

Рубежный контроль по БКИТ №1

Вариант запросов Б

Вариант предметной области 22

Текст программы

используется для сортировки

```
1.  from operator import itemgetter
2.
3.  class Include:
4.      """Библиотека"""
5.      def __init__(self, id, name, popularity, lang_id):
6.          self.id = id
7.          self.name = name
8.          self.popularity = popularity
9.          self.lang_id = lang_id
10.
11. class Lang:
12.     """Язык программирования"""
13.     def __init__(self, id, name):
14.         self.id = id
15.         self.name = name
16.
17. class LangInclude:
18.     """
19.     'Библиотеки языка' для реализации
20.     связи многие-ко-многим
21.     """
22.     def __init__(self, lang_id, include_id):
23.         self.lang_id = lang_id
24.         self.include_id = include_id
25.
26. # Языки
27. lang = [
28.     Lang(1, 'C++'),
29.     Lang(2, 'C'),
30.     Lang(3, 'Python'),
31.
32.     Lang(11, 'C++ (другой)'),
33.     Lang(22, 'C (другой)'),
34.     Lang(33, 'Python (другой)'),
35. ]
36.
37. # Библиотеки
38. include = [
39.     Include(1, 'iostream', 95, 1),
40.     Include(2, 'stddef.h', 89, 2),
```

```

41.     Include(3, 'numpy', 67, 3),
42.     Include(4, 'math.h', 79, 1),
43.     Include(5, 'algorithm', 50, 1),
44. ]
45.
46. lang_include = [
47.     LangInclude(1,1),
48.     LangInclude(2,2),
49.     LangInclude(3,3),
50.     LangInclude(1,4),
51.     LangInclude(1,5),
52.
53.     LangInclude(11,1),
54.     LangInclude(22,2),
55.     LangInclude(33,3),
56.     LangInclude(11,4),
57.     LangInclude(11,5),
58. ]
59.
60. def main():
61.     """Основная функция"""
62.
63.     # Соединение данных один-ко-многим
64.     one_to_many = [(inc.name, inc.popularity, l.name)
65.                     for l in lang
66.                     for inc in include
67.                     if inc.lang_id==l.id]
68.
69.     # Соединение данных многие-ко-многим
70.     many_to_many_temp = [(l.name, li.lang_id, li.include_id)
71.                           for l in lang
72.                           for li in lang_include
73.                           if l.id==li.lang_id]
74.
75.
76.     many_to_many = [(inc.name, inc.popularity, lang_name)
77.                     for lang_name, lang_id, include_id in many_to_many_temp
78.                     for inc in include
79.                     if inc.id==include_id]
80.
81.
82.     print('Задание A1')
83.     res_11 = sorted(one_to_many, key=itemgetter(1))
84.     print(res_11)
85.
86.     print('\nЗадание A2')

```

```

87.     res_12_unsorted = []
88.
89.     for li in lang:
90.
91.         l_include = list(filter(lambda i: i[2]==li.name, one_to_many))
92.
93.         if len(l_include) > 0:
94.
95.             res_12_unsorted.append((li.name, len(l_include)))
96.
97.     res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
98.     print(res_12)
99.
100.    print('\nЗадание A3')
101.    res_13 = {}
102.
103.    for inc in include:
104.        if 'm' in inc.name:
105.
106.            d_emps = list(filter(lambda i: i[0]==inc.name, many_to_many))
107.
108.            d_emps_names = [x for _, x in d_emps]
109.
110.            res_13[inc.name] = d_emps_names
111.
112.    print(res_13)
113.
114.    if __name__ == '__main__':
115.        main()

```

Результаты выполнения программы:

```

Задание A1
[('algorithm', 50, 'C++'), ('numpy', 67, 'Python'), ('math.h', 79, 'C++'), ('stddef.h', 89, 'C'), ('iostream', 95, 'C++')]

Задание A2
[('C++', 3), ('C', 1), ('Python', 1)]

Задание A3
{'iostream': ['C++', 'C++(другой)'], 'numpy': ['Python', 'Python(другой)'], 'math.h': ['C++', 'C++(другой)'], 'algorithm': ['C++', 'C++(другой)']}

```