

РК №2 по БКИТ

Задание

№1: Провести рефакторинг текста программы РК№1 таким образом, чтобы он был пригоден для модульного тестирования

№2: Для текста программы РК№1 создать модульные тесты с применением TDD-фреймворка (3 теста).

Задание для РК№1

№1: "Язык программирования" и "Среда разработки" связаны соотношениями один-ко-многим. Отсортировать среды разработки по популярности и вывести среды разработки и языки программирования.

№2: "Язык программирования" и "Среда разработки" связаны соотношениями один-ко-многим. Отсортировать языки программирования по количеству поддерживающих их средств разработки.

№3: "Язык программирования" и "Среда разработки" связаны соотношениями многие-ко-многим. Вывести все языки программирования, названия которых начинаются на "С" и названия сред, использующих их.

Изменённый текст программы

main.py

```
""" задания:
№1: "Язык программирования" и "Среда разработки" связаны соотношениями
один-ко-многим. Отсортировать среды разработки по популярности и вывести
среды разработки и языки программирования.
№2: "Язык программирования" и "Среда разработки" связаны соотношениями
один-ко-многим. Отсортировать языки программирования по количеству
поддерживающих их средств разработки.
№3: "Язык программирования" и "Среда разработки" связаны соотношениями
многие-ко-многим. Вывести все языки программирования, названия которых
начинаются на "С" и названия сред, использующих их. """

#используется для сортировки
from audioop import reverse
from operator import itemgetter
class Lang:
    def __init__(self, id, name, version):
```

```

        self.id = id
        self.name = name
        self.version = version

class IDE:
    def __init__(self, id, name, totalUsers, lang_id):
        self.id = id
        self.name = name
        self.totalUsers = totalUsers
        self.lang_id = lang_id

class LangIDE:
    def __init__(self, Lang_id, IDE_id):
        self.lang_id = Lang_id
        self.IDE_id = IDE_id

Languages = [
    Lang(0, "Python", "3.10"),
    Lang(1, "Java", "v18"),
    Lang(2, "C++", "17.2.0"),
    Lang(3, "JavaScript", "1.8.5"),
    Lang(4, "Swift", "5.6"),
    Lang(5, "C#", "5.6"),
    Lang(6, "C", "20.0"),
    Lang(7, "HTML", "HTML5"),
    Lang(8, "F#", "6"),
]

IDEs = [
    IDE(0, "Visual Studio", 1450000, 5),
    IDE(1, "VS Code", 530000, 2),
    IDE(2, "WebStorm", 115000, 3),
    IDE(3, "PyCharm", 550000, 0),
    IDE(4, "Xcode", 150000, 4),
    IDE(5, "NetBeans", 120000, 1),
    IDE(6, "Eclipse", 224000, 5),
]

Lang_IDE = [
    LangIDE(0, 0),
    LangIDE(0, 1),
    LangIDE(0, 3),
    LangIDE(1, 5),
    LangIDE(2, 1),
    LangIDE(2, 5),
    LangIDE(2, 6),
    LangIDE(3, 1),
    LangIDE(3, 2),
    LangIDE(4, 4),
    LangIDE(5, 0),
    LangIDE(5, 6),
    LangIDE(6, 0),

```

```

    LangIDE(6,6),
    LangIDE(7,1),
    LangIDE(8,0),
    LangIDE(8,1),
]

def counter(lang_id):
    count = 0
    for i in IDEs:
        if i.lang_id == lang_id:
            count+=1
    return count

def task1(one_to_many):
    B1 = sorted(one_to_many, key = itemgetter(1), reverse=True)
    print('-----')
    print('Задание Б1')
    print('-----')
    for row in B1:
        print(row[:-1])
    return B1

def task2(one_to_many):
    B2 = []
    for l in Languages:
        l_ids = list(filter(lambda i: i[4]==l.id, one_to_many))
        if len(l_ids) > 0 : B2.append((l.name, len(l_ids)))
    newB2 = sorted(B2, key = lambda i: i[1], reverse = True)
    print('-----')
    print('Задание Б2')
    print('-----')
    for row in newB2:
        print(row)
    return B2

def task3(many_to_many):
    B3 = {}
    # Перебираем языки
    for l in Languages:
        if l.name[0] == "C":
            # Список языков
            lst_IDE = list(filter(lambda i: i[1] == l.id, many_to_many))
            # Только названия IDE
            l_ids_names = [x for _, x in lst_IDE]
            # Добавляем результат в словарь
            # ключ - язык, значение - названия IDE
            B3[l.name] = l_ids_names

    print('-----')
    print('Задание Б3')
    print('-----')

```

```

    return B3

def main():
    #Соединение данных один-ко-многим
    one_to_many = [(i.name, i.totalUsers, l.name, l.version, l.id)
                    for l in Languages
                    for i in IDEs
                    if i.lang_id==l.id]

    #Задание Б2

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(i.name, li.IDE_id, li.lang_id)
                          for i in IDEs
                          for li in Lang_IDE
                          if li.IDE_id == i.id]

    many_to_many = [(l.name, lang_id, IDE_name)
                    for IDE_name, IDE_id, lang_id in many_to_many_temp
                    for l in Languages if l.id == lang_id]

    #print(one_to_many)
    task1(one_to_many)
    task2(one_to_many)
    task3(many_to_many)

main()

