|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |
| --- | --- |
| ФАКУЛЬТЕТ «ИНЖЕНЕРНЫЙ БИЗНЕС И МЕНЕДЖМЕНТ»  КАФЕДРА «ПРОМЫШЛЕННАЯ ЛОГИСТИКА» (ИБМ-3)  **Рубежный контроль номер 1**  «Парадигмы и конструкции языков программирования»  38.03.05. Бизнес-информатика, Маркетинг цифровых технологий (уровень бакалавриата)  Студент ИБМ3- 34Б  \_\_\_\_\_\_\_\_\_\_ Разлада К.В  (Подпись, дата)  Руководитель Ю.У  (Подпись, дата)  2024 г.  1. Определение классов  • Класс "Detail":  \* id\_detail: Первичный ключ (уникальный идентификатор для каждой детали).  \* name: Название детали.  \* description: Описание детали.  \* id\_supplier: Вторичный ключ (указывает, какой поставщик поставляет эту деталь). Это поле устанавливает связь один-ко-многим: один поставщик может поставлять много деталей, но каждая деталь поставляется только одним поставщиком.  • Класс "Supplier":  \* id\_supplier: Первичный ключ (уникальный идентификатор для каждого поставщика).  \* name: Название поставщика.  \* address: Адрес поставщика.  • Класс "DetailsSupplier":  \* id\_detail: Вторичный ключ (указывает на идентификатор детали в классе "Detail").  \* id\_supplier: Вторичный ключ (указывает на идентификатор поставщика в классе "Supplier").  \* price: Цена детали, поставляемой конкретным поставщиком.    Этот класс реализует связь многие-ко-многим: одна деталь может поставляться несколькими поставщиками, а один поставщик может поставлять много разных деталей.  2. Списки объектов классов  • details: Список объектов класса "Detail" с тестовыми данными.  • suppliers: Список объектов класса "Supplier" с тестовыми данными.  • details\_supplier: Список объектов класса "DetailsSupplier" с тестовыми данными, связывающий детали и поставщиков.  3. Примеры связи  • Один-ко-многим: Связь между классом "Detail" и классом "Supplier" через поле id\_supplier. Одна деталь может быть поставлена только одним поставщиком, но один поставщик может поставлять несколько деталей.  • Многие-ко-многим: Связь реализована через класс "DetailsSupplier". Одна деталь может быть поставлена несколькими поставщиками, а один поставщик может поставлять несколько деталей.  4. Запросы  В коде представлены 3 запроса, которые иллюстрируют использование данных:  • Запрос 1: Найти все детали, которые поставляются поставщиком с ID 1.  • Запрос 2: Найти все детали, которые поставляются поставщиком "Поставщик 2" и вывести их названия вместе с ценой.  • Запрос 3: Найти всех поставщиков, которые поставляют деталь "Шайба".  Реализация кода:  class Detail:  def \_\_init\_\_(self, id\_detail, name, description, id\_supplier):  self.id\_detail = id\_detail  self.name = name  self.description = description  self.id\_supplier = id\_supplier  class Supplier:  def \_\_init\_\_(self, id\_supplier, name, address):  self.id\_supplier = id\_supplier  self.name = name  self.address = address  class DetailsSupplier:  def \_\_init\_\_(self, id\_detail, id\_supplier, price):  self.id\_detail = id\_detail  self.id\_supplier = id\_supplier  self.price = price  details = [  Detail(1, "Шайба", "Шайба М6", 1),  Detail(2, "Гайка", "Гайка М8", 2),  Detail(3, "Болт", "Болт М10", 3),  Detail(4, "Винт", "Винт саморез", 1),  Detail(5, "Шпилька", "Шпилька М12", 2)  ]  suppliers = [  Supplier(1, "Поставщик 1", "ул. Ленина, 1"),  Supplier(2, "Поставщик 2", "ул. Мира, 2"),  Supplier(3, "Поставщик 3", "ул. Победы, 3")  ]  details\_supplier = [  DetailsSupplier(1, 1, 10),  DetailsSupplier(2, 2, 20),  DetailsSupplier(3, 3, 30),  DetailsSupplier(4, 1, 40),  DetailsSupplier(5, 2, 50)  ]  # Запрос 1: Найти все детали, которые поставляются поставщиком с ID 1.  result = [detail for detail in details if detail.id\_supplier == 1]  print("Запрос 1: ", [f"ID: {detail.id\_detail}, Название: {detail.name}, Описание: {detail.description}, ID поставщика: {detail.id\_supplier}" for detail in result])  # Запрос 2: Найти все детали, которые поставляются поставщиком "Поставщик 2" и вывести их названия вместе с ценой.  result = [(detail.name, ds.price) for detail in details for ds in details\_supplier if detail.id\_detail == ds.id\_detail for supplier in suppliers if ds.id\_supplier == supplier.id\_supplier and supplier.name == "Поставщик 2"]  print("Запрос 2: ", result)  # Запрос 3: Найти всех поставщиков, которые поставляют деталь "Шайба".  result = [supplier for supplier in suppliers for detail in details for ds in details\_supplier if supplier.id\_supplier == ds.id\_supplier and ds.id\_detail == detail.id\_detail and detail.name == "Шайба"]  print("Запрос 3: ", [f"ID: {supplier.id\_supplier}, Название: {supplier.name}, Адрес: {supplier.address}" for supplier in result])    Выходные данные:  Запрос 1: ['ID: 1, Название: Шайба, Описание: Шайба М6, ID поставщика: 1', 'ID: 4, Название: Винт, Описание: Винт саморез, ID поставщика: 1']  Запрос 2: [('Гайка', 20), ('Шпилька', 50)]  Запрос 3: ['ID: 1, Название: Поставщик 1, Адрес: ул. Ленина, 1'] | |