

Парадигмы и конструкции языков программирования.

Рубежный Контроль №1. Вариант 12А.

Задание (вариант А):

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.
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех отделов, у которых в названии присутствует слово «отдел», и список работающих в них сотрудников.

Программа:

```
from operator import itemgetter

class Language:
    """Язык программирования"""

    def __init__(self, id, name, dev_id):
        self.id = id
        self.name = name
        self.dev_id = dev_id

class DevelopmentTool:
    """Средство разработки"""

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

class ToolLang:
    """Связь между средствами разработки и языками
    программирования"""
```

```

def __init__(self, dev_id, lang_id):
    self.dev_id = dev_id
    self.lang_id = lang_id

# Средства разработки
dev_tools = [
    DevelopmentTool(1, 'IDE'),
    DevelopmentTool(2, 'Text Editor'),
    DevelopmentTool(3, 'Framework'),
]

# Языки программирования
languages = [
    Language(1, 'Python', 1),
    Language(2, 'JavaScript', 1),
    Language(3, 'Ruby', 2),
    Language(4, 'Java', 3),
]

# Связь между средствами разработки и языками
tool_langs = [
    ToolLang(1, 1),
    ToolLang(1, 2),
    ToolLang(2, 3),
    ToolLang(3, 4),
]

def main():
    "Основная функция"

    # Задание 1: один-ко-многим
    one_to_many = [(l.name, d.name)
                   for d in dev_tools
                   for l in languages if l.dev_id == d.id]

    res_1 = sorted(one_to_many, key=itemgetter(1))
    print("Задание A1. \nВыведите список всех связанных средств\nразработки и языков программирования, отсортированный \n\nпо средствам разработки, сортировка по языкам программирования\nпроизвольная: \n", res_1)

    # Задание 2: суммарное количество языков
    res_2 = []
    for d in dev_tools:
        d_langs = [l for l in languages if l.dev_id == d.id]
        res_2.append((d.name, len(d_langs)))
        res_2_sorted = sorted(res_2, key=itemgetter(1),
reverse=True)

```

```
print("\nЗадание A2. \nВыведите список средств разработки с  
суммарным количеством языков программирования \n  
\nв средстве разработки, отсортированный по количеству: \n",  
res_2_sorted)
```

```
# Задание 3: фильтрация по введенному слову  
keyword = input("\nЗадание A3. \nВыведите список всех средств  
разработки, у которых в названии присутствует \n  
\nслово, которое вводится с клавиатуры, и список поддерживаемых им  
языков программирования: \nВведите слово для поиска: ")  
res_3 = {d.name: [l.name for l in languages if l.dev_id ==  
d.id]  
          for d in dev_tools if keyword in d.name}  
print(res_3)
```

```
if __name__ == '__main__':  
    main()
```

```
===== RESTART: /Users/veraleonteva/Desktop/RK1.py =====  
Задание A1.  
Выведите список всех связанных средств разработки и языков программирования, отсортированный  
по средствам разработки, сортировка по языкам программирования произвольная:  
[('Java', 'Framework'), ('Python', 'IDE'), ('JavaScript', 'IDE'), ('Ruby', 'Text Editor')]  
  
Задание A2.  
Выведите список средств разработки с суммарным количеством языков программирования  
в средстве разработки, отсортированный по количеству:  
[('IDE', 2), ('Text Editor', 1), ('Framework', 1)]  
  
Задание A3.  
Выведите список всех средств разработки, у которых в названии присутствует  
слово, которое вводится с клавиатуры, и список поддерживаемых им языков программирования:  
Введите слово для поиска: Text  
{'Text Editor': ['Ruby']}  
>>> |
```