

**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 11

Выполнил:

студент группы ИУ5-33Б
Захаров Ф.А.

Подпись и дата:

Проверил:

преподаватель каф.ИУ5
Гапанюк Ю.Е.

Подпись и дата:

Москва, 2023г.

Текст программы

Main.py

```
from operator import itemgetter

class Program:
    """Программа"""
    def __init__(self, id, name, comp_id, size_in_gb):
        self.id = id
        self.name = name
        self.comp_id = comp_id
        self.size_in_gb = size_in_gb

class Computer:
    """Компьютер"""
    def __init__(self, id, model, RAM, owner):
        self.id = id
        self.model = model
        self.RAM = RAM
        self.owner = owner

class ProgramComputer:
    """Связь Программы и Компьютера"""

    def __init__(self, pr_id, comp_id):
        self.pr_id = pr_id
        self.comp_id = comp_id

def a1_solution(one_to_many):
    return sorted(one_to_many, key=itemgetter(2))

def a2_solution(one_to_many):
    res_a2_unsorted = []
    for c in set(item[2] for item in one_to_many):
        c_emps = list(filter(lambda i: i[2] == c, one_to_many))
        if c_emps:
            c_cores = sum(core for _, core, _ in c_emps)
            res_a2_unsorted.append((c, c_cores))
    res_a2 = sorted(res_a2_unsorted, key=itemgetter(1), reverse=True)
    return res_a2

def a3_solution(many_to_many):
    res_a3 = {}
    for p in set(item[0] for item in many_to_many):
        if 'Excel' in p: # Fix the typo here
            p_computers = list(filter(lambda i: i[0] == p, many_to_many))
            p_computer_info = [owner for _, _, owner in p_computers]
            res_a3[p] = p_computer_info
    return res_a3

def main():
    """Основная функция"""
    programs = [
        Program(1, 'Microsoft Word', 1, 2.0),
```

```

Program(2, 'Microsoft Excel', 1, 1.5),
Program(3, 'Google Docs', 3, 1.0),
Program(4, 'LibreOffice Writer', 1, 1.8),
Program(5, 'OpenOffice Calc', 4, 1.2)]

computers = [
    Computer(1, 'Model A-2000X', '8 GB DDR4', 'Иванов Иван Иванович'),
    Computer(2, 'UltraBook Pro 15S', '16 GB DDR4', 'Петрова Анна Сергеевна'),
    Computer(3, 'GamingBeast X9000', '32 GB DDR4', 'Смирнов Сергей
Владимирович'),
    Computer(4, 'OfficeMaster 500', '64 GB DDR4', 'Козлова Екатерина
Павловна'),
    Computer(5, 'PerformanceElite 3000', '128 GB DDR4', 'Михайлов Алексей
Дмитриевич')
]

pr_comp = [
    ProgramComputer(1, 1), ProgramComputer(1, 2), ProgramComputer(1, 4),
    ProgramComputer(2, 3), ProgramComputer(2, 1),
    ProgramComputer(3, 4), ProgramComputer(4, 5),
    ProgramComputer(3, 5)
]

one_to_many = [(p.name, p.size_in_gb, c.owner, c.model)
                for p in programs
                for c in computers
                if p.comp_id == c.id]

many_to_many_temp = [(p.name, p_c.pr_id, p_c.comp_id)
                     for p_c in pr_comp
                     for p in programs
                     if p.id == p_c.pr_id]

many_to_many = [(pr_name, c.model, c.owner)
                for pr_name, dep_id, p_c_id in many_to_many_temp
                for c in computers if c.id == p_c_id]

print('Задание A1')
print(a1_solution(one_to_many))

print('\nЗадание A2')
print(a2_solution(one_to_many))

print('\nЗадание A3')
print(a3_solution(many_to_many))

if __name__ == '__main__':
    main()

```

Test.py

```
import unittest
from main import *
class Test_Program(unittest.TestCase):
    # Глобальные переменные
    programs = [
        Program(1, 'Microsoft Word', 1, 2.0),
        Program(2, 'Microsoft Excel', 1, 1.5),
        Program(3, 'Google Docs', 3, 1.0),
        Program(4, 'LibreOffice Writer', 1, 1.8),
        Program(5, 'OpenOffice Calc', 4, 1.2)
    ]

    computers = [
        Computer(1, 'Model A-2000X', '8 GB DDR4', 'Иванов Иван Иванович'),
        Computer(2, 'UltraBook Pro 15S', '16 GB DDR4', 'Петрова Анна Сергеевна'),
        Computer(3, 'GamingBeast X9000', '32 GB DDR4', 'Смирнов Сергей Владимирович'),
        Computer(4, 'OfficeMaster 500', '64 GB DDR4', 'Козлова Екатерина Павловна'),
        Computer(5, 'PerformanceElite 3000', '128 GB DDR4', 'Михайлов Алексей Дмитриевич')
    ]

    pr_comp = [
        ProgramComputer(1, 1), ProgramComputer(1, 2), ProgramComputer(1, 4),
        ProgramComputer(2, 3), ProgramComputer(2, 1),
        ProgramComputer(3, 4), ProgramComputer(4, 5),
        ProgramComputer(3, 5)
    ]

    def test_A1(self):
        one_to_many = [(p.name, p.size_in_gb, c.owner, c.model)
                        for p in self.programs
                        for c in self.computers
                        if p.comp_id == c.id]

        self.assertEqual(a1_solution(one_to_many),
                         [('Microsoft Word', 2.0, 'Иванов Иван Иванович', 'Model A-2000X'),
                          ('Microsoft Excel', 1.5, 'Иванов Иван Иванович', 'Model A-2000X'),
                          ('LibreOffice Writer', 1.8, 'Иванов Иван Иванович', 'Model A-2000X'),
                          ('OpenOffice Calc', 1.2, 'Козлова Екатерина Павловна', 'OfficeMaster 500'),
                          ('Google Docs', 1.0, 'Смирнов Сергей Владимирович', 'GamingBeast X9000')])

    def test_A2(self):
        one_to_many = [(p.name, p.size_in_gb, c.owner, c.model)
                        for p in self.programs
                        for c in self.computers
                        if p.comp_id == c.id]

        self.assertEqual(a2_solution(one_to_many),
                         [('Иванов Иван Иванович', 5.3),
                          ('Козлова Екатерина Павловна', 1.2),
                          ('Смирнов Сергей Владимирович', 1.0)])
```

```

def test_A3(self):
    many_to_many_temp = [(p.name, p_c.pr_id, p_c.comp_id)
                          for p_c in self.pr_comp
                          for p in self.programs
                          if p.id == p_c.pr_id]

    many_to_many = [(pr_name, c.model, c.owner)
                    for pr_name, dep_id, p_c_id in many_to_many_temp
                    for c in self.computers if c.id == p_c_id]

    self.assertDictEqual(a3_solution(many_to_many), {'Microsoft Excel':
['Смирнов Сергей Владимирович', 'Иванов Иван Иванович']})

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

```