**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: проверил:

студент группы ИУ5-34Б: преподаватель каф. ИУ5

Малютин Илья Гапанюк Ю.Е

Подпись и дата: 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**

