ИУ5-32Б,

Фролов Илья

***Рубежный контроль***

***Вариант 22***

******

Вариант Б:

1. «Язык программирования» и «Библиотека» связаны соотношением один-ко-многим. Выведите список всех связанных библиотек и языков программирования, отсортированных по библиотекам, сортировка по языкам программирования произвольная.
2. «Язык программирования» и «Библиотека» связаны соотношением один-ко-многим. Выведите список языков программирования с количеством библиотек у каждого языка программирования, отсортированный по убыванию количества библиотек.
3. «Язык программирования» и «Библиотека» связаны соотношением многие-ко-многим. Выведите список всех библиотек, у которых имя библиотеки заканчивается на «4», и названия их языков программирования.

**Код программы**

*from operator import itemgetter  
  
  
class ProgramLang:  
 def \_\_init\_\_(self, id, name):  
 self.id = id  
 self.name = name  
  
  
class Library:  
 def \_\_init\_\_(self, id, name, doc\_lib\_href, prog\_lang\_id):  
 self.id = id  
 self.name = name  
 self.doc\_lib\_href = doc\_lib\_href  
 self.prog\_lang\_id = prog\_lang\_id  
  
  
class ProgLangLib:  
 def \_\_init\_\_(self, prog\_lang\_id, lib\_id):  
 self.prog\_lang\_id = prog\_lang\_id  
 self.lib\_id = lib\_id  
  
  
prog\_langs = [  
 ProgramLang(1, "C++"),  
 ProgramLang(2, "Java"),  
 ProgramLang(3, "Kotlin"),  
]  
  
libs = [  
 Library(1, "JUnit4", "https://kotlinlang.org/docs/jvm-test-using-junit.html", 3),  
 Library(2, "JUnit4", "https://junit.org/junit4/", 2),  
 Library(3, "JUnit5", "https://junit.org/junit5/", 3),  
 Library(4, "Cucumber", "https://cucumber.io/docs/installation/java/", 3),  
 Library(5, "iostream", "https://en.cppreference.com/w/cpp/header/iostream", 1),  
 Library(6, "stdlib", "https://en.cppreference.com/w/cpp/header/cstdlib", 1)  
]  
  
pl\_libs = [  
 ProgLangLib(1, 5),  
 ProgLangLib(1, 6),  
 ProgLangLib(3, 4),  
 ProgLangLib(3, 3),  
 ProgLangLib(2, 2),  
 ProgLangLib(3, 1),  
]  
  
  
def first\_task(lib\_list):  
 return sorted(lib\_list, key=itemgetter(0))  
  
  
def second\_task(lib\_list):  
 res\_2 = []  
 temp\_dict = dict()  
 for i in lib\_list:  
 if i[2] in temp\_dict:  
 temp\_dict[i[2]] += 1  
 else:  
 temp\_dict[i[2]] = 1  
 for i in temp\_dict.keys():  
 res\_2.append((i, temp\_dict[i]))  
  
 res\_2.sort(key=itemgetter(1), reverse=True)  
 return res\_2  
  
  
def third\_task(lib\_list, end\_ch):  
 return [(i[0], i[2]) for i in lib\_list if str(i[0]).endswith(end\_ch)]  
  
  
def main():  
 one\_to\_many = [(lib.name, lib.doc\_lib\_href, pl.name)  
 for pl in prog\_langs  
 for lib in libs  
 if lib.prog\_lang\_id == pl.id]  
  
 many\_to\_many\_temp = [(pl.name, ps.prog\_lang\_id, ps.lib\_id)  
 for pl in prog\_langs  
 for ps in pl\_libs  
 if ps.prog\_lang\_id == pl.id]  
  
 many\_to\_many = [(lib.name, lib.doc\_lib\_href, pl\_name)  
 for pl\_name, pl\_id, lib\_id in many\_to\_many\_temp  
 for lib in libs if lib.id == lib\_id]  
  
 print('Задание Б1')  
 [print(i) for i in first\_task(one\_to\_many)]  
  
 print("\nЗадание Б2")  
 print(second\_task(one\_to\_many))  
  
 print("\nЗадание Б3")  
 print(third\_task(many\_to\_many, '4'))  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()*

**Результаты выполнения**

