

## Zadanie 1: Stwórz dekorator z wyborem operacji

Stwórz dekorator w Pythonie, który będzie umożliwiał funkcji wybór jednej z czterech operacji matematycznych: dodawanie, odejmowanie, mnożenie lub dzielenie. Dekorator powinien przyjmować nazwę operacji jako argument i zmieniać sposób działania funkcji, aby wykonywała wybraną operację na dwóch przekazanych do niej argumentach.

### Szczegóły zadania:

- Dekorator powinien obsługiwać cztery operacje: dodawanie ('dodawanie'), odejmowanie ('odejmowanie'), mnożenie ('mnożenie') i dzielenie ('dzielenie').
- Funkcja, na którą zostanie nałożony dekorator, powinna przyjmować dwa argumenty (np. liczby) i zwracać wynik wybranej operacji.
- Upewnij się, że obsłużysz przypadek dzielenia przez zero, zwracając odpowiedni komunikat o błędzie.

**Przykład użycia:** Jeśli dekorator zostanie nałożony na funkcję z wyborem operacji 'mnożenie', wywołanie funkcji z argumentami 5 i 3 powinno zwrócić wynik 15.

## Zadanie 2: Zaprojektuj klasy opisujące zawody biegowe w górach

Zaprojektuj zestaw klas w Pythonie, które opisują różne rodzaje zawodów biegowych w górach, stosując przy tym abstrakcję. Stwórz klasę abstrakcyjną `Zawody`, która będzie definiowała podstawowe atrybuty i metodę, którą klasy dziedziczące będą musiały zaimplementować. Następnie stwórz klasy dziedziczące po niej, które będą opisywać szczegółowe typy zawodów.

### Szczegóły zadania:

- Utwórz klasę abstrakcyjną `Zawody`, która będzie miała atrybuty `nazwa` i `dystans`.
- Klasa abstrakcyjna powinna zawierać metodę `opis_zawodow()`, którą zaimplementują klasy dziedziczące.
- Stwórz klasy dziedziczące:
  - **UltraMaraton:** Dodaj atrybut `przewyższenie`, który opisuje całkowite wzniesienie na trasie.
  - **TrailRun:** Dodaj atrybut `typ_terenu`, który opisuje rodzaj powierzchni trasy.
  - **VerticalRun:** Dodaj atrybut `kąt_wzniesienia`, który reprezentuje nachylenie trasy.
- Zaimplementuj metodę `opis_zawodow()` w każdej z klas dziedziczących, która zwróci szczegółowy opis zawodów.

**Przykład użycia:** Utwórz obiekty każdej z klas i wywołaj metodę `opis_zawodow()`, aby sprawdzić poprawność implementacji.