

МГТУ им. Н. Э. Баумана

Отчет по рубежному контролю №2  
по курсу «Базовые компоненты и интернет-технологии»  
Вариант 7.

Руководитель:  
Гапанюк Ю. Е.  
17.12.2022

Выполнил:  
Студент группы ИУ5-34Б  
Козак А.А.  
17.12.2022

2022 г.

**Полученное задание:**

- 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).

**Запросы:**

1. «Компьютер» и «Микропроцессор» связаны соотношением один-ко-многим. Выведите список всех компьютеров, у которых название начинается с буквы «А», и список их микропроцессоров.
2. «Компьютер» и «Микропроцессор» связаны соотношением один-ко-многим. Выведите список компьютеров с максимальной стоимостью микропроцессора в каждом компьютере, отсортированный по максимальной стоимости.
3. «Компьютер» и «Микропроцессор» связаны соотношением многие-ко-многим. Выведите список всех связанных микропроцессоров и компьютеров, отсортированный по компьютерам, сортировка по микропроцессорам произвольная.

Текст программы

rk\_1.py

```
1  # -*- coding: utf-8 -*-
2  class Mic:
3      """Микропроцессор"""
4      def __init__(self, id, name, price, comp_id):
5          self.id = id
6          self.name = name
7          self.price = price
8          self.comp_id = comp_id
9
10 class Comp:
11     """Компьютер"""
12     def __init__(self, id, name):
13         self.id = id
14         self.name = name
15
16 class Mic_Comp:
17     """
18     'Микропроцессоры компьютера' для реализации
19     связи многие-ко-многим
20     """
21     def __init__(self, comp_id, mic_id):
22         self.comp_id = comp_id
23         self.mic_id = mic_id
24
25 #Компьютеры
26 comps = [
27     Comp(1, 'Acer Z5230'),
28     Comp(2, 'Lenovo Y570'),
29     Comp(3, 'Asus RoG1720'),
30     Comp(4, 'Apple MacBook13')
31 ]
32
```

```

34 mics = [
35     Mic(1, 'Intel i5 7880', 25000, 1),
36     Mic(2, 'Intel i7 3620', 32000, 2),
37     Mic(3, 'Amd Ryzen X5', 42000, 3),
38     Mic(4, 'Intel Ce11eron G3900', 15000, 3),
39     Mic(5, 'Amd FX5300', 20000, 3),
40     Mic(6, 'Apple M1', 27000, 4)
41 ]
42
43 mic_comps = [
44     Mic_Comp(1,1),
45     Mic_Comp(2,2),
46     Mic_Comp(3,3),
47     Mic_Comp(3,4),
48     Mic_Comp(3,5),
49     Mic_Comp(4,6)
50 ]
51
52 one_to_many = [(m.name, m.price, c.name)
53                 for m in mics
54                 for c in comps
55                 if m.comp_id == c.id]
56
57 # Соединение данных многие-ко-многим
58 many_to_many_temp = [(c.name, mc.comp_id, mc.mic_id)
59                       for c in comps
60                       for mc in mic_comps
61                       if c.id==mc.comp_id]
62
63 many_to_many = [(m.name, m.price, comp_name)
64                 for comp_name, comp_id, mic_id in many_to_many_temp
65                 for m in mics if m.id==mic_id]
66
67
68 def task_1(one_to_many):
69     return List(filter(lambda x: x[2].startswith("A"), one_to_many))
70

```

```

68 def task_1(one_to_many):
69     return List(filter(lambda x: x[2].startswith("A"), one_to_many))
70
71 def task_2(one_to_many):
72     res_g2 = []
73     for c in comps:
74         # Список микропроцессоров компьютера
75         c_mic = List(filter(lambda i: i[2] == c.name, one_to_many))
76         # Если у компьютера есть микропроцессоры
77         if len(c_mic) > 0:
78             # Максимальная зарплата
79             max_price = max([price for _, price, _ in c_mic])
80             res_g2.append((c.name, max_price))
81     return sorted(res_g2, key = lambda x: x[1])
82
83 def task_3(many_to_many):
84     return sorted(many_to_many, key = lambda x: x[2])
85
86 if __name__ == "__main__":
87     print("Задание 1")
88     print(task_1(one_to_many))
89     print("Задание 2")
90     print(task_2(one_to_many))
91     print("Задание 3")
92     print(task_3(many_to_many))

```

## rk\_1\_test\_tdd.py

```

> rk_1.py × rk_1_test_tdd.py ×
from rk_1 import *
import unittest

class Test_RK(unittest.TestCase):
    def test_task_1(self):
        ans_1 = [('Intel i5 7880', 25000, 'Acer Z5230'), ('Amd Ryzen X5', 42000, 'Asus RoG1720'),
                  ('Intel Celeron G3900', 15000, 'Asus RoG1720'), ('Amd FX5300', 20000, 'Asus RoG1720'),
                  ('Apple M1', 27000, 'Apple MacBook13')]
        self.assertEqual(task_1(one_to_many), ans_1)
    def test_task_2(self):
        ans_2 = [('Acer Z5230', 25000), ('Apple MacBook13', 27000), ('Lenovo Y570', 32000), ('Asus RoG1720', 42000)]
        self.assertEqual(task_2(one_to_many), ans_2)
    def test_task_3(self):
        ans_3 = [('Intel i5 7880', 25000, 'Acer Z5230'), ('Apple M1', 27000, 'Apple MacBook13'),
                  ('Amd Ryzen X5', 42000, 'Asus RoG1720'), ('Intel Celeron G3900', 15000, 'Asus RoG1720'),
                  ('Amd FX5300', 20000, 'Asus RoG1720'), ('Intel i7 3620', 32000, 'Lenovo Y570')]
        self.assertEqual(task_3(many_to_many), ans_3)

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

```

## Результат выполнения

```
===== test session starts =====
platform win32 -- Python 3.8.5, pytest-7.2.0, pluggy-1.0.0
rootdir: C:\Users\Vegard\Pictures\Lab 3
plugins: bdd-6.0.1
collected 3 items

rk_1_test_tdd.py ... [100%]

===== 3 passed in 0.09s =====
```