

Kolokwium #2- Zestaw 23

Zadania mogą być wykonane jako skrypt lub notatnik jupytera. Rozwiązania należy umieścić:

- opcja 1: (jako kody) w prywatnym repozytorium na Githubie
- opcja 2: spakować jako zip i umieścić na pendrive.

Zadanie A - 20 pkt.

Napisz klasę `Product`. Klasa powinna posiadać prywatne atrybuty instancyjne:

1. `brand` typu `str` (pol. marka)
2. `model` typu `str` (pol. model)
3. `price` typu `float` (pol. cena)

Zaimplementuj inicjalizator z trzema argumentami (zachowaj kolejność i nazwy jak wcześniej). Zadbaj również o to aby inicjalizator w razie podania napisów krótszych niż dwa znaki, ustawiał domyślnie `None` lub pusty napis. Ponadto niezależnie należy sprawdzić, aby cena była dodatnia (w innym przypadku ustawić na 25.99).

Dodaj do klasy właściwości - (getter i setter). Jeśli dla setterów podane wartości argumentów nie spełniają założeń - wyrzuc wyjątek z odpowiednim komunikatem.

Zaimplementuj klasę `Telephone` dziedziczącą po klasie `Product`. Klasa ta powinna posiadać atrybuty instancyjne prywatne:

1. `number` (pol. numer telefonu), typu `str`
2. `diagonal_screen` typu `float` (przekątna ekranu)

Rozszerz w tej klasie inicjalizator. Niezależnie sprawdź aby numer telefonu był w formacie "XXX-XXX-XXX" gdzie X to cyfry. Przekątna ekranu musi być liczbą z przedziału [4, 12].

Zaimplementuj właściwości (setter i getter) dla każdego atrybutu. Jeśli dla setterów podane wartości argumentów nie spełniają założeń, wyrzuc wyjątek z odpowiednim komunikatem.

Poza wspomnianymi wyżej klasami stwórz co najmniej 3 obiekty i przechwyć ew. wyjątki.

Zadanie B - 8 pkt:

1. Stwórz listę zawierającą 10 nazw imion (będących napisami). Następnie za pomocą wyrażeń lambda stwórz listę zawierającą imiona zaczynające się od spółgłoski.

Zadanie C - 8pkt:

1. Napisz funkcję, która przyjmuje listę napisów i zwraca najdłuższy napis. Wyrzuc wyjątek jeśli lista jest pusta lub więcej niż jeden napis jest najdłuższy. Przetestuj działanie funkcji.

Zadanie D - 14 pkt:

1. Stwórz klasę `Smartphone` z prywatnymi atrybutami `brand` oraz `model`. Dodaj w klasie metodę `__init__` z dwoma argumentami ustawiającą z nich odpowiednie atrybuty. Dodaj w klasie właściwości (getter i setter).
2. Stwórz dekorator, który przyjmuje jako parametr listę niedozwolonych wartości i sprawdza, czy argument przekazany do funkcji nie jest na tej liście. Jeśli tak, to zwraca odpowiedni komunikat o błędzie.
3. Uzupełnij metodę `__init__` w klasie `Smartphone` za pomocą dekoratora tak, aby były zabronione marki LG, Huawei. Możesz w razie potrzeby zmodyfikować nieco dekorator.

4. Poza klasą sprawdź działanie ograniczenia i przechwyć ew. wyjątki.