

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования



«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Факультет «Информатика и системы управления»
Курс «Базовые компоненты интернет-технологий»

Рубежный контроль №2

Выполнил:
студент группы ИУ5-33Б
Голумин Н. Д.

Проверил:
Преподаватель кафедры ИУ-5
Гапанюк Ю. Е.

2022 г.

Оглавление

1. Описание задания.....	3
2. Текст программы.....	Ошибка! Закладка не определена.
3. Экранные формы с примерами выполнения программы	6

Описание задания

Вариант Е. Предметная область 8

1. «Компьютер» и «Жесткий диск» связаны соотношением один-многим. Выведите список всех компьютеров, у которых в названии присутствует слово «Core i5», и список жестких дисков в них.
2. «Компьютер» и «Жесткий диск» связаны соотношением один-многим. Выведите список компьютеров со средним размером жесткого диска в каждом компьютере, отсортированный по среднему размеру жесткого диска в каждом компьютере. Эта величина должна быть округлена до 2 знака после запятой.
3. «Компьютер» и «Жесткий диск» связаны соотношением многие-многим. Выведите список всех жестких дисков, у которых название начинается с буквы «S», и названия их книг.

Текст программы

Текст программы из РК1 после рефакторинга

```
from operator import itemgetter
from math import inf

class HardDrive:
    """Жёсткий диск"""

    def __init__(self, id, model, size, computer_id):
        self.id = id
        self.model = model
        self.size = size
        self.computer_id = computer_id

class Computer:
    """Компьютер"""

    def __init__(self, id, processor):
        self.id = id
        self.processor = processor

class HDComp:
    """
    'Жёсткий диск компьютера' для реализации
```

```
СВЯЗИ МНОГИЕ-КО-МНОГИМ
"""
```

```
def __init__(self, computer_id, hard_drive_id):
    self.computer_id = computer_id
    self.hard_drive_id = hard_drive_id
```

```
computers = [
    Computer(1, 'Intel Core i3-6100'),
    Computer(2, 'Intel Core i3-7300'),
    Computer(3, 'Intel Core i5-7400'),

    Computer(11, 'Intel Core i3-7100'),
    Computer(22, 'AMD Ryzen 3 3100'),
    Computer(33, 'Intel Core i5-6500'),
]
```

```
# Жёсткие диски
```

```
hard_drives = [
    HardDrive(1, 'Western Digital Blue WD10EZEX', 1024, 1),
    HardDrive(2, 'Seagate Barracuda ST500LM030', 500, 2),
    HardDrive(3, 'Western Digital Purple WD40PURX', 4096, 3),
    HardDrive(4, 'Seagate SkyHawk ST2000VX008', 2048, 3),
    HardDrive(5, 'Western Digital Blue WD5000LPCX', 500, 3),
]
```

```
hds_comps = [
    HDComp(1, 1),
    HDComp(2, 2),
    HDComp(3, 3),
    HDComp(3, 4),
    HDComp(3, 5),

    HDComp(11, 1),
    HDComp(22, 2),
    HDComp(33, 3),
    HDComp(33, 4),
    HDComp(33, 5),
]
```

```
#функция вычисления среднего значения с округлением до двух знаков после запятой для
второго теста
```

```
def list_average(lst):
    if not(len(lst)):return 0
    else: return round(sum(lst)/len(lst),2)
```

```
# Соединение данных один-ко-многим
```

```
one_to_many = [(hard_drive.model, hard_drive.size, computer.processor)
                for hard_drive in hard_drives
                for computer in computers
                if hard_drive.computer_id == computer.id]
```

```

# Соединение данных многие-ко-многим
many_to_many_temp = [(computer.processor, hd_c.computer_id, hd_c.hard_drive_id)
                      for computer in computers
                      for hd_c in hds_comps
                      if computer.id == hd_c.computer_id]
many_to_many = [(hard_drive.model, hard_drive.size, processor)
                 for processor, computer_id, hard_drive_id in
many_to_many_temp
                 for hard_drive in hard_drives
                 if hard_drive.id == hard_drive_id]

def task_1():
    """
    «Компьютер» и «Жесткий диск» связаны соотношением один-комногим. Выведите список
    всех компьютеров, у которых в названии
    присутствует слово «Core i5», и список жестких дисков в них.
    """

    return [(record[0], record[2]) for record in one_to_many if
record[2].startswith('Intel Core i5')]

def task_2():
    """
    «Компьютер» и «Жесткий диск» связаны соотношением один-комногим. Выведите список
    компьютеров со средним размером жесткого
    диска в каждом компьютере, отсортированный по среднему размеру
    жесткого диска в каждом компьютере. Эта величина должна быть
    округлена до 2 знака после запятой.
    """

    results = []
    for hard_drive in Computer:
        list_temp=[x[1] for x in one_to_many if x[2] == computer.processor]
        results.append((computer.processor, list_average(list_temp)))
    return sorted(results.items(), key=itemgetter(1))

def task_3():
    """
    «Компьютер» и «Жесткий диск» связаны соотношением многие-комногим. Выведите
    список всех жестких дисков, у которых название
    начинается с буквы «S», и названия их книг.
    """

```

```
    return [(record[0], record[2]) for record in many_to_many if
record[0].startswith('S')]
```

Текст модульных тестов

```
import unittest
from rk1 import task_1, task_2, task_3

task_1_result = {('Western Digital Blue WD10EZEX', 'Intel Core i3-6100'),
                  ('Western Digital Purple WD40PURX', 'Intel Core i5-7400'),
                  ('Western Digital Blue WD5000LPCX', 'Intel Core i5-7400')}
task_2_result = [('Intel Core i5-7400', 3072.0), ('AMD Ryzen 3 3100',
2048,0),('Intel Core i3-6100', 1024.0),
                  ('Intel Core i3-7300', 500.0), ('Intel Core i3-7100', 500.0),
                  ('Intel Core i5-6500', 500.0)
                  ]
task_3_result = [('Seagate Barracuda ST500LM030', 'AMD Ryzen 3 3100'), ('Seagate
SkyHawk ST2000VX008', 'Intel Core i5-6500')]
class TasksTestCase(unittest.TestCase):
    def task_task_1(self):
        self.assertEqual(task_1_result, set(task_1()))
    def task_task_2(self):
        self.assertEqual(task_2_result, task_2())
    def task_task_3(self):
        self.assertEqual(task_3_result, task_3())
if __name__ == '__main__':
    unittest.main()
```

Экранные формы с примерами выполнения программы

Testing started at 21:30 ...

Ran 3 tests in 0.003s

OK