

PROGRAMOWANIE FUNKCYJNE W PYTHONIE

ĆWICZENIA DODATKOWE

MODUŁ 4

AltKom Akademia S.A., materiały własne

1 PROGRAMOWANIE FUNKCYJNE

ĆWICZENIE 1.1:

Parametry

UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
 - definiowania funkcji
 - przekazywania argumentów do funkcji

CELE I ZADANIA:

- Utwórz funkcję, która posiada tylko obowiązkowe parametry i wypisuje wartości podanych argumentów
- Przemyśl, w jaki sposób można wywołać tę funkcję, posiadając dwa słowniki zawierające klucze odpowiadające nazwom parametrów
- Każdy parametr funkcji ma odpowiadający mu klucz **w którymś** ze słowników, ale żaden z tych słowników nie posiada kluczy odpowiadających **wszystkim** parametrom funkcji

ALGORYTM WYKONANIA:

- Napisz funkcję, która posiada cztery obowiązkowe parametry (np. a , b , c i d)
- Jej zadaniem jest wypisanie ich wartości
- Utwórz dwa słowniki (2- i 3-elementowe), których klucze odpowiadają nazwom parametrów
 - pierwszy słownik posiada klucze a i b
 - drugi słownik posiada klucze b , c i d
- Zastanów się, jak wywołać funkcję z wartościami, które znajdują się w słownikach – do funkcji powinny być przekazane następujące wartości:
 - parametr a – ze słownika pierwszego
 - parametr b – ze słownika pierwszego
 - parametr c – ze słownika drugiego
 - parametr d – ze słownika drugiego
- W rozwiązaniu wykorzystaj operator wypakowania
- Jak należy zmienić wywołanie, aby do funkcji przekazać wartości:
 - parametr a – ze słownika pierwszego
 - parametr b – ze słownika drugiego
 - parametr c – ze słownika drugiego
 - parametr d – ze słownika drugiego

ĆWICZENIE 1.2:

NWD i NWW

UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
 - definiowania i wywoływania funkcji

CELE I ZADANIA:

- Napisz program, który na podstawie dwóch podanych liczb całkowitych wyznaczy dla nich NWD (największy wspólny dzielnik) i NWW (największą wspólną wielokrotność)
- Do wyliczenia NWD i NWW zdefiniuj funkcje
- W rozwiązaniu wykorzystaj wyniki ćwiczenia 2.1 z modułu 3, rozkładającego liczby na czynniki pierwsze

ALGORYTM WYKONANIA:

- Korzystając z rozwiązania ćwiczenia 2.4 z modułu 3 napisz funkcję, która podaną jako parametr liczbę rozłoży na czynniki pierwsze i zwróci je w postaci listy
- Napisz dwie funkcje, którym jako parametry zostaną podane dwie liczby całkowite
- Ich zadaniem będzie wyliczenie odpowiednio NWD i NWW
- Rozłóż obie liczby na czynniki pierwsze
- Usuń z drugiej listy wszystkie te liczby, które występują w liście pierwszej
- NWD można obliczyć przemnażając usunięte liczby
- NWW można obliczyć przemnażając pierwszą liczbę przez liczby pozostałe na drugiej liście
- Wprowadź z klawiatury dwie liczby całkowite i przetestuj w działaniu funkcje wyliczające NWD i NWW
- Wyniki obliczeń przedstaw na ekranie

ĆWICZENIE 1.3:

Pechowe dni

UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
 - definiowania i wywoływania funkcji
 - stosowania pętli i wyrażeń listowych

CELE I ZADANIA:

- Wykorzystując algorytm wiecznego kalendarza (z wcześniejszych ćwiczeń z prezentacji) napisz program, który zwróci listę nazw dni tygodnia na które najczęściej przypada 13-ty dzień miesiąca (w zadanym przedziale czasu, np. XXI w.)

ALGORYTM WYKONANIA:

- Utwórz listę z nazwami dni tygodnia
- Napisz funkcję wyliczającą dzień tygodnia dla podanej daty (roku, miesiąca i dnia)
- Zlicz w zadanym przedziale czasu, ile razy w danym dniu tygodnia przypada 13-ty dzień miesiąca
- Wyniki zapisz w liście
- Wyszukaj wartość maksymalną
- Zwróć w postaci listy nazwy dni tygodnia, które wystąpiły najczęściej

ĆWICZENIE 1.4:**Odległość***

Rozwiązanie zadania może się przydać do projektu zaliczeniowego :-)

UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
 - definiowania i wywoływania funkcji
 - wykorzystania biblioteki *geopy*

CELE I ZADANIA:

