

Ćwiczenie 7: Prosta baza danych

Cel: celem ćwiczenia jest napisanie programu obsługującego prostą bazę danych.

Struktura danych:

Pisząc aplikację, w pierwszej kolejności musimy określić strukturę danych na której program będzie wykonywał operacje.

Sugerowana baza danych opisuje kolekcje albumów muzycznych (np. płyt CD). Każda płyta opisywana jest w następujący sposób:

Album (krotka):

Pole	Typ
Tytuł	String (tekst)
Autor	String (tekst)
Utwory	Lista elementów typu string (tytuły kolejnych utworów)
Rok wydania	Int (liczba całkowita)
<i>Inne wg. uznania</i>	
...	

Pojedynczy album reprezentowany jest w programie jako krotka (tuple) kolejnych wartości opisujących dany album, np.:

```
('Queen', 'A night at the opera', ['tyt1', 'tyt2'], 1975)
```

Kolekcja albumów przechowywana jest w programie jako słownik (dictionary). Do słownika wstawiamy krotki opisujące pojedyncze albumy.

Kolekcja albumów (słownik)

Indeks	Album
1	Album 1 (krotka typu jw.)
2	Album 2
3	Album 3
4	Album 4
...	...

Kluczem słownika jest indeks. Przy dodawaniu kolejnego albumu nadajemy mu indeks.

Początkowo używamy indeksów 1,2,3... itd. Przykład:

```
{ 1: ('Queen', 'A Night at the Opera', ['tytuł1', 'tytuł2'], 1975),  
  2: ('Queen', 'The Miracle', ['Inny tytuł', '...'], 1988),  
  3: ('Pink Floyd', 'The Wall', ['...', '...'], 1979) }
```

Ogólnie przy dodawaniu nowego albumu należy wyciągnąć z bazy wszystkie indeksy (tzn. wszystkie klucze słownika – robimy to poleceniem `indeksy = D.keys()`, gdzie D jest słownikiem) i jako nowy indeks wziąć wartość maksymalną z wszystkich indeksów powiększoną o 1.

Wersja 1.

Kolekcja albumów (nasz słownik) może być zapisany na dysku do pliku. Możliwe jest też odczytanie danych zapisanych w pliku.

Program powinien umożliwiać:

- Dodawanie nowych albumów do kolekcji
- Zapisywanie bazy do pliku (zakładamy, że jest tylko jeden plik bazy)
- Otwieranie bazy z pliku.
- Wypisywanie na ekranie całej bazy (w przyjaznym formacie tekstowym). Wypisywane mają być np. indeks, tytuł albumu i rok wydania (bez listy utworów)
- Wypisywanie na ekranie albumu (po podaniu indeksu). Wypisywany jest tytuł, rok wydania i lista utworów.
- Wyszukiwania:
 - Wyszukiwanie utworu o podanej nazwie (program pokazuje listę albumów, w których jest utwór o takiej nazwie)
 - Wyszukiwanie wszystkich albumów danego wykonawcy
 - Itp.

Uwagi do rozwiązania:

- Należy pisać nowe funkcje realizujące kolejne funkcjonalności:

Wersja 1.

Pisanie programu:

1. Podstawowy szkielet programu.

Główna pętla programu powinna wyświetlać listę komend, pobierać z klawiatury wybraną komendę i wykonywać dane działanie:

```
# tworzymy słownik, który będzie używany w programie
b = {}
# główna pętla programu
while True:
    wypiszKrotkieInfo(b)      # np. wypisanie ilości albumów w kolekcji

    opcja = wyborOpcji()      # wypisanie „menu” i pobranie wyboru
    if opcja == 0:
        break                 # wyjście z programu
    elif opcja == 1:
        b = załadujZPliku()   # kolejne operacje zaimplementowane jako funkcje
    elif opcja == 2:
        zapiszDoPliku(b)      # zapisanie do pliku
    elif opcja == 3:
        wypiszBaze(b)
    elif opcja == 4:
        album = wprowadzAlbum() # wprowadzenie danych z klawiatury
        dodajAlbumDoBazy(b,album) # dodanie wprowadzonego do bazy
    elif opcja ==5:
        usunAlbumZBazy(b)     # usunięcie z bazy (wg. podanego indeksu)
    else:
        pass
```

Program po rozpoczęciu program „wygląda: np. następująco:

```
Ilość pozycji w bazie: 0
Wybierz opcję:
1. Załaduj z pliku
2. Zapisz do pliku
3. Wypisz całą bibliotekę
4. Dodaj album
5. Usuń album
6. Wyszukaj...
0. Koniec
Wybierz polecenie:
```

Wersja 1.

Zapis i odczyt z pliku możliwy jest dzięki modułowi `pickle`. Na początku trzeba zaimportować moduł:

```
import pickle
```

Funkcja zapisująca cały słownik do pliku wygląda następująco (opcja 2 programu):

```
def zapiszDoPliku(baza):  
    pickle.dump(baza, open('baza.p', 'wb' ) )
```

Funkcja odczytująca słownik z pliku (opcja 1 programu):

```
def zaladujZPliku():  
    baza = pickle.load(open('baza.p', 'rb'))  
    return baza
```