

**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  
Вариант 10**

Выполнил:  
студент группы ИУ5-34Б:  
Малютин Илья  
Подпись и дата:

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

Москва, 2022 г.

## Условия РК №1

### Вариант Г. Предметная область №10

1. «Компьютер» и «Браузер» связаны соотношением один-ко-многим. Выведите список всех компьютеров, у которых в названии есть «1», и список браузеров в них.

2. «Компьютер» и «Браузер» связаны соотношением один-ко-многим. Выведите список компьютеров с максимальной скоростью браузера в каждом компьютере, отсортированный по максимальной скорости.

3. «Компьютер» и «Браузер» связаны соотношением многие-ко-многим. Выведите список всех связанных компьютеров и браузеров, отсортированный по компьютерам, сортировка браузеров произвольная.

## Условия РК №2

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).

### Код программы к РК №1

```
from operator import itemgetter
class browser:
    """Browser"""
    def __init__(self, id, name, sp, pc_id):
        self.id = id
        self.name_b = name
        self.max_sp = sp
        self.pc_id = pc_id

class pc:
    """Computer"""
    def __init__(self, id, name):
        self.id = id
        self.name_pc = name

class bop:
    """
    'Браузеры компьютера для реализации
    связи многие-ко-многим
    """
    def __init__(self, pc_id, b_id):
        self.b_id = b_id
        self.pc_id = pc_id

# Computers
pcs = [
    pc(1, 'pc_01'),
    pc(2, 'pc_02'),
    pc(3, 'pc_03'),

    pc(11, 'pc_11'),
    pc(22, 'pc_22'),
    pc(33, 'pc_33')
]
```

```

# Browsers
browsers = [
    browser(1, 'Opera', 1000, 1),
    browser(2, 'Chrome', 900, 2),
    browser(3, 'Edge', 800, 3),
    browser(4, 'FireFox', 950, 3),
    browser(5, 'Yandex', 900, 3),
    browser(6, 'Explorer', 100, 11)
]

BOP = [
    bop(1, 1),
    bop(2, 2),
    bop(3, 3),
    bop(3, 4),
    bop(3, 5),
    bop(3, 6),

    bop(11, 1),
    bop(22, 2),
    bop(33, 3),
    bop(33, 4),
    bop(33, 5),
    bop(33, 6)
]

def main():
    """Основная функция"""

    # Соединение данных один-ко-многим
    one_to_many = [(b.name_b, b.max_sp, p.name_pc)
                    for p in pcs
                    for b in browsers
                    if b.pc_id == p.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(p.name_pc, bp.pc_id, bp.b_id)
                           for p in pcs
                           for bp in BOP
                           if p.id == bp.pc_id]

    many_to_many = [(b.name_b, b.max_sp, pc_name)
                     for pc_name, pc_id, b_id in many_to_many_temp
                     for b in browsers if b.id == b_id]

    print('Задание Г1')
    res1 = []
    for el in one_to_many:
        if el[2].find('1') != -1:
            res1.append({el[2] : el[0]})
    print(res1)

    print('\nЗадание Г2')
    otm_sorted = sorted(one_to_many, key = itemgetter(1), reverse = 1)
    res2_tmp = [el[2] for el in otm_sorted]
    res2_set = []

    #Реализация списка с уникальными компьютерами отсортированными по скорости
    браузера
    for el in res2_tmp:
        if len(res2_set + [el]) == len(list(set(res2_set + [el]))):
            res2_set.append(el)

```

```

res2 = []
for el in res2_set:
    res2.append(otm_sorted[res2_tmp.index(el)])
print(res2)

print('\nЗадание Г3')
res3 = sorted(many_to_many, key = itemgetter(2))
for el in res3:
    print(el)
if __name__ == '__main__':
    main()

