Отчет по РК №2

Успенский Д.А. ИУ5-31Б

Измененный код РК1:

```
# используется для сортировки
from operator import itemgetter


class Syntax:
    def __init__(self, id, op, LangID):
        self.id = id
        self.op = op
        self.LangID = LangID


class Language:
    def __init__(self, id, name):
        self.id = id
        self.name = name


class LangSynt:
    def __init__(self, LangID, SyntID):
        self.LangID = LangID
        self.SyntID = SyntID


Languages = [
    Language(1, 'Python'),
    Language(2, 'C++'),
    Language(3, 'Pascal'),
    Language(4, 'C#')
]

Syntaxs = [
    Syntax(1, 'print', 1),
    Syntax(2, 'write', 3),
    Syntax(3, 'cout', 2),
    Syntax (4, 'if', 1),
    Syntax (5, 'if', 2),
    Syntax (6, 'if', 4)
]

LangSyntx = [
    LangSynt(1, 1),
    LangSynt(3, 2),
    LangSynt(2, 3),
    LangSynt(1, 4),
```

```
    LangSynt(2, 5),
    LangSynt(4, 6),
]

def one_to_many(Languages, Syntaxs):
    return [(s.op, l.name)
            for s in Syntaxs
            for l in Languages
            if s.LangID == l.id]

def many_to_many(Languages, LangSyntx):
    many2many_temp = [(l.name, ls.LangID)
                      for l in Languages
                      for ls in LangSyntx
                      if l.id == ls.LangID]
    return[(l.name, LangName)
                for LangName, LangID in many2many_temp
                for l in Languages if l.id == LangID]

def A1(Languages, Syntaxs) -> list:
    res1 = sorted(one_to_many(Languages, Syntaxs), key=itemgetter(1, 0))
    return list(res1)

def A2(Languages, Syntaxs) -> list:
    res_2= []
    for i in Languages:
        syntaxes_of_language = [ j[0] for j in filter(lambda a: a[1]==i.name,
one_to_many)]
        res_2.append((i.name, syntaxes_of_language))
    return(sorted(res_2))

def A3(Languages, LangSyntx):
    res_3 = {}
    for l in Languages:
        if 'C' in l.name:
            language = list(filter(lambda i: i[1]== l.id, many_to_many))
            language_name = [x for x,_,_ in language]
            res_3[l.name] = language_name
    print (res_3)



#для запуска кода из командной строки
if __name__ == "__main__":
    print('Задание A1')
    print(A1(Languages, Syntaxs))
    print('Задание A2')
    print(A2(Languages, Syntaxs))
    print('Задание A3')
    print(A2(Languages, LangSyntx))
```

Код для тестов РК1:

```
import unittest
from RK1 import Syntax, Language, LangSynt, A1
class RK1_test(unittest.TestCase):
    def setUp(self):
        self.languages = [
            Language(1, 'Python'),
            Language (2, 'C++'),
            Language(3, 'Pascal'),
            Language(4, 'C#')
        ]
        self.syntaxs = [
            Syntax(1, 'print', 1),
            Syntax(2, 'write', 3),
            Syntax(3, 'cout', 2),
            Syntax (4, 'if', 1),
            Syntax (5, 'if', 2),
            Syntax (6, 'if', 4)
        ]
        self.langsynts = [
            LangSynt(1, 1),
            LangSynt(3, 2),
            LangSynt(2, 3),
            LangSynt(1, 4),
            LangSynt(2, 5),
            LangSynt(4, 6),
        ]

        def test_A1(self):
            expected_result = [
                ('if', 'C#')
                ('cout', 'C++')
                ('if', 'C++')
                ('write', 'Pascal')
                ('print', 'Python')
                ('if', 'Python')
            ]
            result = A1(self.languages, self.syntaxs)
            self.assertEqual(result,expected_result)
        def test_A2(self):
            expected_result = [
                ('Python',['print','if'])
                ('C++',['cout','if'])
                ('Pascal',['write'])
                ('C#',['if'])
            ]
            result = res22(self.languages, self.syntaxs)
            self.assertEqual(result,expected_result)
        def test_A3(self):
```

```
            expected_result = [
                ('C++',[])
                ('C#',[])
            ]
            result = res33(self.languages, 'C')
            self.assertEqual(result,expected_result)

if __name__ == "__main__":
    unittest.main()
```

Анализ результатов:

```
...
----------------------------------------------------------------------
Ran 3 tests in 0.001s

OK
```