

OPERACJE NA PLIKACH

Nadanie wartości początkowych do zmiennej przechowującej dane z pliku
`zmienna = open("ścieżka_do_pliku", "tryb_dostępu")`

Ważne funkcje:

```
open ("ścieżka_do_pliku", "tryb_dostępu") - otwieranie pliku
w miejsce tryb_dostępu należy podać jedną z wartości:
    r - plik do odczytu.
    a - plik do zapisu (dopisanie).
    w - plik do zapisu (nadpisanie).
```

Ważne metody:

```
close() - zamykanie pliku.
read() - czytanie otwartego pliku.
write("treść") - wpisanie do zawartości pliku.
```

Wywołanie metody: `plik.metoda`.

Pętla `for` przechodząca po elementach wersach w pliku
`for zmienna in plik:`
instrukcje do wykonania w pętli
Przydatna metoda na napisach:

```
napis.split() - dzieli słowa oddzielone spacjami na listę.
```

Zadanie 1 (+). Napisz skrypt, który odczyta dane z pliku `liczby.txt` oraz wyświetli największą i najmniejszą z nich na ekranie.

Wejście: Zmienna `liczba` przechowująca kolejne dane pobrane z pliku `liczby.txt`.

Wyjście: Na ekranie pojawia się informacja, które liczby z pliku `liczby.txt` są najmniejsze, a które największe.

Warunki poprawności zadania: Upewnij się, że program prawidłowo pobiera dane z pliku oraz wyświetla prawidłowe wartości.

Zadanie 2. Napisz skrypt, który będzie zawierał listę liczb. Następnie wpisze do pliku `nieparzyste.txt`, które z podanych liczb są liczby nieparzyste oraz do pliku `parzyste.txt` podane liczby parzyste.

Wejście: Zmienna `lista` przechowująca liczby całkowite.

Wyjście: Pliki `parzyste.txt` i `nieparzyste.txt` zawierające odpowiednie wartości.

Warunki poprawności zadania: Upewnij się, że program wpisuje odpowiednie wartości do odpowiednich plików. Upewnij się, że program działa prawidłowo dla różnych danych.

Zadanie 3 (+). Napisz skrypt, który zasymuluje losowanie Lotto, tzn.

wylosuje 6 liczb z zakresu od 1 do 49 w taki sposób, aby żadna się nie powtarzała. Następnie wpisze do pliku `lotto.txt` wyniki losowania.

Podpowiedź: W celu wylosowania liczby możesz zaimportować bibliotekę `random` oraz użyć funkcji `randint`. Pseudolosowanie liczby całkowitej z zakresu od `liczba1` do `liczba2`: `random.randint(liczba1, liczba2)`

Wejście: Brak.

Wyjście: Pliki `lotto.txt` zawierający pseudolosowe liczby z zakresu 1 do 49.

Warunki poprawności zadania: Upewnij się, że program wpisuje odpowiednie liczby do pliku, które się nie powtarzają.

Zadanie 4. Napisz skrypt, który sprawdzi czy w pliku baza.txt się osoba o podanym imieniu i nazwisku. Wypisz odpowiedni komunikat na ekranie. Zdefiniuj słownik w którym będziesz przechowywał odpowiednie informacje na temat osób.

Wejście: Zmienne imię i nazwisko przechowujące pobrane dane.

Wyjście: Na ekranie pojawia się informacja czy osoba o podanym imieniu i podanym nazwisku istnieje czy nie.

Warunki poprawności zadania: Upewnij się, że program przeszukuje prawidłowo dane i wypisuje prawidłowe informację.

Zadanie 5 (+). Napisz skrypt, który wyświetli informację ile jest liter (dużych i małych), cyfr, znaków białych (spacja, nowa linia, tabulator) oraz pozostałych znaków w pliku wiersz.txt lub innego przez siebie wybranego pliku.

Wejście: Zmienna dane przechowujące dane pobrane z pliku.

Wyjście: Na ekranie pojawia się odpowiednie informacje.

Warunki poprawności zadania: Upewnij się, że program wypisuje prawidłowe informację dla plików o różnych wartościach.

Zadanie 6. Palindromem nazywamy słowo, które czytane od lewej do prawej i od prawej do lewej jest brzmi taki samo. Przykładowo słowo kajak jest palindromem.

