Zad1

Stwórz 3 klasy: Material, Ubrania, Sweter. Klasa: Material

Atrybuty:

* rodzaj,
* długość
* szerokość

Metody:

* konstruktor
* wyświetl\_nazwę

Klasa: Ubrania

Atrybuty:

* rozmiar
* kolor
* dla\_kogo

Metody:

* wyświetl\_dane – do wyświetlania informacji o ubraniu

klasa: Sweter

Atrybuty:

* rodzaj\_swetra – np. Rozpinany, z golfem itd.

Metody:

* wyświetl\_dane

Ubrania dziedziczą po klasie Materiał, a Swetry po klasie Ubrania. Stwórz kilka instancji obiektów i sprawdź, które metody można wykorzystać.

Zad. 2

Przeciąż metodę ``\_\_add\_\_()`` dla klasy Kwadrat, która będzie zwracała instancje klasy Kwadrat o nowym boku, będącym sumą długości boku kwadratu oraz obwodu kwadratu

Zad. 3

Poczytaj o metodach `\_\_ge\_\_, \_\_gt\_\_, \_\_le\_\_, \_\_lt\_\_,` przeciąż je i spróbuj wykorzystać w instrukcji warunkowej do porównania dwóch instancji obiektów klasy Kwadrat.

Zad. 4

Korzystając z powyższego kodu stwórz kilka instancji klasy Point i spróbuj odwołać się do zmiennej counter z poziomu różnych instancji, porównując jej wartość dla każdej z nich oraz spróbuj zmienić jej wartość.

Zad. 5

Za pomocą funkcji `isinstance()` oraz `issubclass()` sprawdź wynik dla instancji obiektu `Pracownik` oraz `Menadzer` dla klas `Osoba, Pracownik i Manadzer`.

Zad. 6

Przetestuj powyższy iterator na kilku różnych kolekcjach.

Zad. 7

Napisz własny iterator, który będzie zwracał tylko elementy z parzystych indeksów przekazanej kolekcji.

Zad. 8

Napisz własny iterator, który będzie zwracał tylko samogłoski z przekazanego ciągu tekstowego. Zaimplementuj sprawdzanie czy przekazany został string jako argument konstruktora tego iteratora (sprawdź funkcję `isinstance()`).

Zad. 9

Przepisz jeden z napisanych przez siebie iteratorów na generator.

Zad. 10

Napisz generator, który będzie zwracał kolejne wartości ciągu arytmetycznego.