```

test.py

```

from unittest import TestCase, main
from main import task1, task2, task3

class testrk2(TestCase):
    def test_task1(self):
        self.assertEqual(task1(
            [('PyCharm', 550000, 'Python', '3.10', 0),
             ('NetBeans', 120000, 'Java', 'v18', 1),
             ('VS Code', 530000, 'C++', '17.2.0', 2),
             ('WebStorm', 115000, 'JavaScript', '1.8.5', 3),
             ('Xcode', 150000, 'Swift', '5.6', 4),
             ('Visual Studio', 1450000, 'C#', '5.6', 5),
             ('Eclipse', 224000, 'C#', '5.6', 5)]),
            [
                ('Visual Studio', 1450000, 'C#', '5.6', 5),
                ('PyCharm', 550000, 'Python', '3.10', 0),
                ('VS Code', 530000, 'C++', '17.2.0', 2),
                ('Eclipse', 224000, 'C#', '5.6', 5),
                ('Xcode', 150000, 'Swift', '5.6', 4),
                ('NetBeans', 120000, 'Java', 'v18', 1),
            ]
        )

```

```

        ('WebStorm', 115000, 'JavaScript', '1.8.5', 3)])

def test_task2(self):
    self.assertEqual(task2(
        [('PyCharm', 550000, 'Python', '3.10', 0),
         ('NetBeans', 120000, 'Java', 'v18', 1),
         ('VS Code', 530000, 'C++', '17.2.0', 2),
         ('WebStorm', 115000, 'JavaScript', '1.8.5', 3),
         ('Xcode', 150000, 'Swift', '5.6', 4),
         ('Visual Studio', 1450000, 'C#', '5.6', 5),
         ('Eclipse', 224000, 'C#', '5.6', 5)]),

        [('Python', 1),
         ('Java', 1),
         ('C++', 1),
         ('JavaScript', 1),
         ('Swift', 1),
         ('C#', 2)])

def test_task3(self):
    self.assertEqual(task3(
        [('Python', 0, 'Visual Studio'),
         ('C#', 5, 'Visual Studio'),
         ('C', 6, 'Visual Studio'),
         ('F#', 8, 'Visual Studio'),
         ('Python', 0, 'VS Code'),
         ('C++', 2, 'VS Code'),
         ('JavaScript', 3, 'VS Code'),
         ('HTML', 7, 'VS Code'),
         ('F#', 8, 'VS Code'),
         ('JavaScript', 3, 'WebStorm'),
         ('Python', 0, 'PyCharm'),
         ('Swift', 4, 'Xcode'),
         ('Java', 1, 'NetBeans'),
         ('C++', 2, 'NetBeans'),
         ('C++', 2, 'Eclipse'),
         ('C#', 5, 'Eclipse'),
         ('C', 6, 'Eclipse')]),

        {'C++': ['VS Code', 'NetBeans', 'Eclipse'],
         'C#': ['Visual Studio', 'Eclipse'],
         'C': ['Visual Studio', 'Eclipse']})

if __name__ == "__main__":
    main()

```

Результаты прохождения тестов

```
[Running] python -u "c:\BKIT\rk2\test.py"
```

```
-----  
1  
-----  
( 'Visual Studio', 1450000, 'C#', '5.6')  
( 'PyCharm', 550000, 'Python', '3.10')  
( 'VS Code', 530000, 'C++', '17.2.0')  
( 'Eclipse', 224000, 'C#', '5.6')  
( 'Xcode', 150000, 'Swift', '5.6')  
( 'NetBeans', 120000, 'Java', 'v18')  
( 'WebStorm', 115000, 'JavaScript', '1.8.5')  
-----
```

```
2  
-----
```

```
( 'C#', 2)  
( 'Python', 1)  
( 'Java', 1)  
( 'C++', 1)  
( 'JavaScript', 1)  
( 'Swift', 1)  
-----
```

```
3  
-----
```

```
1  
-----  
( 'Visual Studio', 1450000, 'C#', '5.6')  
( 'PyCharm', 550000, 'Python', '3.10')  
( 'VS Code', 530000, 'C++', '17.2.0')  
( 'Eclipse', 224000, 'C#', '5.6')  
( 'Xcode', 150000, 'Swift', '5.6')  
( 'NetBeans', 120000, 'Java', 'v18')  
( 'WebStorm', 115000, 'JavaScript', '1.8.5')  
-----
```

```
2  
-----
```

```
( 'C#', 2)  
( 'Python', 1)  
( 'Java', 1)  
( 'C++', 1)  
( 'JavaScript', 1)  
( 'Swift', 1)  
..  
-----
```

```
3  
-----
```

```
Ran 3 tests in 0.001s
```

```
OK
```

```
[Done] exited with code=0 in 0.22 seconds
```