**РК №1**

**ИУ5-55Б**

**Усынин Юрий**

Запрос Д  
Предметная область 21 - Оператор и Язык программирования

Задание:

1. «Язык программирования» и «Оператор» связаны соотношением один-ко-многим. Выведите список всех операторов, у которых название заканчивается на «ия», и их языки программирования.

2. «Язык программирования» и «Оператор» связаны соотношением один-ко-многим. Выведите список языков программирования со средним значением «number» у соответствующих операторов, отсортированный по этому значению (отдельной функции вычисления среднего значения в Python нет, нужно использовать комбинацию функций вычисления суммы и количества значений).

«number» - любое произвольное число, необходимое для выполнения задания рубежного контроля.

3. «Язык программирования» и «Оператор» связаны соотношением многие-ко-многим. Выведите список всех языков программирования, у которых название начинается с буквы «P», и список используемых в них операторов.

Листинг:

*# Усынин Юрий ИУ5-55Б  
# Запрос Д  
# Предметная область 21 - Оператор и Язык программирования*from operator import itemgetter  
  
class Operator:  
 *# Оператор* def \_\_init\_\_(self, id, title, number, pl\_id):  
 self.id = id  
 self.title = title  
 *# number - любое произвольное число, так как по заданию рк необходимо любое числовое значение* self.number = number  
 self.pl\_id = pl\_id  
  
class Programming\_language:  
 *# Язык программирования* def \_\_init\_\_(self, id, name):  
 self.id = id  
 self.name = name  
  
class OperProg\_lan:  
 *# Операторы языка программирования  
 # для реализации связи многие-ко-многим* def \_\_init\_\_(self, progLan\_id, oper\_id):  
 self.progLan\_id = progLan\_id  
 self.oper\_id = oper\_id  
  
*# Языки программирования*progLang = [  
 Programming\_language(1, **"Basic"**),  
 Programming\_language(2, **"Pascal"**),  
 Programming\_language(3, **"C++"**),  
 Programming\_language(4, **"Python"**),  
 Programming\_language(5, **"Java"**),  
 Programming\_language(6, **"C#"**)  
]  
  
*# операторы*oper = [  
 Operator(1, **"Not - инверсия"**, 17, 1),  
 Operator(2, **"And - конъюнкция"**, 50, 2),  
 Operator(3, **"Or - дизъюнкция"**, 22, 2),  
 Operator(4, **"== - равно"**, 14, 3),  
 Operator(5, **"! - инверсия"**, 38, 3),  
 Operator(6, **"&& - конъюнкция"**, 32, 3),  
 Operator(7, **"|| - дизъюнкция"**, 28, 3)  
]  
  
oper\_progLang = [  
 OperProg\_lan(1, 1),  
 OperProg\_lan(2, 2),  
 OperProg\_lan(2, 3),  
 OperProg\_lan(3, 4),  
 OperProg\_lan(3, 5),  
 OperProg\_lan(3, 6),  
 OperProg\_lan(3, 7),  
  
 OperProg\_lan(4, 1),  
 OperProg\_lan(4, 2),  
 OperProg\_lan(4, 3),  
 OperProg\_lan(5, 4),  
 OperProg\_lan(6, 5),  
 OperProg\_lan(6, 6),  
 OperProg\_lan(6, 7),  
]  
  
def main():  
 *# соединение данных один-ко-многим* one\_to\_many = [(c.title, c.number, o.name)  
 for o in progLang  
 for c in oper  
 if c.pl\_id == o.id]  
  
 *# соединение данных многие-ко-многим* many\_to\_many\_temp = [(o.name, co.progLan\_id, co.oper\_id)  
 for o in progLang  
 for co in oper\_progLang  
 if o.id == co.progLan\_id]  
  
 many\_to\_many = [(c.title, c.number, pl\_name)  
 for pl\_name, progLan\_id, oper\_id in many\_to\_many\_temp  
 for c in oper if c.id == oper\_id]  
  
 print(**'Задание Д1'**)  
 res1 = []  
 for o in one\_to\_many:  
 if o[0][-2:] == **"ия"**:  
 res1.append(o[0:3:2])  
 print(res1)  
  
 print(**'**\n**Задание Д2'**)  
 res2\_unsorted = []  
 for o in progLang:  
 o\_oper = list(filter(lambda i: i[2] == o.name, one\_to\_many))  
 if len(o\_oper) > 0:  
 o\_number = [number for \_, number, \_ in o\_oper]  
 o\_number\_sum = sum(o\_number)  
 o\_number\_count = len(o\_number)  
 o\_number\_average = o\_number\_sum / o\_number\_count  
 res2\_unsorted.append((o.name, int(o\_number\_average)))  
 res2 = sorted(res2\_unsorted, key=itemgetter(1), reverse=True)  
 print(res2)  
  
 print(**'**\n**Задание Д3'**)  
 res3 = {}  
 for o in progLang:  
 if o.name[0] == **"P"**:  
 o\_oper = list(filter(lambda i: i[2] == o.name, many\_to\_many))  
 o\_oper\_titles = [x for x, \_, \_ in o\_oper]  
 res3[o.name] = o\_oper\_titles  
 print(res3)  
  
  
if \_\_name\_\_ == **'\_\_main\_\_'**:  
 main()

Результаты:

