**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: B

Вариант предметной области: 12

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-33Б |  | преподаватель каф. ИУ5 |
| Иванов Николай |  | Гапанюк Ю. Е. |
|  |  |  |

Москва, 2023 г.

**Вариант запросов B. Предметная область 12.**

# «Язык программирования» и «Средство разработки» связаны соотношением один-ко-многим. Выведите список всех языков, начинающихся с буквы ‘J’.

# «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.

# «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

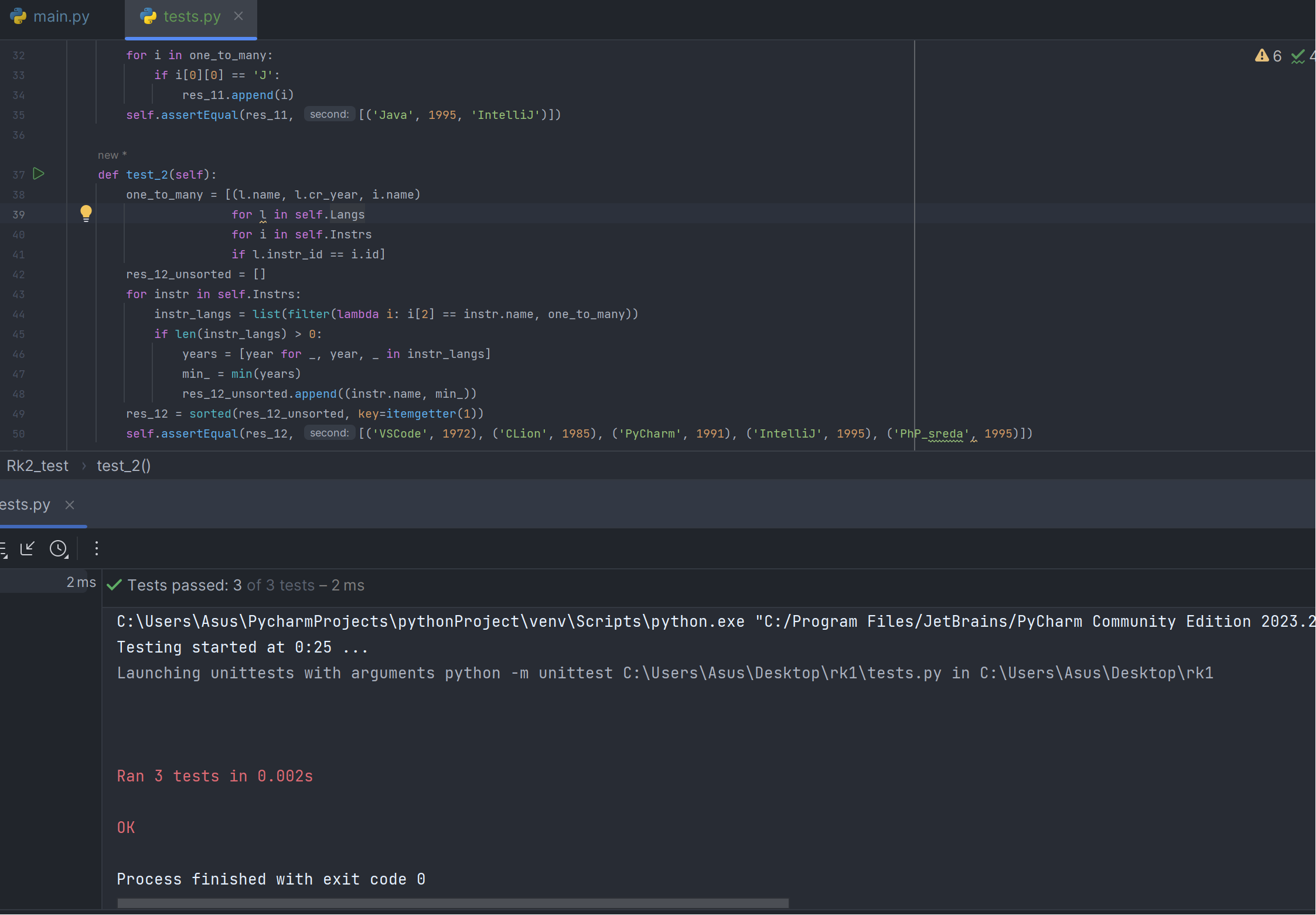
from operator import itemgetter  
  
  
class Lang:  
 def \_\_init\_\_(self, id, name, cr\_year, instr\_id):  
 self.id = id  
 self.name = name  
 self.cr\_year = cr\_year  
 self.instr\_id = instr\_id  
  
  
class Instr:  
 def \_\_init\_\_(self, id, name):  
 self.id = id  
 self.name = name  
  
  
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



Все тесты выдали ошибку

