ИУ5-34Б

Матвеев Илья

**Отчет РК1 по дисциплине**

**«Парадигмы и конструкции языков программирования»**

**Запрос 1**  
Список всех операционных систем,   
у которых название начинается с буквы «W»,   
и список установленных на них компьютеров   
(связь "один ко многим")

**Запрос 2**  
Список операционных систем с максимальной ценой компьютеров,   
отсортированный по цене (связь "один ко многим")

**Запрос 3**  
Список всех связанных компьютеров и операционных систем,   
отсортированный по операционным системам (связь "многие ко многим")

**Текст программы:**

from operator import itemgetter  
  
class Computer:  
 *"""Компьютер"""* def \_\_init\_\_(self, id, name, price, os\_id):  
 self.id = id  
 self.name = name  
 self.price = price  
 self.os\_id = os\_id  
  
  
class OperatingSystem:  
 *"""Операционная система"""* def \_\_init\_\_(self, id, name):  
 self.id = id  
 self.name = name  
  
  
class ComputersOS:  
 *"""  
 'Компьютеры с ОС' для реализации  
 связи многие-ко-многим  
 """* def \_\_init\_\_(self, os\_id, comp\_id):  
 self.os\_id = os\_id  
 self.comp\_id = comp\_id  
  
  
# Операционные системы  
oses = [  
 OperatingSystem(1, 'Windows'),  
 OperatingSystem(2, 'Linux'),  
 OperatingSystem(3, 'macOS'),  
 OperatingSystem(11, 'Windows Server'),  
 OperatingSystem(22, 'Ubuntu'),  
]  
  
# Компьютеры  
computers = [  
 Computer(1, 'PC1', 1000, 1),  
 Computer(2, 'PC2', 1200, 1),  
 Computer(3, 'PC3', 900, 11),  
 Computer(4, 'PC4', 1500, 3),  
 Computer(5, 'PC5', 1100, 2),  
]  
  
# Связь многие-ко-многим (Компьютеры и ОС)  
computers\_os = [  
 ComputersOS(1, 1),  
 ComputersOS(1, 2),  
 ComputersOS(2, 3),  
 ComputersOS(2, 5),  
 ComputersOS(3, 4),  
 ComputersOS(11, 1),  
 ComputersOS(22, 3),  
]  
  
  
def main():  
 *"""Основная функция"""* # Соединение данных один-ко-многим  
 one\_to\_many = [(c.name, c.price, os.name)  
 for os in oses  
 for c in computers  
 if c.os\_id == os.id]  
  
 # Соединение данных многие-ко-многим  
 many\_to\_many\_temp = [(os.name, co.os\_id, co.comp\_id)  
 for os in oses  
 for co in computers\_os  
 if os.id == co.os\_id]  
  
 many\_to\_many = [(c.name, c.price, os\_name)  
 for os\_name, os\_id, comp\_id in many\_to\_many\_temp  
 for c in computers if c.id == comp\_id]  
  
  
 """  
 Задание 1  
 Список всех операционных систем,   
 у которых название начинается с буквы «W»,   
 и список установленных на них компьютеров   
 (связь "один ко многим")  
 """  
 print("Задание Г1")  
 # Выбираем ОС, название которых начинается с буквы "W"  
 selected\_os = [os for os in oses if os.name.startswith('W')]  
  
 # Для каждой ОС находим компьютеры  
 result\_1 = {}  
 for os in selected\_os:  
 result\_1[os.name] = [comp.name for comp in computers if comp.os\_id == os.id]  
  
 print(result\_1)  
  
 """  
 Задание 2  
 Список операционных систем с максимальной ценой компьютеров,   
 отсортированный по цене (связь "один ко многим")  
 """  
 print("\nЗадание Г2")  
 res\_2\_unsorted = []  
 # Перебираем все ОС  
 for os in oses:  
 # Список компьютеров с данной ОС  
 os\_comps = list(filter(lambda i: i[2] == os.name, one\_to\_many))  
 # Если есть компьютеры с этой ОС  
 if len(os\_comps) > 0:  
 # Цены компьютеров с этой ОС  
 os\_prices = [price for \_, price, \_ in os\_comps]  
 # Максимальная цена  
 os\_max\_price = max(os\_prices)  
 res\_2\_unsorted.append((os.name, os\_max\_price))  
  
 # Сортировка по максимальной цене  
 res\_2 = sorted(res\_2\_unsorted, key=itemgetter(1), reverse=True)  
 print(res\_2)  
  
 """  
 Задание 3  
 Список всех связанных компьютеров и операционных систем,   
 отсортированный по операционным системам (связь "многие ко многим")  
 """  
 # Создаем словарь для связи ОС с компьютерами  
 print("\nЗадание Г3")  
 result\_3 = {}  
 for os in oses:  
 # Находим компьютеры, связанные с этой ОС  
 result\_3[os.name] = [comp.name for relation in computers\_os for comp in computers if  
 relation.os\_id == os.id and relation.comp\_id == comp.id]  
  
 # Сортировка по имени ОС  
 sorted\_result\_3 = {os: result\_3[os] for os in sorted(result\_3.keys())}  
  
 print(sorted\_result\_3)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

**Результаты выполнения программы:**

**Задание Г1**

**{'Windows': ['PC1', 'PC2'], 'Windows Server': ['PC3']}**

**Задание Г2**

**[('macOS', 1500), ('Windows', 1200), ('Linux', 1100), ('Windows Server', 900)]**

**Задание Г3**

**{'Linux': ['PC3', 'PC5'], 'Ubuntu': ['PC3'], 'Windows': ['PC1', 'PC2'], 'Windows Server': ['PC1'], 'macOS': ['PC4']}**

**Process finished with exit code 0**