



Instrukcja CASE w Pythonie

M@я3k Pүđ€£kØ

Programowanie w Pythonie

Spis treści

- Pattern matching
- Przykład instrukcji MATCH/CASE
- Łączenie warunków w jeden wzorzec
- Strażnicy

Pattern matching

- Instrukcję MATCH/CASE stosuje się przy wyborze wśród wielu elementów lub operacji w programie.
- Jako argument przyjmuje ona wyrażenie i porównuje je ze zdefiniowanymi wzorcami (każdy wzorzec następuje po słowie kluczowym case).
 - Jeśli wyrażenie jest zgodne ze zdefiniowanym wzorcem, wykonywane są związane z nim instrukcje.

```
match wyrażenie
  case przypadek_1:
    operacje_1
  case przypadek_2:
    operacje_2
  case _:
    operacje_pozostałe
```

Przykład instrukcji MATCH/CASE

```
x = int(input("Podaj liczbę "))
match x:
    case 1:
        print("jedynka")
    case 2:
        print("dwójka")
    case 3:
        print("trójka")
    case _:
        print("Inna liczba")
```

Opis polecenia

- Sprawdzanie wartości następuje kolejno z góry na dół, co warunek zostanie zidentyfikowany, to cały block match zostanie opuszczony natychmiast po wykonaniu zdefiniowanej operacji.
- Jeśli żaden z bloków case nie pasuje, żadna z gałęzi nie jest wykonywana.
- Domyślny przypadek (w niektórych językach określa go słowo kluczowe default), do którego wpadają wartości nie pasujące nigdzie indziej, jest określany przez symbol _.
- Oprócz prostego porównywania wartości, możemy też dokonywać dokładniejszego uwarunkowania obiektów w celu bardziej szczegółowego dopasowania.

Łączenie warunków w jeden wzorzec

- Możliwe jest połączenie kilku literałów w jeden wzorzec przy użyciu operatora | („or”).

```
case 1:  
    print("Kiepska ocena")
```

```
case 2:  
    print("Kiepska ocena")
```

```
case 3:  
    print("Kiepska ocena")
```

da się zastąpić jedną linijką łączącą te warunki:

```
case 1|2|3 :  
    print("Kiepska ocena")
```

Straznicy (Guards)

- Instrukcja MATCH/CASE obsługuje "strażników" (ang. guards)
- Pozwala to na określenie dodatkowych warunków, które muszą być spełnione, aby blok kodu w CASE został wykonany.
- Realizuje się to poprzez dodanie IF po słowie kluczowym CASE.
- Wzór jest dopasowywany, a następnie sprawdzany jest dodatkowy warunek logiczny podany w if.
 - Operacje wykonują się tylko, gdy obie te rzeczy są prawdziwe

```
match wyrażenie
```

```
    case przypadek 1 if warunek_1:
```

```
        operacje1
```

```
    case przypadek 2 if warunek_2:
```

```
        operacje2
```

```
    case _:
```

```
        operacje_pozostałe
```

Przykład użycia strażnika

```
liczba = 3
```

```
match liczba:
```

```
    case liczba    if liczba < 0:
```

```
        print("Liczba ujemna")
```

```
    case 0:
```

```
        print("Liczba zero")
```

```
    case liczba    if liczba > 0:
```

```
        print("Liczba dodatnia")
```

Ćwiczenie

1. Wczytaj numer dnia (1–7) i za pomocą MATCH/CASE wypisz jego nazwę (np. 1 → poniedziałek, 2 → wtorek itd.)
2. Użytkownik podaje numer miesiąca (1–12). Program wypisuje nazwę miesiąca.
3. Wczytaj ocenę (1–6) i wypisz jej opis
 - 1 – niedostateczny, 2 – dopuszczający, 3 – dostateczny, 4 – dobry, 5 – bardzo dobry, 6 – celujący
4. Wczytaj nazwę dnia tygodnia i wypisz, czy to dzień roboczy czy weekend. Wykorzystaj kilka przypadków w jednym CASE.
5. Wczytaj nazwę miesiąca i wypisz jaka to jest pora roku. Wykorzystaj kilka przypadków w jednym CASE.
6. Wczytaj dwa liczby i znak działania (+, -, *, **, /, //, %). Użyj MATCH/CASE, aby wykonać odpowiednie działanie matematyczne. Dla wyjątków użyj strażników.
7. Wyświetl menu z wyborem opcji (np. 1 – dodaj użytkownika, 2 – usuń, 3 – pokaż listę, 0 – zakończ). Obsłuż każdą opcję przy pomocy MATCH/CASE.

Powtórzenie

1. Jak uzyskać w Pythonie możliwość wyboru wśród wielu odpowiedzi?
2. Opisz działanie instrukcji MATCH/CASE ?
3. W jaki sposób połączyć kilka warunków w jeden wzorzec?
4. Jakie możliwości daje zastosowanie strażników?