

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по Рубежному контролю №2
«Основные конструкции языка Python»

Выполнил:
студент группы ИУ5-32Б:
Долинский Александр
Александрович
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:

Москва, 2021 г.

Постановка задачи:

Вариант №4

Вариант Б.

1. «Дисплейный класс» и «Компьютер» связаны соотношением один-ко-многим. Выведите список всех связанных компьютеров и дисплейных классов, отсортированный по компьютерам, сортировка по дисплейным классам произвольная.
2. «Дисплейный класс» и «Компьютер» связаны соотношением один-ко-многим. Выведите список дисплейных классов с количеством компьютеров в каждом, отсортированный по количеству компьютеров.
3. «Дисплейный класс» и «Компьютер» связаны соотношением многие-ко-многим. Выведите список всех компьютеров, у которых название бренда заканчивается на «о», и названия их дисплейных классов.

Условия рубежного контроля №2 по курсу БКИТ

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).

Текст кода:

main.py

```
from operator import itemgetter

class Comp:
    """Компьютер"""

    def __init__(self, id, brand, price, disp_cls_id):
        self.id = id
        self.brand = brand
        self.price = price
        self.disp_cls_id = disp_cls_id

class Disp_cls:
    """Дисплейный класс"""

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

class CompDisp_cls:
    """
    'Компьютеры дисплейного класса' для реализации
    связи многие-ко-многим
    """
```

```

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

# Дисплейные классы
disp_classes = [
    Disp_cls(1, 'А-класс'),
    Disp_cls(2, 'Б-класс'),
    Disp_cls(3, 'В-класс'),
    Disp_cls(11, 'Г-класс'),
    Disp_cls(22, 'Д-класс'),
    Disp_cls(33, 'Е-класс'),
]

# Компьютеры
comps = [
    Comp(1, 'HP', 1168390, 1),
    Comp(2, 'Asus', 55390, 1),
    Comp(3, 'Vaio', 78940, 3),
    Comp(4, 'Acer', 67450, 2),
    Comp(5, 'Lenovo', 86980, 3),
]

# Компьютеры и Дисплейные классы для связи многие-ко-многим
comps_disp_classss = [
    CompDisp_cls(1, 1),
    CompDisp_cls(1, 2),
    CompDisp_cls(3, 3),
    CompDisp_cls(2, 4),
    CompDisp_cls(3, 5),

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

def first(one_to_many):
    task_1 = sorted(one_to_many, key=itemgetter(0))
    return task_1

def second(one_to_many):
    task2_unsorted = []
    for c in disp_classes:
        d_disp_cls = list(filter(lambda i: i[2] == c.name, one_to_many))
        if len(d_disp_cls) > 0:
            task2_unsorted.append((c.name, len(d_disp_cls)))

    task_2 = sorted(task2_unsorted, key=itemgetter(1), reverse=True)
    return task_2

def third(many_to_many):
    task_3 = {}
    # Перебираем все компьютеры
    for s in comps:
        if s.brand.endswith("o"):
            c_comps = list(filter(lambda i: i[0] == s.brand, many_to_many))
            c_comps_brands = [x[2] for x in c_comps]

```

```

        task_3[s.brand] = c_comps_brands
    return(task_3)

def main():
    """Основная функция"""
    # Соединение данных один-ко-многим
    one_to_many = [(c.brand, c.price, d.name)
                    for d in disp_classes
                    for c in comps
                    if c.disp_cls_id == d.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(d.name, dc.disp_cls_id, dc.comp_id)
                           for d in disp_classes
                           for dc in comps_disp_classss
                           if d.id == dc.disp_cls_id]

    many_to_many = [(c.brand, c.price, disp_class_name)
                     for disp_class_name, disp_cls_id, comp_id in
many_to_many_temp
                     for c in comps if c.id == comp_id]

    print('\nЗадание 1 \n', first(one_to_many))
    print('\nЗадание 2 \n', second(one_to_many))
    print('\nЗадание 3 \n', third(many_to_many))

if __name__ == '__main__':
    main()

```

Test_TDD.py

```

import unittest
import sys, os

sys.path.append(os.getcwd())
from main import *
class Test(unittest.TestCase):
    def test_1(self):
        one_to_many = [(c.brand, c.price, d.name)
                        for d in disp_classes
                        for c in comps
                        if c.disp_cls_id == d.id]
        self.assertEqual(first(one_to_many), [('Acer', 67450, 'Б-класс'),
        ('Asus', 55390, 'А-класс'),
        ('HP', 1168390, 'А-класс'),
        ('Lenovo', 86980, 'В-класс'),
        ('Vaio', 78940, 'В-класс')])
    def test_2(self):
        one_to_many = [(c.brand, c.price, d.name)
                        for d in disp_classes
                        for c in comps
                        if c.disp_cls_id == d.id]
        self.assertEqual(second(one_to_many), [('А-класс', 2), ('В-класс',
2), ('В-класс', 1)])
    def test_3(self):
        many_to_many_temp = [(d.name, dc.disp_cls_id, dc.comp_id)
                              for d in disp_classes
                              for dc in comps_disp_classss
                              if d.id == dc.disp_cls_id]
        many_to_many = [(c.brand, c.price, disp_class_name)
                          for disp_class_name, disp_cls_id, comp_id in many_to_many_temp

```

```
        for c in comps if c.id == comp_id]
    self.assertEqual(third(many_to_many),
                     {'Vaio': ['В-класс', 'Г-класс'], 'Lenovo': ['В-класс', 'Д-класс']})
```

Тестирование: main.py

```
C:\Users\LENOV0\PycharmProjects\RK2\venv\Scripts\python.exe C:/Users/LENOV0/PycharmProjects/RK2/main.py
```

```
Задание 1
[('Acer', 67450, 'Б-класс'), ('Asus', 55390, 'А-класс'), ('HP', 1168390, 'А-класс'), ('Lenovo', 86980, 'В-класс'), ('Vaio', 78940, 'В-класс')]
```

```
Задание 2
[('А-класс', 2), ('В-класс', 2), ('Б-класс', 1)]
```

```
Задание 3
{'Vaio': ['В-класс', 'Г-класс'], 'Lenovo': ['В-класс', 'Д-класс']}
```

```
Process finished with exit code 0
```

Test_TDD.py

```
Testing started at 14:47 ...
```

```
C:\Users\LENOV0\PycharmProjects\RK2\venv\Scripts\python.exe "C:\Program Files\JetBrains\PyCharm Community Edition 2020.1
.plugins\python-ce\helpers\pycharm\_jb_unittest_runner.py" --path C:/Users/LENOV0/PycharmProjects/RK2/Test_TDD.py
```

```
Launching unittests with arguments python -m unittest C:/Users/LENOV0/PycharmProjects/RK2/Test_TDD.py in C:\Users\LENOV0\PycharmProjects\RK2
```

```
Ran 3 tests in 0.007s
```

```
OK
```

```
Process finished with exit code 0
```