**Московский государственный технический университет**

**им. Н. Э. Баумана**

Факультет «Информатика и системы управления»

Отчёт по рубежному контролю №1  
по курсу «Разработка интернет-приложений»

Выполнил: Проверил:

студент группы ИУ5-51Б преподаватель каф. ИУ5 Гришин С. В. Гапанюк Ю. Е.

Подпись и дата: Подпись и дата:  
22.10.2021 22.10.2021

Москва, 2021 г.

**Описание задания:**

1) Необходимо создать два класса данных в соответствии с вариантом предметной области, которые связаны отношениями один-ко-многим и многие-ко-многим.

2) Необходимо создать списки объектов классов, содержащих тестовые данные, таким образом, чтобы первичные и вторичные ключи соответствующих записей были связаны по идентификаторам.

|  |  |  |
| --- | --- | --- |
| **№ варианта** | **Класс 1** | **Класс 2** |
| 11 | Программа | Компьютер |

**Вариант А:**

1. «Программа» и «Компьютер» связаны соотношением один-ко-многим. Выведите список всех связанных программ и компьютеров, отсортированный по компьютерам, сортировка по программам произвольная.
2. «Программа» и «Компьютер» связаны соотношением один-ко-многим. Выведите список компьютеров с суммарной памятью программ в каждом компьютере, отсортированный по суммарной памяти.
3. «Программа» и «Компьютер» связаны соотношением многие-ко-многим. Выведите список всех компьютеров, у которых в названии присутствует буква «А», и список работающих в них программ.

**Текст программы:**

from operator import itemgetter

class PC:

    """Компьютер"""

    def \_\_init\_\_(self, id, name, os):

        self.id = id

        self.pcname = name

        self.os = os

class Program:

    """Программа"""

    def \_\_init\_\_(self, id,name,memory, pc\_id):

        self.id = id

        self.programname = name

        self.memory=memory

        self.pc\_id = pc\_id

class ProgramPC:

    """

    'Программы компьютера' для реализации

    связи многие-ко-многим

    """

    def \_\_init\_\_(self, pc\_id, program\_id):

        self.pc\_id = pc\_id

        self.program\_id = program\_id

*# Программы*

programs = [

    Program(1, 'Visual Studio',30,1),

    Program(2, 'Visual Studio',30,2),

    Program(3, 'Visual Studio',30,3),

    Program(4, 'Visual Studio',30,5),

    Program(5, 'Qlick Sense',3,2),

    Program(6, 'SQL Server Managment Studio',1,1),

    Program(7, 'SQL Server Managment Studio',1,5),

    Program(8, 'Microsoft Office 2016',9,1),

    Program(9, 'Adobe Acrobat Reader',0.1,4),

    Program(10, 'Adobe Acrobat Reader',0.1,5),

    Program(11, 'Safari',0.2,3),

    Program(12, 'MATLAB',6,2),

    Program(13, 'OpenOffice',0.2,4),

    Program(14, 'Visual Studio',30,4),

]

*# Сотрудники*

pcs = [

    PC(1, 'Asus', 'Windows'),

    PC(2, 'Acer', 'Windows'),

    PC(3, 'Apple','MacOS'),

    PC(4, 'Dell', 'Linux'),

    PC(5, 'Lenovo', 'Windows'),

]

pcs\_programs = [

    ProgramPC(1,1),

    ProgramPC(1,6),

    ProgramPC(1,8),

    ProgramPC(2,2),

    ProgramPC(2,5),

    ProgramPC(2,12),

    ProgramPC(3,3),

    ProgramPC(3,11),

    ProgramPC(4,9),

    ProgramPC(4,13),

    ProgramPC(5,4),

    ProgramPC(5,7),

    ProgramPC(5,10),

]

def main():

    """Основная функция"""

    one\_to\_many = [(d.programname, d.memory, e.pcname, e.os)

        for d in programs

        for e in pcs

        if d.pc\_id==e.id]

    print('Задание А1')

    res\_1 = sorted(one\_to\_many, key=itemgetter(2))

    print(res\_1)

    print("\nЗадание А2:")

    res2 = []

    for p in pcs:

        a = sum([k.memory for k in programs if (k.pc\_id == p.id)])

        res2.append([p.pcname, a])

    res2=sorted(res2,key=itemgetter(1))

    print(res2)

    print("\nЗадание А3:")

    many\_to\_many\_temp = [(l.pcname, l.os, m.program\_id)

        for l in pcs

        for m in pcs\_programs

        if (l.id == m.pc\_id)

    ]

    many\_to\_many = [(j.programname, j.memory, pcname, os)

        for pcname, os, program\_id in many\_to\_many\_temp

        for j in programs

        if (j.id == program\_id)

    ]

    res3 = {}

    for l in pcs:

        if 'A' in l.pcname:

            Pcs = list(filter(lambda x: x[2] == l.pcname ,many\_to\_many))

            Programms = [programname for programname, mem, pcname, os in Pcs]

            res3[l.pcname] = Programms

    print(res3)

if \_\_name\_\_ == '\_\_main\_\_':

    main()

**Результат выполнения программы:**