- Napisz funkcję, która dla dwóch podanych lokalizacji, określonych przez szerokość i długość geograficzną (wyrażone w stopniach) obliczy odległość pomiędzy nimi (w kilometrach)
- Spróbuj zadanie zrealizować w kilku wariantach:
 - załóż, że powierzchnia Ziemi ma kształt sfery o promieniu 6371.009 km (możesz też tę wartość odczytać po zaimportowaniu biblioteki *geopy*) i oblicz odległość pomiędzy dwoma punktami na sferze, stosując podane wzory (p. algorytm wykonania)
 - użyj biblioteki *geopy* i odszukaj funkcję, która oblicza odległość sferyczną
 - użyj biblioteki *geopy* i odszukaj funkcję, która oblicza odległość geodezyjną
- Porównaj otrzymane wyniki

ALGORYTM WYKONANIA:

- Napisz funkcję, która jako argumenty przyjmie dwie lokalizacje miejscowości i obliczy jaką je dzieli odległość (w km)
- Lokalizację powinna opisywać 2-elementowa krotka (odpowiednio szerokość i długość geograficzna podana w stopniach)
- CZĘŚĆ 1:
 - zamień otrzymane współrzędne w stopniach na radiany (możesz odszukać gotową funkcję w bibliotece matematycznej)
 - wtedy, jeśli oznaczymy zmiennymi:

* $szer_1$ – szerokość geograficzna pierwszej lokalizacji (w radianach)

* $dlug_1$ – długość geograficzna pierwszej lokalizacji (w radianach)

* $szer_2$ – szerokość geograficzna drugiej lokalizacji (w radianach)

* $dlug_2$ – długość geograficzna drugiej lokalizacji (w radianach)

to do obliczenia odległości możesz wykorzystać następujące wzory:

$$a = \sin^2\left(\frac{szer_2 - szer_1}{2}\right) + \cos(szer_1) * \cos(szer_2) * \sin^2\left(\frac{dlug_2 - dlug_1}{2}\right)$$

$$odleglosc = 2 * R * \arctg\left(\sqrt{\frac{a}{1-a}}\right)$$

gdzie: R – oznacza promień Ziemi (możesz użyć wartości 6371.009 km)

- do obliczenia powyższych wyrażeń posłuż się biblioteką matematyczną (zapoznaj się z funkcjami *sin*, *cos*, *sqr*t oraz *atan2*)
- CZEŚĆ 2:
 - napisz funkcję, która zrealizuje te same zadania, co funkcja w części 1, ale wykorzystaj do tego bibliotekę *geopy*
 - do zaimportowania biblioteki możesz zastosować polecenie:
`from geopy import distance`
 - w dokumentacji do biblioteki *geopy*, dostępnej pod adresem:
<https://geopy.readthedocs.io/en/stable/>
odszukaj funkcję, która obliczy odległość pomiędzy dwoma lokalizacjami, przy założeniu, że powierzchnia Ziemi ma kształt sfery
 - aby ją wywołać poprzedź jej nazwę przedrostkiem *distance* i kropką
- CZEŚĆ 3:
 - korzystając z biblioteki *geopy* odszukaj funkcję, która obliczy odległość geodezyjną pomiędzy dwoma lokalizacjami
- Wywołaj funkcje opisane w częściach: 1, 2 i 3 i porównaj otrzymane wyniki

ĆWICZENIE 1.5:

Lokalizacja*

Rozwiązanie zadania może się przydać do projektu zaliczeniowego :-)

UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
 - definiowania i wywoływania funkcji
 - wykorzystania biblioteki *geopy*

CELE I ZADANIA:

- Napisz dwie funkcje, z których:
 - pierwsza – na podstawie podanego słownego opisu lokalizacji zwróci jej współrzędne geograficzne (szerokość i długość geograficzną)
 - druga – odwrotnie, na podstawie współrzędnych geograficznych, zwróci opis słowny podanej lokalizacji
- Wykorzystaj do tego możliwości biblioteki *geopy*

ALGORYTM WYKONANIA:

- Napisz funkcję, która jako argument przyjmie słowny opis lokalizacji (np. *Poznań*, *Collegium Da Vinci*) i na tej podstawie dokona lokalizacji i zwróci współrzędne geograficzne (szerokość i długość geograficzną) opisanego miejsca
- Do tego celu wykorzystaj bibliotekę *geopy*
 - zaimportuj z tej biblioteki moduł *geocoders* – możesz użyć polecenia:
`from geopy import geocoders`
 - w dokumentacji do biblioteki *geopy*, dostępnej pod adresem:
<https://geopy.readthedocs.io/en/stable/>
odszukaj funkcję, która potrafi zlokalizować opisane miejsce po nazwie
 - jako geolokalizatora możesz użyć usługi *Nominatim*
 - nazwę usługi trzeba poprzedzić przedrostkiem *geocoders* i kropką
- Aby sprawdzić działanie funkcji napisz drugą funkcję, która będzie działała odwrotnie
 - na podstawie podanych współrzędnych geograficznych zwróci opis słowny podanej lokalizacji
 - do tego celu wykorzystaj odpowiednią funkcję z biblioteki *geopy*
- Przetestuj działanie obu funkcji podając opisy tekstowe kilku miejsc i sprawdź, czy zwrócone przez funkcję opisy są zbieżne

