***Zadania na e-learning***

Lista 1, Lista 2, Lista 3

1. Przypisz do zmiennej dowolną liczbę całkowitą i wypisz ją na ekran.

2. Poproś użytkownika o podanie swojego imienia i wypisz je na ekranie. Przy pisaniu kodu źródłowego wykorzystaj komentarze.

3. Napisz program, który sprawdzi czy podany wiek kwalifikuje osobę do emerytury (czyli czy ma więcej niż 60 lat dla kobiet i więcej niż 65 lat dla mężczyzn).

4. Napisz program, który sprawdzi czy podane trzy liczby mogą być bokami trójkąta.

Ćw. 4, Ćw. 5

1. Stwórz słownik zawierający informacje o kilku studentach (imię, nazwisko, wiek). Następnie wyświetl dane jednego z studentów.

2. Stwórz krotkę zawierającą imiona "Anna", "Jan" i "Maria". Czy jest możliwe dodanie nowego imiona do nowo utworzonej krotki?

3. Posortuj listę liczb [5, 2, 8, 1, 3] rosnąco. Usuń drugi element z listy. Wypisz nowo utworzoną posortowaną listę.

4. Stwórz zagnieżdżoną listę zawierającą informacje o pracownikach firmy (imię, nazwisko, stanowisko, pensja). Napisz program, który wyświetli wszystkich pracowników z pensją powyżej 5000 zł.

5. Stwórz zagnieżdżony słownik zawierający informacje o dwóch osobach. Każda osoba powinna mieć przypisane imię, nazwisko i wiek. Następnie wyświetl imię pierwszej osoby.

6. Stwórz zagnieżdżoną krotkę zawierającą dwie krotki: (1, 2, 3) oraz (4, 5, 6). Następnie wyświetl trzeci element pierwszej krotki.

Cw. 6

1.Napisz program, która przyjmuje listę imion i listę nazwisk oraz zwraca listę zawierającą pełne imiona i nazwiska. Wykorzystaj funkcję zip().

Przykładowe dane wejściowe:

imiona = ['Anna', 'Jan', 'Maria']

nazwiska = ['Kowalska', 'Nowak', 'Wiśniewska']

Oczekiwany wynik:

['Anna Kowalska', 'Jan Nowak', 'Maria Wiśniewska']

2. Utwórz listę zawierającą oceny z egzaminów studentów. Następnie użyj funkcji enumerate() do wyświetlenia indeksów oraz odpowiadających im ocen na ekranie.

3. Stwórz program, który przy użyciu funkcji len() sprawdzi długość słownika zawierającego nazwy przedmiotów i wyświetli tę długość na ekranie.

4. Napisz program w Pythonie, który będzie symulował działanie kolejki za pomocą wbudowanej struktury danych `deque` z modułu `collections`. Program powinien umożliwiać dodawanie elementów do kolejki, usuwanie elementów z kolejki oraz wyświetlanie zawartości kolejki.

Ćw. 7

1. Napisz program w Pythonie, która przyjmuje dwa napisy jako argumenty i zwraca informację czy są one anagramami (czy składają się z tych samych liter).

2. Napisz program, który obliczy pole powierzchni koła o zadanym promieniu, korzystając z modułu math.

3. Napisz program, który będzie generować losową liczbę z przedziału od 1 do 100 i sprawdzi, czy jest ona liczbą pierwszą.

Ćw. 8

1. Napisz funkcję w Python, która przyjmuje jako argument listę napisów i zwraca listę zawierającą tylko te napisy, które zawierają literę 'a'.

2. Napisz funkcję w Python, która przyjmuje jako argument listę liczb całkowitych i zwraca listę zawierającą tylko te liczby, które są kwadratami innych liczb z listy.

3. Stwórz funkcję, która przyjmuje słownik zawierający nazwy produktów i ich ceny, a następnie zwraca nazwę produktu o najwyższej cenie.

Ćw. 9

1. Stwórz klasę "Książka" z atrybutami: tytuł, autor, rok\_wydania. Napisz metodę, która wyświetli informacje o książce oraz metodę, która sprawdzi czy książka została wydana przed 2000 rokiem.

2. Stwórz klasę "Produkt" z atrybutami: nazwa, cena, ilość. Napisz metody, które umożliwią dodanie produktu do koszyka, zmniejszenie ilości produktu w koszyku oraz obliczenie całkowitej wartości koszyka.

3. Stwórz klasę "Trójkąt" dziedziczącą po klasie "FiguraGeometryczna", która będzie miała dodatkowy atrybut długości boków oraz metodę obliczającą pole i obwód trójkąta.

4. Stwórz klasę "Pojazd" z atrybutami marka, model i rok\_produkcji. Następnie stwórz klasy "Samochod" i "Motocykl", które dziedziczą po klasie "Pojazd" i dodają atrybuty specyficzne dla danego typu pojazdu.

Ćw. 10

1. Polimorfizm:

Stwórz dwie klasy: Kwadrat i Kolo. Obie klasy powinny mieć metodę obliczająca pole powierzchni (np. oblicz\_pole()). Następnie stwórz listę obiektów tych klas i wywołaj dla każdego z nich metodę oblicz\_pole().

2. Hermetyzacja:

Stwórz klasę Osoba z atrybutami imie, nazwisko i wiek. Zabezpiecz atrybuty imie i nazwisko przed bezpośrednim dostępem z zewnątrz klasy.

3. Abstrakcja:

Stwórz klasę Zwierze z metodą abstrakcyjną daj\_glos(). Następnie stwórz dwie klasy dziedziczące po klasie Zwierze: Pies i Kot. Każda z tych klas powinna zaimplementować metodę daj\_glos() w sposób odpowiedni dla danego zwierzęcia.

4.

Napisz dekorator, który będzie mnożył wynik funkcji przez 2.