Московский государственный технический университет имени Н. Э. Баумана

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Факультет Радиотехнический

Кафедра ИУ5 Системы обработки информации и управления

**Отчет по рубежному контролю №1 по курсу**

**Базовые компоненты интернет-технологий**

4

(количество листов)

Вариант № **16**

|  |  |  |
| --- | --- | --- |
| Исполнитель |  |  |
| студент группы РТ5-31б | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Нижаметдинов М.Ш. |
|  |  | “ 23 ” октября 2021 г. |
|  |  |  |
| Проверил |  |  |
| Доцент кафедры ИУ5 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Гапанюк Ю.Е. |
|  |  | “\_\_\_” \_\_\_\_\_\_\_\_\_\_ 2021 г. |

Москва, 2021 г.

### **Задание**

Рубежный контроль представляет собой разработку программы на языке Python, которая выполняет следующие действия:

1) Необходимо создать два класса данных в соответствии с Вашим вариантом предметной области, которые связаны отношениями один-ко-многим и многие-ко-многим.

Пример классов данных для предметной области Сотрудник-Отдел:

1. Класс «Сотрудник», содержащий поля:
   * ID записи о сотруднике;
   * Фамилия сотрудника;
   * Зарплата (количественный признак);
   * ID записи об отделе. (для реализации связи один-ко-многим)
2. Класс «Отдел», содержащий поля:
   * ID записи об отделе;
   * Наименование отдела.
3. (Для реализации связи многие-ко-многим) Класс «Сотрудники отдела», содержащий поля:
   * ID записи о сотруднике;
   * ID записи об отделе.

2) Необходимо создать списки объектов классов, содержащих тестовые данные (3-5 записей), таким образом, чтобы первичные и вторичные ключи соответствующих записей были связаны по идентификаторам.

3) Необходимо разработать запросы в соответствии с Вашим вариантом. Запросы сформулированы в терминах классов «Сотрудник» и «Отдел», которые используются в примере. Вам нужно перенести эти требования в Ваш вариант предметной области. При разработке запросов необходимо по возможности использовать функциональные возможности языка Python (list/dict comprehensions, функции высших порядков).

Для реализации запроса №2 введите в класс, находящийся на стороне связи «много», произвольный количественный признак, например, «зарплата сотрудника».

Результатом рубежного контроля является документ в формате PDF, который содержит текст программы и результаты ее выполнения.

### **Текст программы**

*# используется для сортировки*

*from operator import itemgetter*

*class Book:*

*"""Книга"""*

*def \_\_init\_\_(self, id, name, author, price, shop\_id):*

*self.id = id*

*self.name = name*

*self.author = author*

*self.price = price*

*self.shop\_id = shop\_id*

*class Shop:*

*"""Книжный магазин"""*

*def \_\_init\_\_(self, id, name):*

*self.id = id*

*self.name = name*

*class BookShop:*

*"""*

*'Книги магазина' для реализации*

*связи многие-ко-многим*

*"""*

*def \_\_init\_\_(self, shop\_id, book\_id):*

*self.shop\_id = shop\_id*

*self.book\_id = book\_id*

*# Книжные магазины*

*shops = [*

*Shop(1, 'Книжный бульвар'),*

*Shop(2, 'Книга инженера'),*

*Shop(3, 'Книжная полка'),*

*Shop(4, 'Зелёная книга'),*

*Shop(11, 'Новый книжный'),*

*Shop(22, 'Книгоград'),*

*Shop(33, 'Книга – друг'),*

*]*

*# Книги*

*books = [*

*Book(1, 'Властелин колец', 'Джон Р. Р. Толкин', 300, 1),*

*Book(2, 'Адвокат Дьявола', 'Эндрю Найдерман', 350, 3),*

*Book(3, 'Визуализация в научных исследованиях. Учебное пособие', 'Корнеев В.И.', 900, 2),*

*Book(4, '1984', 'Джордж Оруэлл', 400, 4),*

*Book(5, 'Тёмные начала', 'Филип Пулман', 425, 3),*

*Book(6, 'Гарри Поттер и Кубок огня', 'Джоан Роулинг', 340, 1),*

*Book(7, 'Убить пересмешника', 'Харпер Ли', 530, 3),*

*Book(8, 'Алая буква', 'Натаниэль Готорн', 229, 4),*

*]*

*books\_shops = [*

*BookShop(1, 1),*

*BookShop(3, 2),*

*BookShop(2, 3),*

*BookShop(4, 4),*

*BookShop(3, 5),*

*BookShop(1, 6),*

*BookShop(3, 7),*

*BookShop(4, 8),*

*BookShop(22, 1),*

*BookShop(11, 2),*

*BookShop(33, 3),*

*BookShop(11, 4),*

*BookShop(33, 5),*

*BookShop(22, 6),*

*BookShop(33, 7),*

*]*

*def main():*

*"""Основная функция"""*

*# Соединение данных один-ко-многим*

*one\_to\_many = [(b.name, b.author, b.price, s.name)*

*for s in shops*

*for b in books*

*if b.shop\_id == s.id]*

*# Соединение данных многие-ко-многим*

*many\_to\_many\_temp = [(s.name, bs.shop\_id, bs.book\_id)*

*for s in shops*

*for bs in books\_shops*

*if s.id == bs.shop\_id]*

*many\_to\_many = [(b.name, b.author, b.price, shop\_name)*

*for shop\_name, shop\_id, book\_id in many\_to\_many\_temp*

*for b in books*

*if b.id == book\_id]*

*print('Задание E1')*

*res = list()*

*# Перебираем все книги во всех магазинах*

*for b\_name, b\_author, \_, s\_name in one\_to\_many:*

*# Если в названии магазина содержится слово "книга",*

*# добавить название этого магазина и все книги*

*# этого магазина в результирующий список*

*if 'книга' in s\_name.lower():*

*res.append((s\_name, b\_name, b\_author))*

*print(res)*

*print('\nЗадание E2')*

*res = []*

*# Перебираем все магазины*

*for s in shops:*

*# Получаем книги текущего магазина*

*s\_books = list(filter(lambda i: i[3] == s.name, one\_to\_many))*

*if len(s\_books) > 0:*

*# Получаем список цен текущего магазина*

*s\_prices = [price for \_, \_, price, \_ in s\_books]*

*# Добавляем в результирующий список имя магазина*

*# и среднюю цену книги*

*res.append((s.name, round(sum(s\_prices) / len(s\_prices), 2)))*

*res = sorted(res, key=itemgetter(1))*

*print(res)*

*print('\nЗадание E3')*

*res = []*

*# Перебираем все книги во всех магазинах*

*for book\_name, book\_author, \_, shop\_name in many\_to\_many:*

*# Если название книги начинается с "А",*

*# в результирующий список добавляется*

*# название и автор книги и все магазины,*

*# в которых она продаётся*

*if book\_name[0] == 'А':*

*res.append((book\_name, book\_author, shop\_name))*

*res = sorted(res, key=itemgetter(0))*

*print(res)*

*if \_\_name\_\_ == '\_\_main\_\_':*

*main()*

### **Экранная форма с примерами выполнения программы**