ĆWICZENIE 1.6:**Stacje pomiarowe, stanowiska pomiarowe, dane pomiarowe***

Rozwiązanie zadania może się przydać do projektu zaliczeniowego :-)

UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
 - tworzenia i wywoływania własnych funkcji
 - użycia wbudowanych funkcji, które jako argumenty przyjmują inne funkcje

CELE I ZADANIA:

- Napisz program, o analogicznej funkcjonalności, jak w ćwiczeniach dodatkowych 2.1, 2.2 i 2.3 z modułu 3
- Zamknij zawarty tam kod w funkcjach
- Zastąp wyrażenia listowe użyciem wbudowanych funkcji *filter* oraz *map*
- Tam, gdzie to możliwe, użyj funkcji *lambda*

ALGORYTM WYKONANIA:

- Utwórz “funkcyjną wersję” kodu z ćwiczeń 2.1, 2.2 i 2.3 z modułu 3
- W tym celu utwórz funkcje umożliwiające wybór danych zgodnie z zadanymi kryteriami, np.:
 - funkcję, która na podstawie podanej nazwy lokalizacji (nazwy miejscowości) zwróci listę znajdujących się tam stacji pomiarowych
 - funkcję, która na podstawie podanej listy stanowisk pomiarowych zwróci listę nazw parametrów monitorowanych na tych stanowiskach
 - funkcję, która na podstawie podanej daty oraz listy danych pomiarowych zwróci listę danych pomiarowych pochodzących tylko z podanego dnia
 - funkcję, która na podstawie podanej daty oraz listy danych pomiarowych zwróci statystyki dla niepustych danych z podanego dnia:
 - * wartość najmniejszą wraz z informacją o której godzinie ta wartość została zarejestrowana
 - * wartość największą wraz z informacją o której godzinie ta wartość została zarejestrowana
 - * wartość średnią z danych niepustych (należy pominąć wartości *None*)
 - w powyższych funkcjach należy zastąpić użycie wyrażań listowych funkcjami wbudowanymi *filter* oraz *map*
 - tam, gdzie to możliwe, jako argumenty tym funkcjom przekaz wyrażenia *lambda*
 - zwróć uwagę, że te funkcje wbudowane nie zwracają listy, tylko iterator – takie rozwiązanie będzie mniej obciążające pamięć
- Sprawdź działanie funkcji na przykładowych danych

ĆWICZENIE 1.7:

Najbliższe stacje pomiarowe*

Rozwiązanie zadania może się przydać do projektu zaliczeniowego :-)

UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
 - tworzenia i wywoływania własnych funkcji
 - użycia wbudowanych funkcji, które jako argumenty przyjmują inne funkcje
 - sortowania danych

CELE I ZADANIA:

- Utwórz funkcję, która na podstawie podanego opisu lokalizacji zwróci listę najbliższych stacji pomiarowych znajdujących się w odległości niewiększej od podanej
- Lista powinna być uporządkowana wg odległości (od najbliższych do bardziej oddalonych)

ALGORYTM WYKONANIA:

- Utwórz funkcję, która jako argument przyjmie opis tekstowy lokalizacji, promień oraz listę dostępnych stacji
- Zadaniem funkcji jest zwrócenie listy najbliższych stacji pomiarowych w kolejności od najbliższych do bardziej oddalonych wraz z informacją o odległości
- W tym celu:
 - korzystając z rozwiązania ćwiczenia dodatkowego 1.5 z tego modułu zlokalizuj położenie określone przez podany opis
 - utwórz listę krotek – każda krotka powinna zawierać informację o kolejnej stacji i jej odległość od punktu centralnego
 - do obliczenia odległości pomiędzy kolejną stacją, a punktem centralnym wykorzystaj rozwiązanie ćwiczenia dodatkowego 1.4 z bieżącego modułu
 - utwórz listę krotek zawierających informacje o stacjach oraz ich odległościach, tych, które znajdują się nie dalej niż podany promień
 - uporządkuj te krotki rosnąco wg odległości
- Przetestuj działanie tej funkcji