

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Факультет “Информатика и системы управления”
Кафедра “Системы обработки информации и управления”



Дисциплина “Программирование и конструкции языков
программирования”

Отчет по рубежному контролю №1

Вариант: А-14

A handwritten signature in black ink, appearing to read 'Rakhmanov'.

Выполнил:
Рахманов Виталий
ИУ5-31Б

Преподаватель:
Гапанюк Ю.Е.

Условия рубежного контроля №1 по курсу Пик ЯП.

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.

Вариант А.

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по отделам, сортировка по сотрудникам произвольная.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов с суммарной зарплатой сотрудников в каждом отделе, отсортированный по суммарной зарплате.
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех отделов, у которых в названии присутствует слово «отдел», и список работающих в них сотрудников.

Вариант 14: CD-диск; Библиотека CD-дисков

Текст программы

```
from dataclasses import dataclass
from typing import List

@dataclass # класс для связи многие-ко-многим
class CDLibraryDisk:
    id: int
    libraryId: int
    diskId: int

@dataclass # класс данных для Библиотеки CD-дисков
class CDLibrary:
    id: int
    name: str
    location: str

@dataclass # класс данных для CD-диска
class CDDisk:
    id: int
    title: str
    artist: str
    genre: str
    price: float # для запроса 2
    year: int

def main():
    libraries = [
        CDLibrary(1, "Главный отдел музыки", "Центральный район"),
        CDLibrary(2, "Отдел классики", "Северный район"),
```

```

CDLibrary(3, "Рок-коллекция", "Западный район"), #не пишем слово "отдел" для 3 запроса
CDLibrary(4, "Джазовый отдел", "Восточный район"),
CDLibrary(5, "Поп-музыка отдел", "Южный район")
]

disks = [
    CDDisk(1, "The Dark Side of the Moon", "Pink Floyd", "Rock", 25, 1973),
    CDDisk(2, "Thriller", "Michael Jackson", "Pop", 19, 1982),
    CDDisk(3, "Kind of Blue", "Miles Davis", "Jazz", 22, 1959),
    CDDisk(4, "Symphony No. 9", "Beethoven", "Classical", 18, 1987),
    CDDisk(5, "Back in Black", "AC/DC", "Rock", 21, 1980),
    CDDisk(6, "The Wall", "Pink Floyd", "Rock", 24, 1979)
]

OneToMany = [ # один диск находится в одной библиотеке
    (lib, disk) for lib in libraries
    for disk in disks
    if (lib.id == 1 and disk.id in [1, 2]) or # главный отдел
        (lib.id == 2 and disk.id == 4) or # отдел классики
        (lib.id == 3 and disk.id in [1, 5, 6]) or # рок-коллекция
        (lib.id == 4 and disk.id == 3) or # джазовый отдел
        (lib.id == 5 and disk.id == 2) # поп-музыка отдел
    ]

ManyToMany = [ # связи многие-ко-многим
    CDLibraryDisk(1, 1, 1), # Главный отдел - The Dark Side of the Moon
    CDLibraryDisk(2, 1, 2), # Главный отдел - Thriller
    CDLibraryDisk(3, 2, 4), # Отдел классики - Beethoven
    CDLibraryDisk(4, 3, 1), # Рок-коллекция - The Dark Side of the Moon
    CDLibraryDisk(5, 3, 5), # Рок-коллекция - Back in Black
    CDLibraryDisk(6, 3, 6), # Рок-коллекция - The Wall
    CDLibraryDisk(7, 4, 3), # Джазовый отдел - Kind of Blue
    CDLibraryDisk(8, 5, 2), # Поп-музыка отдел - Thriller
    CDLibraryDisk(9, 1, 6), # Главный отдел - The Wall
    CDLibraryDisk(10, 3, 1), # Рок-коллекция - The Dark Side of the Moon
    ]
print("ЗАПРОС 1")
print("Список всех связанных библиотек и CD-дисков (один-ко-многим), отсортированный по
названиям библиотек:")

sortedLibraryDisks = sorted(OneToMany, key=lambda x: x[0].name)

for library, disk in sortedLibraryDisks:
    print(f"{library.name}: {disk.artist} - {disk.title}")

print("\n")

print("ЗАПРОС 2")
print("Список библиотек с суммарной стоимостью CD-дисков в каждой библиотеке:")

libraryCosts = {} # группируем по библиотекам и суммируем стоимость
for library, disk in OneToMany:
    if library.id not in libraryCosts:
        libraryCosts[library.id] = {'library': library, 'total_cost': 0}

