

**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: В
Вариант предметной области: 12

Выполнил:
студент группы ИУ5-33Б
Иванов Николай

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю. Е.

Москва, 2023 г.

Вариант запросов В. Предметная область 12.

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

Условие РК2 по ПиКЯП:

Рубежный контроль представляет собой разработку тестов на языке Python.

- 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.

Main.py

```
from operator import itemgetter

class Lang:
    def __init__(self, id, name, cr_year, instr_id):
        self.id = id
        self.name = name
        self.cr_year = cr_year
        self.instr_id = instr_id

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

```

class Lang_Instr:
    def __init__(self, instr_id, lang_id):
        self.instr_id = instr_id
        self.lang_id = 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)
        ]

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

Lang_instrs = [Lang_Instr(1, 1),
               Lang_Instr(2, 2),
               Lang_Instr(3, 3),
               Lang_Instr(4, 1),
               Lang_Instr(4, 4),
               Lang_Instr(4, 5),
               Lang_Instr(5, 6),
               Lang_Instr(4, 7)
              ]

def main():
    one_to_many = [(l.name, l.cr_year, i.name)
                   for l in Langs
                   for i in Instrs
                   if l.instr_id == i.id]
    many_to_many_temp = [(i.name, li.instr_id, li.lang_id)
                         for i in Instrs
                         for li in Lang_instrs
                         if i.id == li.instr_id
                        ]
    many_to_many = [(l.name, l.cr_year, instr_name)
                   for instr_name, instr_id, lang_id in many_to_many_temp
                   for l in Langs if l.id == lang_id]

    print('Task 1')
    res_11 = []
    for i in one_to_many:
        if i[0][0] == 'J':
            res_11.append(i)
    print(res_11)

    print('Task 2')
    res_12_unsorted = []
    for instr in Instrs:
        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]
            min_ = min(years)

```

```

        res_12_unsorted.append((instr.name, min_))
    res_12 = sorted(res_12_unsorted, key=itemgetter(1))
    print(res_12)

    print('Task 3')
    print(sorted(many_to_many, key=lambda item: (item[0], item[1])))

if __name__ == '__main__':
    main()

```

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, 'NotJavaScript', 1995, 4)
            ]
    Instrs = [Instr(1, 'CLion'),
              Instr(2, 'PyCharm'),
              Instr(3, 'IntelliJ'),
              Instr(4, 'VSCode'),
              Instr(5, 'PhP_sreda')]

    Lang_instrs = [Lang_Instr(1, 1),
                   Lang_Instr(2, 2),
                   Lang_Instr(3, 3),
                   Lang_Instr(4, 1),
                   ]

    def test_1(self):
        one_to_many = [(l.name, l.cr_year, i.name)
                        for l in self.Langs
                        for i in self.Instrs
                        if l.instr_id == i.id]
        res_11 = []
        for i in one_to_many:
            if i[0][0] == 'J':
                res_11.append(i)
        self.assertEqual(res_11, [('Java', 1995, 'IntelliJ')])

    def test_2(self):
        one_to_many = [(l.name, l.cr_year, i.name)
                        for l in self.Langs
                        for i in self.Instrs
                        if l.instr_id == i.id]
        res_12_unsorted = []
        for instr in self.Instrs:

```

```

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]
    min_ = min(years)
    res_12_unsorted.append((instr.name, min_))
res_12 = sorted(res_12_unsorted, key=itemgetter(1))
self.assertEqual(res_12, [('VSCode', 1972), ('CLion', 1985), ('PyCharm', 1991), ('IntelliJ', 1995), ('PHP_sreda', 1995)])

def test_3(self):
    many_to_many_temp = [(i.name, li.instr_id, li.lang_id)
                          for i in self.Instrs
                          for li in self.Lang_instrs
                          if i.id == li.instr_id
                          ]
    many_to_many = [(l.name, l.cr_year, instr_name)
                    for instr_name, instr_id, lang_id in many_to_many_temp
                    for l in Langs if l.id == lang_id]
    res = sorted(many_to_many, key=lambda item: (item[0], item[1]))
    self.assertEqual(res, [('C++', 1985, 'CLion'), ('C++', 1985, 'VSCode'), ('Java', 1995, 'IntelliJ'), ('Python', 1991, 'PyCharm')])

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

```

Результаты работы.