```

## Рефакторинг кода к РК №1

```

from operator import itemgetter
class browser:
    """Browser"""
    def __init__(self, id, name, sp, pc_id):
        self.id = id
        self.name_b = name
        self.max_sp = sp
        self.pc_id = pc_id

class pc:
    """Computer"""
    def __init__(self, id, name):
        self.id = id
        self.name_pc = name

class bop:
    """
    Браузеры компьютера для реализации
    связи многие-ко-многим
    """
    def __init__(self, pc_id, b_id):
        self.b_id = b_id
        self.pc_id = pc_id

# Computers
pcs = [
    pc(1, 'pc_01'),
    pc(2, 'pc_02'),
    pc(3, 'pc_03'),

    pc(11, 'pc_11'),
    pc(22, 'pc_22'),
    pc(33, 'pc_33')
]

# Browsers
browsers = [
    browser(1, 'Opera', 1000, 1),
    browser(2, 'Chrome', 900, 2),
    browser(3, 'Edge', 800, 3),
    browser(4, 'FireFox', 950, 3),
    browser(5, 'Yandex', 900, 3),
    browser(6, 'Explorer', 100, 11)
]

BOP = [
    bop(1, 1),
    bop(2, 2),
    bop(3, 3),
    bop(3, 4),
    bop(3, 5),

```

```

        bop(3, 6),

        bop(11, 1),
        bop(22, 2),
        bop(33, 3),
        bop(33, 4),
        bop(33, 5),
        bop(33, 6)
    ]

# Соединение данных один-ко-многим
one_to_many = [(b.name_b, b.max_sp, p.name_pc)
                for p in pcs
                for b in browsers
                if b.pc_id == p.id]

# Соединение данных многие-ко-многим
many_to_many_temp = [(p.name_pc, bp.pc_id, bp.b_id)
                      for p in pcs
                      for bp in BOP
                      if p.id == bp.pc_id]

many_to_many = [(b.name_b, b.max_sp, pc_name)
                 for pc_name, pc_id, b_id in many_to_many_temp
                 for b in browsers if b.id == b_id]

def task_1(one_to_many):
    res1 = []
    for el in one_to_many:
        if el[2].find('1') != -1:
            res1.append({el[2] : el[0]})
    return res1

def task_2(one_to_many):
    otm_sorted = sorted(one_to_many, key = itemgetter(1), reverse = 1)
    res2_tmp = [el[2] for el in otm_sorted]
    res2_set = []

    #Реализация списка с уникальными компьютерами отсортированными по скорости
    #браузера
    for el in res2_tmp:
        if len(res2_set + [el]) == len(list(set(res2_set + [el]))):
            res2_set.append(el)

    res2 = []
    for el in res2_set:
        res2.append(otm_sorted[res2_tmp.index(el)])
    return res2

def task_3(many_to_many):
    res3 = sorted(many_to_many, key = itemgetter(2))
    return res3

def main():
    print('\nЗадание Г1')
    print(task_1(one_to_many))
    print('\nЗадание Г2')
    print(task_2(one_to_many))
    print('\nЗадание Г3')
    print(task_3(many_to_many))

```

```
if __name__ == '__main__':  
    main()
```

## Код программы к РК №2 testing.py

```
import os  
import sys  
import unittest  
  
sys.path.append(os.getcwd()) # current working directory  
from RK1_BKIT import task_1, task_2, task_3  
from RK1_BKIT import one_to_many, many_to_many  
  
class test_task_1(unittest.TestCase):  
    def testtask_1(self):  
        self.assertEqual(task_1(one_to_many), [{'pc_01': 'Opera'}, {'pc_11':  
'Explorer'}])  
  
class test_task_2(unittest.TestCase):  
    def testtask_2(self):  
        self.assertEqual(task_2(one_to_many), [('Opera', 1000, 'pc_01'),  
        ('Firefox', 950, 'pc_03'),  
        ('Chrome', 900, 'pc_02'),  
        ('Explorer', 100, 'pc_11')])  
  
class test_task_3(unittest.TestCase):  
    def testtask_3(self):  
        self.assertEqual(task_3(many_to_many), [('Opera', 1000, 'pc_01'),  
        ('Chrome', 900, 'pc_02'),  
        ('Edge', 800, 'pc_03'),  
        ('Firefox', 950, 'pc_03'),  
        ('Yandex', 900, 'pc_03'),  
        ('Explorer', 100, 'pc_03'),  
        ('Opera', 1000, 'pc_11'),  
        ('Chrome', 900, 'pc_22'),  
        ('Edge', 800, 'pc_33'),  
        ('Firefox', 950, 'pc_33'),  
        ('Yandex', 900, 'pc_33'),  
        ('Explorer', 100, 'pc_33')])  
  
if __name__ == '__main__':  
    unittest.main()
```

## Результат

```
...  
-----  
Ran 3 tests in 0.000s  
|  
OK  
  
Process finished with exit code 0
```

