ИУ5-32Б

Ушаков Л. И.

Отчёт по РК1 по курсу ПиК ЯП, вариант 14Б

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 её выполнения.

Листинг кода:

```
from dataclasses import dataclass
from itertools import groupby

@dataclass
class CDDisk:
    '''
    Класс объекта "CD-диск"
    '''
    id: int
    library_id: int
    name: str
    capacity: int

@dataclass
class CDLibrary:
    '''
    Класс объекта "Библиотека CD-дисков"
    '''
    id: int
    name: str

@dataclass
class CDDiskInCDLibrary:
    '''
    Класс связи многие-ко-многим объектов "CD-диск" и "Библиотека CD-дисков"
    '''
    disk_id: int
    library_id: int



def main():
    libraries = [
        CDLibrary(1, "Main Library"),
        CDLibrary(2, "Branch Library"),
        CDLibrary(3, "University Archive")
    ]

    disks = [
        CDDisk(1, 1, "Windows Pack", 700),
        CDDisk(2, 1, "Classical Pack", 650),
        CDDisk(3, 2, "Movie Pack", 4700),
```

```
        CDDisk(4, 3, "Backup Collection", 4900),
        CDDisk(5, 1, "Linux Pack", 780),
        CDDisk(6, 2, "Pop Pack", 640),
        CDDisk(7, 2, "Educational Pack", 3500),
        CDDisk(8, 3, "Photo Archive", 4600),
        CDDisk(9, 1, "Software Pack", 1200),
        CDDisk(10, 3, "Database Dump", 5000)
    ]

    library_disks = [
        # Основные связи
        CDDiskInCDLibrary(1, 1),
        CDDiskInCDLibrary(2, 1),
        CDDiskInCDLibrary(3, 2),
        CDDiskInCDLibrary(4, 3),
        CDDiskInCDLibrary(5, 1),
        CDDiskInCDLibrary(6, 2),
        CDDiskInCDLibrary(7, 2),
        CDDiskInCDLibrary(8, 3),
        CDDiskInCDLibrary(9, 1),
        CDDiskInCDLibrary(10, 3),

        # Дополнительные перекрёстные связи
        CDDiskInCDLibrary(2, 2),
        CDDiskInCDLibrary(1, 3),
        CDDiskInCDLibrary(7, 1),
        CDDiskInCDLibrary(9, 2),
    ]

    # Функции-помощники для улучшения читабельности кода
    def disk_capacity(entry): return entry[0].capacity
    def disk_name(entry): return entry[0].name
    def library_name(entry): return entry[1].name

    # Таблица связей один-ко-многим
    table = list(map(lambda disk: (disk, next(filter(lambda lib:
disk.library_id == lib.id, libraries))), disks))

    # Вывод всех дисков (CDDisk), отсортированных по вместительности
(capacity)
    print("------- Запрос Б-1 -------")
    print(*map(lambda entry:
f"{disk_name(entry):<32}\t{f'{disk_capacity(entry)}
MB':<8}\t{library_name(entry)}", sorted(table, key=lambda entry:
disk_capacity(entry))), sep='\n')
    print()
```

```
    # Вывод всех библиотек (CDLibrary), отсортированный по количеству входящих
в них дисков (CDDisk)
    print("------- Запрос Б-2 -------")
    print(*map(lambda entry: f"{entry[0]:<32}\t{entry[1]}", sorted(map(lambda
entry: (entry[0], len(list(entry[1]))), groupby(sorted(table, key=lambda
entry: library_name(entry)),  key=lambda entry: library_name(entry))),
key=lambda entry: entry[1])), sep='\n')
    print()

    # Таблица связей многие-ко-многим
    table.clear()
    table = list(map(lambda link: (next(filter(lambda disk: link.disk_id ==
disk.id, disks)), next(filter(lambda lib: link.library_id == lib.id,
libraries))), library_disks))

    # Вывод всех дисков (CDDisk), названия которых оканчиваются на "Pack"
    print("------- Запрос Б-3 -------")
    print(*map(lambda entry: f"{disk_name(entry):<32}\t{library_name(entry)}",
sorted(filter(lambda entry: disk_name(entry).endswith("Pack"), table),
key=lambda entry: disk_name(entry))), sep='\n')

if __name__ == "__main__":
    main()
```

Результат выполнения:

```
------- Запрос Б-1 -------
Pop Pack                         640 MB     Branch Library
Classical Pack                   650 MB     Main Library
Windows Pack                     700 MB     Main Library
Linux Pack                       780 MB     Main Library
Software Pack                    1200 MB    Main Library
Educational Pack                 3500 MB    Branch Library
Photo Archive                    4600 MB    University Archive
Movie Pack                       4700 MB    Branch Library
Backup Collection                4900 MB    University Archive
Database Dump                    5000 MB    University Archive

------- Запрос Б-2 -------
Branch Library                   3
University Archive               3
Main Library                     4

------- Запрос Б-3 -------
```

Classical Pack	Main Library
Classical Pack	Branch Library
Educational Pack	Branch Library
Educational Pack	Main Library
Linux Pack	Main Library
Movie Pack	Branch Library
Pop Pack	Branch Library
Software Pack	Main Library
Software Pack	Branch Library
Windows Pack	Main Library
Windows Pack	University Archive