**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 №1  
«Основные конструкции языка Python»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-32Б: |  | преподаватель каф. ИУ5 |
| Долинский Александр Александрович |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | 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 каждо, 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.

**Текст кода:**

# используется для сортировки  
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', 97450, 2),  
 Comp(5, 'Lenovo', 86980, 3),  
]  
  
# Компьютеры и Дисплейные классы для связи многие-ко-многим  
comps\_disp\_classs = [  
 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 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\_classs  
 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('Задание Б1')  
 res\_1 = sorted(one\_to\_many, key=itemgetter(0))  
 print(res\_1)  
  
 print('\nЗадание Б2')  
 res\_2\_unsorted = []  
 for c in disp\_classes:  
 # Все компьютеры дисплейного класса  
 c\_comps = list(filter(lambda i: i[2] == c.name, one\_to\_many))  
 if len(c\_comps) > 0:  
 res\_2\_unsorted.append((c.name, len(c\_comps)))  
  
 res\_2 = sorted(res\_2\_unsorted, key=itemgetter(1), reverse=True)  
 print(res\_2)  
  
 print('\nЗадание Б3')  
 res\_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]  
 res\_3[s.brand] = c\_comps\_brands  
  
 print(res\_3)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

**Тестирование:**