Napisz skrypt, który wczyta dane z pliku znaki.txt oraz wypisze na ekranie informację na temat tego ile jest palindromów oraz wypisze do pliku palindrom.txt wszystkie wczytane palindromy.

Wejście: Zmienna napis, która przechowuje odpowiedni ciąg znaków.

Wyjście: Na ekranie pojawia się odpowiednia liczba palindromów oraz do plik palindrom.txt zawiera wszystkie słowa będące, które są palindromem z pliku znaki.txt.

Warunki poprawności zadania: Upewnij się, że program wypisuje prawidłową liczbę palindromów oraz wpisuje wszystkie palindromy do pliku palindrom.txt.

Zadanie 7 (+). Pewien szyfr polega na zastępowaniu każdej litery tekstu inną literą, oddaloną od niej o stałą liczbę pozycji w alfabecie, bez rozróżniania wielkości liter. Przykładowo dla przesunięcia 3 litera B zostanie zastąpiona literą E natomiast litera Y literą B.

W pliku szyfr.txt znajdują się słowa zaszyfrowane tym szyfrem (pierwsza kolumna) oraz liczby odpowiadające przesunięciom (druga kolumna). Napisz skrypt, który odszyfruje słowa z pliku szyfr.txt oraz zapisze je do pliku odszyfr.txt.

Uwaga: Przyjmij, że słowa składają się tylko z dużych liter i nie zawierają polskich znaków.

Wejście: Zmienna szyfr, która przechowywać kolejne pobrane napisy i zmienna przesuniecie przechowująca odpowiednie przesunięcie.

Wyjście: Plik odszyfr.txt zawiera odszyfrowane dane.

Warunki poprawności zadania: Upewnij się, że program wypisuje prawidłowo odszyfrowuje dane. Plik odszyfr.txt powinien mieć zawartość składającą się z prawidłowo odszyfrowanych polskich słów.

TESTY

Uruchomienie

Linia poleceń:

Zainstaluj Pytest. <https://docs.pytest.org/en/latest/getting-started.html>

Wpisz polecenie: `pytest plik`

w przypadku niepowodzenia spróbuj: `python -m pytest plik`

plik musi mieć rozszerzenie: `*.py`

Pycharm:

Zainstaluj Pytest.

Konfiguracja interpretera uruchamiającego testy

File → Settings → Tools → Python Integrated Tools → W Default test runner

wybierz pytest → OK.

Uruchamianie testu

Kliknij prawym przyciskiem na skrypt → Run 'nazwa'.

Tworzenie testu

```
def funkcja():  
    # instrukcje do wykonania  
    assert warunek1  
    # instrukcje do wykonania  
    assert warunek2  
    # ...  
    assert warunekN  
    # ...
```

funkcja - musi się zaczynać od przedrostka test.

warunek - test będzie spełniony jeżeli warunek będzie prawdziwy.

Zadanie 1 (+). Napisz skrypt, który będzie zawierał małą bazę danych przechowującą dane osób (imię, nazwisko, adres), gdzie adres jest słownikiem, których klucze ustal wg własnego uznania. Dane powinny być przechowywane w liście zawierające słowniki. Zdefiniuj funkcje, które umożliwią wykonywać operację CRUDS:

funkcja, która umożliwi(a) dodanie nowej osoby (CREATE). Funkcja jako argument powinna przyjąć dane, które powinny być dodane oraz zwrócić listę zawierającą nowo dodaną daną.

funkcja/funkcje, która zwróci(ca) informację na temat określonych danych (READ). Funkcja jako argument powinna przyjąć jakieś dane i zwrócić jakąś informację odnośnie pobranych danych. Przykładowo zwracamy informację gdzie mieszka osoba, której imię i nazwisko zostało podane jako argument. Możesz stworzyć kilka funkcji zwracających informację wedle własnego uznania.

funkcja/funkcje, która umożliwi(a) zmianę danych (UPDATE). Funkcja jako argument powinna przyjąć dane na podstawie, których chcemy zmienić wartości i wartości, na które mają być one zmienione oraz zwraca listę z odpowiednio zawierającą wszystkie dane z odpowiednimi modyfikacjami. Przykładowo zmiana wieku (podajemy wiek jako argument) osób o podanym imieniu i nazwisku. Możesz stworzyć kilka funkcji zwracających informację wedle własnego uznania.

