

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по РК №2
Вариант запросов: Г
Вариант предметной области: 12

Выполнил:
студент группы ИУ5-34Б
Петров Никита

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю. Е.

Москва, 2023 г.

Вариант запросов Г. Предметная область 12.

1. «Язык программирования» и «Средство разработки» связаны соотношением один-ко-многим. Выведите список всех языков, начинающихся с буквы 'P'.
2. «Язык программирования» и «Средство разработки» связаны соотношением один-ко-многим. Выведите список средств разработки с максимальным годом создания поддерживаемого языка, отсортированный по максимальному году создания.
3. «Язык программирования» и «Средства разработки» связаны соотношением многие-ко-многим. Выведите список всех связанных языков программирования и средств разработки, отсортированный по средам разработки, сортировка по языкам программирования произвольная.

Задание

- 1) Проведите рефакторинг текста программы рубежного контроля №1 таким образом, чтобы он был пригоден для модульного тестирования.
- 2) Для текста программы рубежного контроля №1 создайте модульные тесты с применением TDD - фреймворка (3 теста).

Листинг программы.

Файл main.py

```
from operator import itemgetter
```

```
class Lang:
```

```
    def __init__(self, id, name, year, ide_id):
        self.id = id
        self.name = name
        self.year = year
        self.ide_id = ide_id
```

```
    def __str__(self):
        return f"{self.name}"
```

```
class IDE:
```

```
    def __init__(self, id, name):
        self.id = id
        self.name = name
```

```
    def __str__(self):
        return f"{self.name}"
```

```

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

    def __str__(self):
        return f"{self.ide_id} - {self.lang_id}"

Langs = [Lang(1, 'C++', 1985, 1),
          Lang(2, 'Python', 1991, 2),
          Lang(3, 'Java', 1995, 3),
          Lang(4, 'C', 1972, 4),
          Lang(5, 'Go', 2003, 4),
          Lang(6, 'PHP', 1995, 5),
          Lang(7, 'JavaScript', 1995, 4)
          ]

IDEs = [IDE(1, 'CLion'),
        IDE(2, 'PyCharm'),
        IDE(3, 'IntelliJ'),
        IDE(4, 'VSCode'),
        IDE(5, 'PHP_sreda')]

Lang_IDE = [LangIDE(1, 1),
            LangIDE(2, 2),
            LangIDE(3, 3),
            LangIDE(1, 4),
            LangIDE(4, 4),
            LangIDE(5, 4),
            LangIDE(6, 5),
            LangIDE(7, 4)]

def main():
    one_to_many = [(l.name, l.year, i.name)
                    for l in Langs
                    for i in IDEs
                    if l.ide_id == i.id]
    many_to_many_temp = [(i.name, li.ide_id, li.lang_id)
                          for i in IDEs
                          for li in Lang_IDE
                          if i.id == li.ide_id
                          ]
    many_to_many = [(l.name, l.year, ide_name)
                    for ide_name, ide_id, lang_id in many_to_many_temp
                    for l in Langs if l.id == lang_id]
    print("Задание 1")
    ans1 = []
    for i in one_to_many:
        if i[0][0] == "P":
            ans1.append(i[0])
    print(ans1)

    print("Задание 2")
    ans2_uns = []
    for instr in IDEs:
        instr_langs = list(filter(lambda i: i[2] == instr.name, one_to_many))
        if len(instr_langs) > 0:
            years = [year for _, year, _ in instr_langs]
            max_ = max(years)

```

```

        ans2_uns.append((instr.name, max_))
ans2 = sorted(ans2_uns, key=itemgetter(1))
print(ans2)

print('Задание 3')
print(sorted(many_to_many, key=lambda item: (item[2], item[1])))

if __name__ == '__main__':
    main()

```

Файл tests.py

```

import unittest
from main import *

class Rk2_test(unittest.TestCase):
    Langs = [Lang(1, 'C++', 1985, 1),
             Lang(2, 'Python', 1991, 2),
             Lang(3, 'Java', 1995, 3),
             Lang(4, 'C', 1972, 4),
             Lang(5, 'Go', 2003, 4),
             Lang(6, 'PhP', 1995, 5),
             Lang(7, 'JavaScript', 1995, 4)
            ]

    IDEs = [IDE(1, 'CLion'),
            IDE(2, 'PyCharm'),
            IDE(3, 'IntelliJ'),
            IDE(4, 'VSCode'),
            IDE(5, 'PhP_sreda')]

    Lang_IDE = [LangIDE(1, 1),
                LangIDE(2, 2),
                LangIDE(3, 3),
                LangIDE(1, 4),
                LangIDE(4, 4),
                LangIDE(5, 4),
                LangIDE(6, 5),
                LangIDE(7, 4)]

    def test_1(self):
        one_to_many = [(l.name, l.year, i.name)
                        for l in Langs
                        for i in IDEs
                        if l.ide_id == i.id]

        ans1 = []
        for i in one_to_many:
            if i[0][0] == "P":
                ans1.append(i[0])
        self.assertEqual(ans1, ['Python', 'PhP'])

    def test_2(self):
        one_to_many = [(l.name, l.year, i.name)
                        for l in Langs
                        for i in IDEs
                        if l.ide_id == i.id]

        ans2_uns = []
        for instr in IDEs:
            instr_langs = list(filter(lambda i: i[2] == instr.name, one_to_many))

```

```

if len(instr_langs) > 0:
    years = [year for _, year, _ in instr_langs]
    max_ = max(years)
    ans2_uns.append((instr.name, max_))
ans2 = sorted(ans2_uns, key=itemgetter(1))
self.assertEqual(ans2, [('CLion', 1985), ('PyCharm', 1991), ('IntelliJ', 1995),
                        ('PhP_sreda', 1995), ('VSCode', 2003)])

def test_3(self):
    many_to_many_temp = [(i.name, li.ide_id, li.lang_id)
                        for i in IDEs
                        for li in Lang_IDE
                        if i.id == li.ide_id
                        ]
    many_to_many = [(l.name, l.year, ide_name)
                    for ide_name, ide_id, lang_id in many_to_many_temp
                    for l in Langs if l.id == lang_id]
    ans3 = sorted(many_to_many, key=lambda item: (item[2], item[1]))
    self.assertEqual(ans3, [('C++', 1985, 'CLion'), ('Java', 1995, 'IntelliJ'), ('PhP', 1995, 'PhP_sreda'),
                            ('Python', 1991, 'PyCharm'), ('C', 1972, 'VSCode'), ('C++', 1985, 'VSCode'),
                            ('JavaScript', 1995, 'VSCode'), ('Go', 2003, 'VSCode')])

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

```

Результат выполнения программы


```

class RK2_test(unittest.TestCase):
    Langs = [Lang(1, 'C++', 1985, 1),
             Lang(2, 'Python', 1991, 2),
             Lang(3, 'Java', 1995, 3),
             Lang(4, 'C', 1972, 4),
             Lang(5, 'Go', 2003, 4),
             Lang(6, 'PhP', 1995, 5),
             Lang(7, 'JavaScript', 1995, 4)
            ]

    IDEs = [IDE(1, 'CLion'),
            IDE(2, 'PyCharm'),
            IDE(3, 'IntelliJ'),
            IDE(4, 'VSCode'),
            IDE(5, 'PhP_sreda')]

```

5 ms

>>  Tests passed: 3 of 3 tests – 5 ms

Ran 3 tests in 0.007s

OK

Process finished with exit code 0