

## Zadanie 2A: Tworzenie metaklasy

Celem tego zadania jest zaprojektowanie metaklasy, która sprawdza, czy każda klasa, która z niej dziedziczy, zawiera wymagane atrybuty. Przyda się to w sytuacjach, gdy tworzysz klasy bazowe lub interfejsy, które wymagają, by klasy pochodne implementowały określone metody.

### Specyfikacja:

- Metaklasa ma wymagać, aby każda klasa pochodna implementowała atrybuty (np. `name` i `execute`).
- Jeśli klasa nie implementuje wymaganego atrybutu, ma zostać zgłoszony wyjątek `TypeError` podczas tworzenia klasy.

### Zadanie:

1. Utwórz metaklasę o nazwie `RequiredAttributesMetaclass`, która sprawdza, czy każda klasa pochodna zawiera atrybuty `name` i `execute`.
2. Napisz dwie klasy:
  - `MyCommand` – która poprawnie implementuje te atrybuty.
  - `IncompleteCommand` – która nie implementuje atrybutu `execute` (ta klasa powinna zgłosić błąd podczas próby jej zdefiniowania).

W ten sposób upewnisz się, że każda klasa pochodna przestrzega kontraktu określonego przez klasę bazową.

## Zadanie 2B: Praktyczne zastosowanie deskryptora z metodami `get`, `set` i `delete`

Celem tego zadania jest stworzenie deskryptora, który zarządza dostępem do atrybutu w klasie. Deskryptor będzie automatycznie konwertować wartość przypisywaną do atrybutu na małe litery, a także logować operacje pobierania, ustawiania i usuwania wartości. Przykład zastosowania może dotyczyć zarządzania nazwą użytkownika w systemie.

### Specyfikacja:

- Utwórz deskryptor o nazwie `LowercaseString`, który:
  - Używa metody `__get__` do zwracania wartości atrybutu.
  - Używa metody `__set__` do ustawiania wartości atrybutu, konwertując je na małe litery.
  - Używa metody `__delete__` do usuwania wartości atrybutu.
- Klasa `User` powinna używać tego deskryptora do przechowywania i zarządzania nazwą użytkownika.

### Zadanie:

1. Utwórz deskryptor `LowercaseString` z metodami `__get__`, `__set__` i `__delete__`.
2. Zaimplementuj klasę `User`, która wykorzystuje deskryptor do zarządzania atrybutem `username`.

3. Przetestuj funkcjonalność deskryptora, wykonując następujące operacje:

- Ustaw wartość username na dowolny ciąg znaków.
- Pobierz i wyświetl wartość username.
- Usuń atrybut username i sprawdź, czy został poprawnie usunięty (spróbuj go pobrać).

W ten sposób poznasz praktyczne zastosowanie deskryptorów do zarządzania atrybutami w klasach.