

Рубежный контроль №2

Текст программы

Rk.py

```
class Pr_lang:
    def __init__(self, id, name, popular, type):
        self.id = id
        self.name = name
        self.popular = popular
        self.type = type

class Operator:

    def __init__(self, id, name, work, lang_id):
        self.id = id
        self.name = name
        self.work = work
        self.lang_id = lang_id

class Have:
    def __init__(self, id1, id2):
        self.lang_id = id1
        self.oper_id = id2

def number1(): #список операций в каждом языке
    res = []
    for lang in pr_lang:
        sp = [x.name for x in opers if x.lang_id == lang.id]
        res.append(f"список операций в языке {lang.name}: {sp}")
    return res

def number2(): #список яп с подсчетом операторов в них, сортировка по
    количеству операторов
```

```

res = []
for lang in pr_lang:
    res.append([len([x.name for x in opers if x.lang_id == lang.id]),lang.name])
res.sort()
return [f"список операций в языке {x[1]}: {x[0]}" for x in res]

def number3(type): #список операций если язык подходит
    sp = [x for x in pr_lang if x.type == type]
    return [f"список операций в языке {x.name}: {[y.name for y in opers if
y.lang_id == x.id]}" for x in sp]

pr_lang = [
    Pr_lang(1,"python",1000000, "oop"),
    Pr_lang(2,"c",3000000, "proced"),
    Pr_lang(3,"java",100000000, "oop")
]

opers = [Operator(1, "**", "all nums", 1),
         Operator(2, "%", "all nums", 1),
         Operator(3, "/", "int nums", 2),
         Operator(4, "-", "all nums", 3)
        ]

have = [Have(x.lang_id, x.id) for x in opers]

def main():
    print(number1())
    print(number2())
    print(number3("oop"))

if __name__ == '__main__':
    main()

```

unit_tests.py

```
import unittest
from Rk import Pr_lang, Operator, Have, number1, number2, number3
```

```
class Tests(unittest.TestCase):
```

```
    def setUp(self):
```

```
        self.pr_lang = [
            Pr_lang(1, "python", 1000000, "oop"),
            Pr_lang(2, "c", 3000000, "proced"),
            Pr_lang(3, "java", 100000000, "oop")
        ]
```

```
        self.operators = [
            Operator(1, "**", "all nums", 1),
            Operator(2, "%", "all nums", 1),
            Operator(3, "//", "int nums", 2),
            Operator(4, "-", "all nums", 3)
        ]
```

```
        self.have = [Have(oper.lang_id, oper.id) for oper in self.operators]
```

```
    def test_num1(self):
```

```
        exp_out = [
            "список операций в языке python: ['**', '%']",
            "список операций в языке c: ['//']",
            "список операций в языке java: ['-']"
        ]
```

```
        res = number1()
```

```
        self.assertEqual(sorted(res), sorted(exp_out))
```

```
    def test_num2(self):
```

```
        exp_out = [
            "список операций в языке java: 1",
            "список операций в языке c: 1",
            "список операций в языке python: 2"
        ]
```

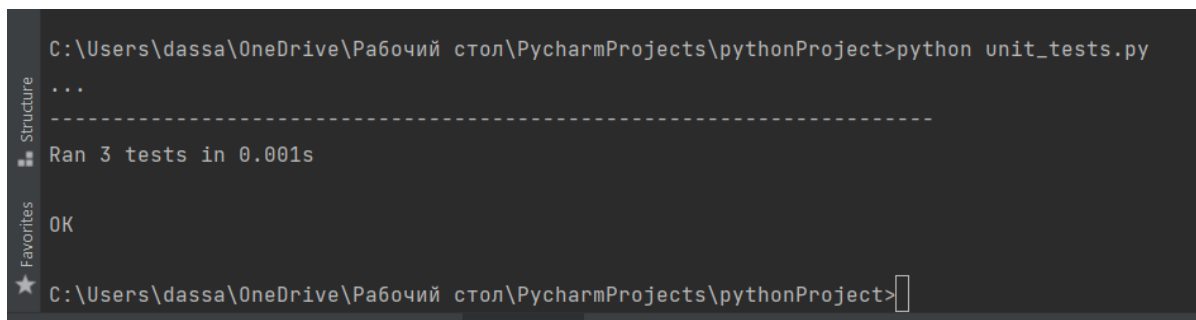
```
        res = number2()
```

```
self.assertEqual(sorted(res), sorted(exp_out))

def test_number3(self):
    exp_out = [
        "список операций в языке python: ['**', '%']",
        "список операций в языке java: ['-']"
    ]
    res = number3("oop")
    self.assertEqual(sorted(res), sorted(exp_out))

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

Результат выполнения тестов:



```
C:\Users\dassa\OneDrive\Рабочий стол\PycharmProjects\pythonProject>python unit_tests.py
...
-----
Ran 3 tests in 0.001s
OK
C:\Users\dassa\OneDrive\Рабочий стол\PycharmProjects\pythonProject>
```