**Текст программы:**

from operator import itemgetter

class Language:

def \_\_init\_\_(self, id, name, complexity):

self.id = id

self.name = name

self.complexity = complexity

class Library:

def \_\_init\_\_(self, id, name, language\_id, symbols):

self.id = id

self.name = name

self.language\_id = language\_id

self.symbols = symbols

class LanguageLibrary:

def \_\_init\_\_(self, language\_id, lib\_id):

self.language\_id = language\_id

self.lib\_id = lib\_id

languages = [

Language(1, 'Python', 'medium'),

Language(2, 'C++', 'medium'),

Language(3, 'C', 'easy'),

Language(4, 'Swift', 'hard'),

Language(5, 'JavaScript', 'medium'),

Language(6, 'Java', 'medium'),

Language(7, 'C#', 'medium'),

]

libs = [

Library(1, 'requests', 2, 100),

Library(2, 'system', 2, 200),

Library(3, 'numbers', 6, 250),

Library(4, 'math', 1, 150),

Library(5, 'menu', 1, 100),

Library(6, 'play', 6, 50),

Library(7, 'reject', 5, 250),

Library(8, 'formuli', 3, 150),

Library(9, 'picture', 4, 100),

Library(10, 'css', 7, 200),

]

language\_libs = [

LanguageLibrary(1, 4),

LanguageLibrary(1, 5),

LanguageLibrary(2, 1),

LanguageLibrary(2, 2),

LanguageLibrary(3, 8),

LanguageLibrary(4, 9),

LanguageLibrary(5, 7),

LanguageLibrary(6, 6),

LanguageLibrary(6, 3),

LanguageLibrary(7, 10),

]

def main():

one\_to\_many = [(language.name, language.complexity, lib.name, lib.symbols)

for language in languages

for lib in libs

if lib.language\_id == language.id]

many\_to\_many\_temp = [(language.name, language\_lib.language\_id, language\_lib.lib\_id)

for language\_lib in language\_libs

for language in languages

if language.id == language\_lib.language\_id]

many\_to\_many = [(lib.name, language\_name)

for language\_name, language\_id, lib\_id in many\_to\_many\_temp

for lib in libs if lib.id == lib\_id]

print('Задание E1')

print(list(filter(lambda i: 'C' in i[0], one\_to\_many)))

print('Задание E2')

language\_symbols = {}

for language in languages:

language\_symbols[language.name] = []

for row in one\_to\_many:

language\_name, \_, \_, symbols = row

language\_symbols[language\_name].append(symbols)

res\_12 = [(language, round(sum(symbols) / len(symbols), 2))

for language, symbols in language\_symbols.items() if symbols]

res\_12 = sorted(res\_12, key=itemgetter(1), reverse=True)

print(res\_12)

print('Задание E3')

res\_13 = list(filter(lambda i: i[0][0] == 'm', many\_to\_many))

print(res\_13)

if \_\_name\_\_ == '\_\_main\_\_':

main()

**Результат выполнения**

Задание E1

[('C++', 'medium', 'requests', 100), ('C++', 'medium', 'system', 200), ('C', 'easy', 'formuli', 150), ('C#', 'medium', 'css', 200)]

Задание E2

[('JavaScript', 250.0), ('C#', 200.0), ('C++', 150.0), ('C', 150.0), ('Java', 150.0), ('Python', 125.0), ('Swift', 100.0)]

Задание E3

[('math', 'Python'), ('menu', 'Python')]