**Московский государственный технический университет им. Н.Э. Баумана**

**Факультет ИУ Кафедра ИУ5**

**Курс «Основы информатики» Отчет по Рубежному контролю №1 Вариант Г25**

Выполнил студент группы ИУ5-36Б: Турланов В. Е.

Подпись и дата:

Проверил преподаватель каф.: Гапанюк Ю. Е.

Подпись и дата:

Москва, 2024 г

**Постановка задачи**

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)

* 1. Класс «Отдел», содержащий поля:
     + ID записи об отделе;
     + Наименование отдела.
  2. (Для реализации связи многие-ко-многим) Класс «Сотрудники отдела», содержащий поля:
     + ID записи о сотруднике;
     + ID записи об отделе.

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

**Вариант Г.**

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех отделов, у которых название начинается с буквы «А», и список работающих в них сотрудников.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов с максимальной зарплатой сотрудников в каждом отделе, отсортированный по максимальной зарплате.
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по отделам, сортировка по сотрудникам произвольная.

|  |  |  |
| --- | --- | --- |
| **Вариант** | **Класс 1** | **Класс 2** |
| 25 | Раздел | Документ |

**Текст программы**

class Razdel:  
 def \_\_init\_\_(self, id\_razdela, name):  
 self.id\_razdela = id\_razdela # Идентификатор раздела  
 self.name = name # Название раздела  
 self.documents = [] # Список документов в разделе  
  
 def add\_document(self, document):  
 self.documents.append(document) # Добавление документа в раздел  
  
  
class Dokument:  
 def \_\_init\_\_(self, id\_dokumenta, name, salary):  
 self.id\_dokumenta = id\_dokumenta # Идентификатор документа  
 self.name = name # Название документа  
 self.salary = salary # Признак, связанный с документом (зарплата)  
  
  
# Создание разделов  
razdel1 = Razdel(1, "Анализ")  
razdel2 = Razdel(2, "Бухгалтерия")  
razdel3 = Razdel(3, "Маркетинг")  
  
# Создание документов  
dokument1 = Dokument(1, "Отчет по продажам", 50000)  
dokument2 = Dokument(2, "Финансовый отчет", 60000)  
dokument3 = Dokument(3, "Анализ рынка", 70000)  
dokument4 = Dokument(4, "План продаж", 55000)  
  
# Добавление документов в разделы  
razdel1.add\_document(dokument3)  
razdel1.add\_document(dokument1)  
razdel2.add\_document(dokument2)  
razdel3.add\_document(dokument4)  
  
# Список всех разделов  
razdely = [razdel1, razdel2, razdel3]  
  
# Запрос 1: Вывести список всех разделов, у которых название начинается с буквы «А», и список документов в них.  
razdely\_s\_A = [  
 (razdel.name, [document.name for document in razdel.documents])  
 for razdel in razdely if razdel.name.startswith("А")  
]  
  
print("Разделы с названием на 'А' и их документы:")  
for name, documents in razdely\_s\_A:  
 print(f"Раздел: {name}, Документы: {', '.join(documents)}")  
  
# Запрос 2: Вывести список отделов с максимальной зарплатой документов в каждом отделе, отсортированный по максимальной зарплате.  
max\_salary = [  
 (razdel.name, max(document.salary for document in razdel.documents))  
 for razdel in razdely  
]  
  
max\_salary.sort(key=lambda x: x[1], reverse=True) # Сортировка по убыванию  
  
print("Разделы с максимальной зарплатой документов:")  
for name, salary in max\_salary:  
 print(f"Раздел: {name}, Максимальная зарплата: {salary}")  
  
# Запрос 3: Вывести список всех связанных документов и разделов, отсортированный по разделам.  
svyazannye\_dannye = [  
 (razdel.name, document.name)  
 for razdel in razdely for document in razdel.documents  
]  
  
svyazannye\_dannye.sort(key=lambda x: x[0]) # Сортировка по названиям разделов  
  
print("Связанные документы и разделы:")  
for razdel, dokument in svyazannye\_dannye:  
 print(f"Раздел: {razdel}, Документ: {dokument}"

Анализ результатов

Разделы с названием на 'А' и их документы:

Раздел: Анализ, Документы: Анализ рынка, Отчет по продажам

Разделы с максимальной зарплатой документов:

Раздел: Анализ, Максимальная зарплата: 70000

Раздел: Бухгалтерия, Максимальная зарплата: 60000

Раздел: Маркетинг, Максимальная зарплата: 55000

Связанные документы и разделы:

Раздел: Анализ, Документ: Анализ рынка

Раздел: Анализ, Документ: Отчет по продажам

Раздел: Бухгалтерия, Документ: Финансовый отчет

Раздел: Маркетинг, Документ:План продаж