Все тесты пройдены успешно

```

main.py  tests.py x
32:         for i in one_to_many:
33:             if i[0][0] == 'J':
34:                 res_11.append(i)
35:         self.assertEqual(res_11, second: [('Java', 1995, 'IntelliJ')])
36:
37: new *
38: def test_2(self):
39:     one_to_many = [(l.name, l.cr_year, i.name)
40:                   for l in self.Lang
41:                   for i in self.Instrs
42:                   if l.instr_id == i.id]
43:     res_12_unsorted = []
44:     for instr in self.Instrs:
45:         instr_langs = list(filter(lambda i: i[2] == instr.name, one_to_many))
46:         if len(instr_langs) > 0:
47:             years = [year for _, year, _ in instr_langs]
48:             min_ = min(years)
49:             res_12_unsorted.append((instr.name, min_))
50:     res_12 = sorted(res_12_unsorted, key=itemgetter(1))
51:     self.assertEqual(res_12, second: [('VSCode', 1972), ('CLion', 1985), ('PyCharm', 1991), ('IntelliJ', 1995), ('PHP_sreda', 1995)])

Rk2_test  test_2()

ests.py x
2 ms  ✓ Tests passed: 3 of 3 tests – 2 ms

C:\Users\Asus\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:/Program Files/JetBrains/PyCharm Community Edition 2023.2
Testing started at 0:25 ...
Launching unittests with arguments python -m unittest C:\Users\Asus\Desktop\rk1\tests.py in C:\Users\Asus\Desktop\rk1

Ran 3 tests in 0.002s

OK

Process finished with exit code 0

```

Все тесты выдали ошибку

```
Project
Run Python tests in tests.py x
Tests failed: 3 of 3 tests - 61ms
C:\Users\Asus\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:/Program Files/JetBrains/PyCharm Community Edition 2023.2.1/plugins/python-ce/helpers/pycharm/_jb_unittest_runner.py" --path C:\Users\Asus\Desktop\rsl\tests.py
Testing started at 8:28 ...
Launching unittests with arguments python -m unittest C:\Users\Asus\Desktop\rsl\tests.py in C:\Users\Asus\Desktop\rsl

Ran 3 tests in 0.062s

FAILED (failures=3)

[('Java', 1995, 'IntelliJ')] != [('Java', 1995, 'IntelliJ'), ('JavaScript', 1995, 'VSCode')]

Expected: [('Java', 1995, 'IntelliJ'), ('JavaScript', 1995, 'VSCode')]
Actual: [('Java', 1995, 'IntelliJ')]
Click to see difference

Traceback (most recent call last):
  File "C:\Users\Asus\Desktop\rsl\tests.py", line 25, in test_1
    self.assertEqual(res, 1, [('Java', 1995, 'IntelliJ')])
AssertionError: Lists differ: [('Java', 1995, 'IntelliJ'), ('JavaScript', 1995, 'VSCode')] != [('Java', 1995, 'IntelliJ')]

First list contains 1 additional elements.
First extra element 1:
('JavaScript', 1995, 'VSCode')

- [('Java', 1995, 'IntelliJ'), ('JavaScript', 1995, 'VSCode')]
+ [('Java', 1995, 'IntelliJ')]

[('VSCode', 1972), ('CLion', 1985), ('IntelliJ', 1995), ('PHP_sreda', 1995)] != [('VSCode', 1972), ('CLion', 1985), ('PyCharm', 1991), ('IntelliJ', 1995), ('PHP_sreda', 1995)]
Click to see difference

Traceback (most recent call last):
  File "C:\Users\Asus\Desktop\rsl\tests.py", line 58, in test_2
    self.assertEqual(res, 12, [('VSCode', 1972), ('CLion', 1985), ('IntelliJ', 1995), ('PHP_sreda', 1995)])
AssertionError: Lists differ: [('VS[27 chars]', ('PyCharm', 1991), ('IntelliJ', 1995), ('PHP_sreda', 1995))] != [('VS[27 chars]', ('IntelliJ', 1995), ('PHP_sreda', 1995))]
```

```
Tests failed: 3 of 3 tests - 61ms

First list contains 1 additional elements.
First extra element 4:
('PHP_sreda', 1995)

+ [('VSCode', 1972), ('CLion', 1985), ('IntelliJ', 1995), ('PHP_sreda', 1995)]
- [('VSCode', 1972), ('CLion', 1985), ('PyCharm', 1991), ('IntelliJ', 1995), ('PHP_sreda', 1995)]

[('C++', 1985, 'CLion'), ('C++', 1985, 'VSCode'), ('Python', 1991, 'PyCharm')] != [('C++', 1985, 'CLion'), ('C++', 1985, 'VSCode'), ('Java', 1995, 'IntelliJ'), ('Python', 1991, 'PyCharm')]
Click to see difference

Traceback (most recent call last):
  File "C:\Users\Asus\Desktop\rsl\tests.py", line 62, in test_3
    self.assertEqual(res, [('C++', 1985, 'CLion'), ('C++', 1985, 'VSCode'), ('Python', 1991, 'PyCharm')])
AssertionError: Lists differ: [('C++[38 chars]ode'), ('Java', 1995, 'IntelliJ'), ('Python', 1991, 'PyCharm')] != [('C++[38 chars]ode'), ('Python', 1991, 'PyCharm')]

First differing element 2:
('Java', 1995, 'IntelliJ')
('Python', 1991, 'PyCharm')

First list contains 1 additional elements.
First extra element 3:
('Python', 1991, 'PyCharm')

+ [('C++', 1985, 'CLion'), ('C++', 1985, 'VSCode'), ('Python', 1991, 'PyCharm')]
- [('C++', 1985, 'CLion'), ('C++', 1985, 'VSCode'), ('Java', 1995, 'IntelliJ'), ('Python', 1991, 'PyCharm')]

Process finished with exit code 1
```