

Ćwiczenia 12

Dekoratory z parametrem

Wprowadzanie:

<https://gist.github.com/pjastr/ecfef5cf3902b04365df38561eeda2c2>

<https://gist.github.com/pjastr/affc4ad138207857932f0fe41803c2a3>

<https://gist.github.com/pjastr/6521f1154e03b54546bd5a0f4724818e>

Zadania:

1. Stwórz dekorator, który przyjmuje jako parametr nazwę argumentu i wartość argumentu i zwraca funkcję, która przypisuje argumentowi wartość domyślną jeśli nie została przekazana.
2. Stwórz dekorator, który przyjmuje jako parametr typ i sprawdza, czy argument przekazany do funkcji jest tego typu. Jeśli nie, to zwraca odpowiedni komunikat o błędzie.
3. Stwórz dekorator, który przyjmuje jako parametr limit i sprawdza, czy ilość wywołań funkcji nie przekroczyła tego limitu. Jeśli tak, to zwraca odpowiedni komunikat o błędzie.
4. Stwórz klasę `Person` z prywatnymi atrybutami `name` i `age`. Za pomocą dekoratora ze wcześniejszego punktu zabezpiecz klasę tak, by była możliwość maksymalnie stworzenia 5 obiektów w danym typie.
5. 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.
6. Stwórz dekorator, który przyjmuje jako parametr słownik z wartościami domyślnymi i aktualizuje/nadpisuje słownik z argumentami przekazanymi do funkcji.
7. Stwórz dekorator, który przyjmuje jako parametr nazwę pliku i zapisuje informacje o wywołaniu funkcji (np. nazwa funkcji, argumenty, wartość zwracana) do tego pliku.
8. Stwórz dekorator, który przyjmuje jako parametr liczbę i wyświetla informacje o wywołaniu funkcji tylko co n-te raz.

Bibliografia:

- <https://www.geeksforgeeks.org/decorators-with-parameters-in-python/>