Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»



**Дисциплина «Разработка интернет приложений»**

**Отчет по рубежному контролю №1  
Вариант Д-22**

**ИСПОЛНИТЕЛЬ:**

Группа ИУ5-55

\_Филатова А.Е.

"31"\_октября\_2020 г.

**ПРЕПОДАВАТЕЛЬ:**

\_\_\_Гапанюк Ю.Е.\_\_

"31"\_октября\_2020 г.

Москва 2020

**Классы для предметной области**

1. Класс «Библиотека», содержащий поля:

ID записи о библиотеке;

Название библиотеки;

ID записи о языке программирования. (для реализации связи одинко-многим)

1. Класс «Язык программирования», содержащий поля:

ID записи о языке программирования;

Наименование языка программирования.

1. (Для реализации связи многие-ко-многим) Класс «Библиотеки языка программирования», содержащий поля:

ID записи о библиотеке;

ID записи о языке программирования.

**Код**

# Филатова Анастасия ИУ5-55Б  
# Запрос Д  
# Предметная область 22 - Библиотека и Язык программирования  
from operator import itemgetter  
  
class Library:  
 # Библиотека  
 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 LibrProg\_lan:  
 # Операторы языка программирования  
 # для реализации связи многие-ко-многим  
 def \_\_init\_\_(self, progLan\_id, libr\_id):  
 self.progLan\_id = progLan\_id  
 self.libr\_id = libr\_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  
libr = [  
 Library(1, "String", 17, 1),  
 Library(2, "Twig", 50, 2),  
 Library(3, "Int", 22, 2),  
 Library(4, "Double", 14, 3),  
 Library(5, "Queue", 38, 3),  
 Library(6, "Numpy", 32, 3),  
 Library(7, "Vector", 28, 3)  
]  
  
libr\_progLang = [  
 LibrProg\_lan(1, 1),  
 LibrProg\_lan(2, 2),  
 LibrProg\_lan(2, 3),  
 LibrProg\_lan(3, 4),  
 LibrProg\_lan(3, 5),  
 LibrProg\_lan(3, 6),  
 LibrProg\_lan(3, 7),  
  
 LibrProg\_lan(4, 1),  
 LibrProg\_lan(4, 2),  
 LibrProg\_lan(4, 3),  
 LibrProg\_lan(5, 4),  
 LibrProg\_lan(6, 5),  
 LibrProg\_lan(6, 6),  
 LibrProg\_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 libr  
 if c.pl\_id == o.id]  
  
 # соединение данных многие-ко-многим  
 many\_to\_many\_temp = [(o.name, co.progLan\_id, co.libr\_id)  
 for o in progLang  
 for co in libr\_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 libr if c.id == oper\_id]  
  
 print('Задание Д1')  
 res1 = []  
 for o in one\_to\_many:  
 if o[0][-1:] == "g":  
 res1.append(o[0:3:2])  
 print(res1)  
  
 print('\nЗадание 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('\nЗадание 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 выполнения заданий**