```

```

libraryCosts[library.id]['total_cost'] += disk.price

sortedLibraryCosts = sorted(libraryCosts.values(), key=lambda x: x['total_cost'])

for item in sortedLibraryCosts:
    print(f"{item['library'].name}: ${item['total_cost']:.2f}")

print("\n")

print("ЗАПРОС 3")
print("Список библиотек с 'отдел' в названии и имеющиеся в них CD-диски (многие-ко-многим):")

departmentLibraries = [lib for lib in libraries if "отдел" in lib.name.lower()]

for library in departmentLibraries:
    # находим связи для текущей библиотеки
    libraryLinks = [link for link in ManyToMany if link.libraryId == library.id]

    # находим CD-диски по связям
    libraryDisks = [disk for link in libraryLinks
                     for disk in disks if disk.id == link.diskId]

    print(f"\n{library.name}:")
    if libraryDisks:
        for disk in libraryDisks:
            print(f" - {disk.artist} - {disk.title} ({disk.genre})")
    else:
        print(" - Нет CD-дисков")

print("\n")
print(f"Всего библиотек: {len(libraries)}")
print(f"Всего CD-дисков: {len(disks)}")
print(f"Связей один-ко-многим: {len(OneToMany)}")
print(f"Связей многие-ко-многим: {len(ManyToMany)}")

if __name__ == "__main__":
    main()

```

Результаты выполнения программы

ЗАПРОС 1

Список всех связанных библиотек и CD-дисков (один-ко-многим), отсортированный по названиям библиотек:

Главный отдел музыки: Pink Floyd - The Dark Side of the Moon

Главный отдел музыки: Michael Jackson - Thriller

Джазовый отдел: Miles Davis - Kind of Blue

Отдел классики: Beethoven - Symphony No. 9

Поп-музыка отдел: Michael Jackson - Thriller

Рок-коллекция: Pink Floyd - The Dark Side of the Moon

Рок-коллекция: AC/DC - Back in Black

Рок-коллекция: Pink Floyd - The Wall

ЗАПРОС 2

Список библиотек с суммарной стоимостью CD-дисков в каждой библиотеке:

Отдел классики: \$18.00

Поп-музыка отдел: \$19.00

Джазовый отдел: \$22.00

Главный отдел музыки: \$44.00

Рок-коллекция: \$70.00

ЗАПРОС 3

Список библиотек с 'отдел' в названии и имеющиеся в них CD-диски (многие-ко-многим):

Главный отдел музыки:

- Pink Floyd - The Dark Side of the Moon (Rock)
- Michael Jackson - Thriller (Pop)
- Pink Floyd - The Wall (Rock)

Отдел классики:

- Beethoven - Symphony No. 9 (Classical)

Джазовый отдел:

- Miles Davis - Kind of Blue (Jazz)

Поп-музыка отдел:

- Michael Jackson - Thriller (Pop)

Всего библиотек: 5

Всего CD-дисков: 6

Связей один-ко-многим: 8

Связей многие-ко-многим: 10

Анализ реализации требований

1. Создание классов данных

CDLibrary - представляет библиотеку CD-дисков (аналог "Отдела")

CDDisk - представляет CD-диск с количественным признаком price (аналог "Сотрудника")

CDLibraryDisk - вспомогательный класс для связи многие-ко-многим

2. Тестовые данные

Создано 5 объектов библиотек и 6 объектов CD-дисков со связями по идентификаторам.

3. Реализация запросов

Запрос 1 - "Библиотека CD-дисков" и "CD-диск" связаны соотношением один-ко-многим. Вывод списка всех связанных библиотек и CD-дисков, отсортированный по библиотекам. Используется сортировка с lambda-функцией.

Запрос 2 - "Библиотека CD-дисков" и "CD-диск" связаны соотношением один-ко-многим. Вывод списка библиотек с суммарной стоимостью CD-дисков в каждой библиотеке, отсортированный по суммарной стоимости. Используется словарь для группировки и суммирования.

Запрос 3 - "Библиотека CD-дисков" и "CD-диск" связаны соотношением многие-ко-многим. Вывод списка всех библиотек, у которых в названии присутствует слово "отдел", и список CD-дисков в них.