funkcja/funkcje, która umożliwi(a) usunięcie istniejącej osoby (DELETE). Funkcja jako argument powinna przyjąć informację na podstawie, których chcemy usunąć dane oraz zwrócić nową listę nie zawierającą odpowiednich danych. Przykładowo usunięcie osób, które mają na nazwisko

Kowalski. Możesz stworzyć kilka funkcji zwracających informację wedle własnego uznania.

funkcje wyszukiwania wyszukiwania (SEARCH), które wyszukują dane spełniające określone, kryteria oraz zwracają listę przechowującą słowniki, które je spełniają np.:

operacja znajdowania osób, które mieszkają w podanej miejscowości.

operacja znajdowania osób, które są starsze/młodsze niż podany wiek.

Wszystkie funkcje mogą też przyjąć aktualną bazę danych jako argument i inne dodatkowe rzeczy jeśli uznasz to za konieczne. Przetestuj działanie utworzonych funkcji.

Wejście: Zmienna lista przechowująca listę zawierająca dane słownikowe, które będą przechowywać informacje o osobach.

Wyjście: Na ekranie pojawia się odpowiednia informacja odnośnie osób.

Warunki poprawności zadania: Powinny zostać wykonane odpowiednie testy wywołań funkcji. Upewnij się, że zdefiniowane funkcje działają prawidłowo, tzn. prawidłowo działa dodawanie, czytanie, zmienianie i usuwanie danych o osobach oraz inne operacje.

Zadanie 2. Napisz skrypt, który będzie zawierał małą bazę danych przechowującą dane uczniów wraz z ich ocenami z poszczególnych przedmiotów. Dane powinny być przechowywane w liście zawierającej słowniki. Klucze ustal wg własnego uznania. Zdefiniuj funkcje, które umożliwią wykonywać operację CRUDS:

funkcja, która umożliwi(a) dodanie nowego ucznia (CREATE). Funkcja jako argument powinna przyjąć dane, które powinny być dodane oraz zwrócić listę zawierającą nowo dodane dane.

funkcja/funkcje, która zwróci(ca) informację na temat określonych danych (READ). Funkcja jako argument powinna przyjąć jakieś dane i zwrócić jakąś informację odnośnie pobranych danych. Przykładowo zwracamy informację jakie oceny ma uczeń, którego numer PESEL został podany jako argument. Możesz stworzyć kilka funkcji zwracających informację wedle własnego uznania.

funkcja/funkcje, która umożliwi(a) zmianę danych (UPDATE). Funkcja jako argument powinna przyjąć dane na podstawie, których chcemy zmienić wartości i wartości, na które mają być one zmienione oraz zwraca listę z odpowiednio zawierającą wszystkie dane z odpowiednimi modyfikacjami. Przykładowo zmiana ocen (podajemy oceny jako argument) uczniów o podanym numerze PESEL. Możesz stworzyć kilka funkcji zwracających informację wedle własnego uznania.

funkcja/funkcje, która umożliwi(a) usunięcie istniejącego ucznia (DELETE). Funkcja jako argument powinna przyjąć informację na podstawie, których chcemy usunąć dane oraz zwrócić nową listę nie zawierającą odpowiednich danych. Przykładowo usunięcie uczniów, którzy mają na nazwisko Kowalski.

funkcje wyszukiwania wyszukiwania (SEARCH), które wyszukują dane spełniające określone, kryteria oraz zwracają listę przechowującą dane (słowniki), które je spełniają, np.:

operacja znajdowania uczniów, którzy posiadają średnią arytmetyczną ocen od podanej.

operacja znajdowania przedmiotu z którego średnia arytmetyczna jest największa.

Wszystkie funkcje mogą też przyjąć aktualną bazę danych jako argument i inne dodatkowe rzeczy jeśli uznasz to za konieczne. Przetestuj działanie utworzonych funkcji.

Wejście: Zmienna lista przechowująca listę zawierająca dane słownikowe, które będą przechowywać odpowiednie dane.

Wyjście: Na ekranie pojawia się odpowiedni wynik.

Warunki poprawności zadania: Powinny zostać wykonane odpowiednie testy wywołań funkcji. Upewnij się, że zdefiniowane funkcje działają prawidłowo, tzn. prawidłowo działa dodawanie, czytanie, zmienianie i usuwanie danych o osobach oraz inne operacje.