', 18,  
'Python')]
```

Задание 2

```
[('Python', 62), ('Pascal', 69), ('C', 91)]
```

** Process exited - Return Code: 0 **

Press Enter to exit terminal