Московский государственный технический университет им. Н.Э. Баумана

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

Отчет по рубежному контролю №12
по курсу «Базовые компоненты интернет-технологии»
Вариант №20

Студент:

Теряева Ксения Владимировна, ИУ5-34Б

Руководитель:

Гапанюк Ю.Е.

Дата 16.10.2022

Москва, 2022 г

Задание к РК1:
Необходимо создать два класса данных в соответствии с Вашим вариантом предметной области, которые связаны отношениями один-ко-многим и многие-ко-многим.
Необходимо создать списки объектов классов, содержащих тестовые данные (3-5 записей), таким образом, чтобы первичные и вторичные ключи соответствующих записей были связаны по идентификаторам.

Необходимо разработать запросы в соответствии с Вашим вариантом.

Вариант Г.

1. «Поставщик» и «Деталь» связаны соотношением один-ко-многим. Выведите список всех поставщиков, у которых фамилия начинается с буквы «А», и список деталей.
2. «Поставщик» и «Деталь» связаны соотношением один-ко-многим. Выведите список поставщиков с максимальной ценой детали, отсортированный по максимальной цене.
3. «Поставщик» и «Деталь» связаны соотношением многие-ко-многим. Выведите список всех связанных поставщиков и деталей, отсортированный по поставщикам, сортировка по деталям произвольная.

Задание к РК2:
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 программы и результаты выполнения:

```
class Detail:
    """Деталь"""
    def __init__(self, id, name, price, det_id):
        self.id = id
        self.name = name
        self.price = price
        self.det_id = det_id


class Provider:
    """Поставщик"""
    def __init__(self, id, name):
        self.id = id
        self.name = name


class DetPro:
    """
    'Детали для поставщика' для реализации
    связи многие-ко-многим
    """
    def __init__(self, prov_id, det_id):
        self.det_id = det_id
        self.prov_id = prov_id

# Поставщики
providers = [
    Provider(1, 'Williams'),
    Provider(2, 'Gibson'),
    Provider(3, 'Jordan'),
    Provider(4, 'Anderson'),
    Provider(5, 'Styles'),
]
```

```python
# Детали
details = [
    Detail(1, 'Video card', 25000, 1),
    Detail(2, 'CPU', 35000, 2),
    Detail(3, 'Monitor', 45000, 3),
    Detail(4, 'HDD', 35000, 4),
    Detail(5, 'Video card', 50000, 5),
    Detail(6, 'CPU', 60000, 5),
    Detail(7, 'CPU', 85000, 1),
]

det_pros = [
    DetPro(1,1),
    DetPro(2,2),
    DetPro(3,3),
    DetPro(4,4),
    DetPro(5,5),
    DetPro(5,6),
    DetPro(2,7),
    #DetPro(3,5),

]

# Соединение данных один-ко-многим
one_to_many = [(e.name, e.price, d.name)
    for d in providers
    for e in details
    if e.det_id==d.id]

# Соединение данных многие-ко-многим
many_to_many_temp = [(d.name, ed.prov_id, ed.det_id)
    for d in providers
    for ed in det_pros
    if d.id == ed.prov_id]

many_to_many = [(e.name, e.price, prov_name)
    for prov_name, prov_id, det_id in many_to_many_temp
    for e in details if e.id==det_id]

one_to_many

many_to_many

def task_1(one_to_many):
    for i in one_to_many:
      if (i[2][0] == "A"):
        return i

def task_2(one_to_many):
    mas = []
    for i in providers:
      max_1 = 0
      for k in one_to_many:
        if (i.name == k[2]):
          if (k[1]>max_1):
            max_1 = k[1]
      mas.append((i.name, max_1))

    return sorted(mas,key = lambda x: x[1])
def task_3(many_to_many):
    return sorted(many_to_many,key = lambda x: x[2])
```

```
============================= test session starts =============================
platform win32 -- Python 3.10.7, pytest-7.2.0, pluggy-1.0.0
rootdir: C:\Users\Ксения\Desktop\Студентка получается\3ий сем\proger\RK1
plugins: anyio-3.6.2, bdd-6.1.1
collected 3 items

rk_1_test_tdd.py ...                                                     [100%]

============================== 3 passed in 0.02s ==============================
```