INSTRUKCJA LABORATORYJNA / 45 MINUT / PRACA INDYWIDUALNA

Switch/case w Pythonie

Wybór jednej z opcji za pomocą **match...case**

Jak to się nazywa w Pythonie?

W Pythonie używamy słów `match` i `case`. To mechanizm dopasowania wzorców. Na tej lekcji wykorzystasz go do wyboru jednej z kilku wartości, podobnie jak switch/case w innych językach. **Potrzebny jest Python 3.10 lub nowszy.**

Imię i nazwisko: ..... Klasa: ..... Data: .....

Po lekcji potrafisz

- zbudować menu i obsłużyć wybór użytkownika;
- zastosować `case _` oraz połączyć wartości znakiem `|`;
- sprawdzić program dla poprawnych i nieznanych opcji.

Plan pracy - łącznie 45 minut

Minuty lekcji	Co robisz?	Czas
0-5	Przygotowanie środowiska i cel ćwiczenia	5 min
5-10	Poznanie składni na krótkim przykładzie	5 min
10-23	Biletomat: napisanie, testy i rozbudowa	13 min
23-28	Wiele wartości w jednym case	5 min
28-43	Dwa zadania samodzielne	15 min
43-45	Sprawdzenie i pokazanie wyników	2 min

Start | 0-5 min

1. Otwórz szkolny edytor Pythona. W jego konsoli sprawdź wersję:

Konsola Pythona

```
import sys
print(sys.version)
```

2. Jeżeli masz wersję 3.9 lub starszą, zgłoś to nauczycielowi. Utwórz folder `Nazwisko_match_case`. Programy zapisuj w nim jako pliki `.py` i uruchamiaj przyciskiem Run/Uruchom w edytorze.

Przyda się: znajomość zmiennych, `input()`, `print()` i podstawowych działań matematycznych. Programy w tej instrukcji nie wymagają dodatkowych bibliotek.

ETAP 1 / 5-10 MIN / 5 MINUT

Poznaj match...case

Przepisz przykład do pliku `skladnia.py` i uruchom go.

skladnia.py

```
opcja = input("Wpisz A lub B: ")

match opcja:
    case "A":
        print("Wybrano opcję A")
    case "B":
        print("Wybrano opcję B")
    case _:
        print("Nie ma takiej opcji")

print("Koniec programu")
```

Jak czytać ten program?

Fragment	Znaczenie
`input(...)`	Pobiera tekst z klawiatury i zapisuje go w zmiennej opcja.
`match opcja:`	Rozpoczyna dopasowywanie wartości zmiennej do kolejnych wzorców.
`case "A":`	Ten blok wykona się, jeśli wpisany tekst to dokładnie A.
`case _:`	Obsługuje każdy pozostały wybór. Umieszczamy go na końcu.
Wcięcia	Stosuj 4 spacje przed case i 8 spacji przed jego instrukcjami.

Wykonuje się pierwszy pasujący blok

Po jego wykonaniu program przechodzi za cały `match`. **Nie dopisuj** `break`**.** Ostatni `print` jest poza blokiem i wyświetli napis „Koniec programu” niezależnie od wyboru.

Przewidź, potem sprawdź

Uruchom program trzy razy. Wpisuj kolejno `A`, `B` i `a`. Przed naciśnięciem Enter powiedz, który komunikat się pojawi. **Wielkość liter ma znaczenie.** W konsoli nie dopisuj cudzysłowów.

Uwaga: tekst „1” i liczba 1 to różne wartości

`input()` zwraca tekst, więc wybór numeru z menu obsługuj przez `case "1":`. Zapis `case 1:` służy do dopasowania liczby, np. po zamianie wejścia funkcją `int()`.

Blok `case _` jest opcjonalny. Jeśli go pominiesz i nic nie pasuje, żaden blok case się nie wykona, a program będzie działał dalej.

ETAP 2 / 10-23 MIN / 13 MINUT

Zbuduj biletomat

Program ma wyświetlać cenę biletu wybranego przez użytkownika.

Krok 1. Utwórz plik `biletomat.py`. Przepisz kod, zachowując dwukropki, cudzysłowy i wcięcia. Odczytaj na głos jeden wybrany przypadek.

biletomat.py

```
print("BILETOMAT - SZKOLNE KINO")
print("1 - bilet normalny: 20 zł")
print("2 - bilet ulgowy: 12 zł")
print("3 - bilet rodzinny: 45 zł")

wybor = input("Wybierz bilet (1-3): ")

match wybor:
    case "1":
        print("Bilet normalny. Do zapłaty: 20 zł")
    case "2":
        print("Bilet ulgowy. Do zapłaty: 12 zł")
    case "3":
        print("Bilet rodzinny. Do zapłaty: 45 zł")
    case _:
        print("Nie ma takiego biletu")
```

Krok 2. Uruchamiaj program osobno dla każdego testu. Porównaj komunikaty. Program przyjmuje jeden wybór na uruchomienie.

Wpisz	Oczekiwany wynik	Zgodny?
1	Bilet normalny. Do zapłaty: 20 zł	☐
2	Bilet ulgowy. Do zapłaty: 12 zł	☐
3	Bilet rodzinny. Do zapłaty: 45 zł	☐
9	Nie ma takiego biletu	☐

Krok 3. Dodaj opcję `4`: bilet VIP za 60 zł. Dopisz informację w menu oraz nowy blok case **przed** `case _`. Zmień zakres w pytaniu na 1-4. Sprawdź wybór 4 i niepoprawny wybór 0.

Zatrzymaj się na chwilę

Czy samo dopisanie pozycji w menu zmienia działanie `match`? Wyjaśnij, po co potrzebne są obie zmiany.

ETAP 3 / 23-28 MIN / 5 MINUT

Kilka wartości, jedna reakcja

Rozbuduj biletomat: bilet wybierzesz numerem albo nazwą.

Zostaw menu. W pliku `biletomat.py` zastąp pobieranie wyboru i cały blok match poniższym fragmentem:

Nowa wersja fragmentu biletomat.py

```
wybor = input("Numer lub nazwa biletu: ").strip().lower()

match wybor:
    case "1" | "normalny":
        print("Bilet normalny. Do zapłaty: 20 zł")
    case "2" | "ulgowy":
        print("Bilet ulgowy. Do zapłaty: 12 zł")
    case "3" | "rodzinny":
        print("Bilet rodzinny. Do zapłaty: 45 zł")
    case "4" | "vip":
        print("Bilet VIP. Do zapłaty: 60 zł")
    case _:
        print("Nie ma takiego biletu")
```

Znak `|` oznacza alternatywę: `"1"` **lub** `"normalny"`. W tym miejscu używamy `|`, a nie słowa `or`. `.strip()` usuwa spacje na początku i końcu tekstu. `.lower()` zamienia litery na małe.

Sprawdź: wpisz `VIP` - oczekujesz ceny 60 zł. Następnie wpisz `normalny` ze spacjami po obu stronach - oczekujesz ceny 20 zł. Dla `xyz` nadal ma pojawić się komunikat o braku biletu.

Gdy coś nie działa - krótka pomoc

Problem	Co sprawdzić?
Błąd przy match	Wersję Pythona (minimum 3.10), dwukropek oraz wcięcia.
Wybór 1 trafia do case _	Dla input() użyj `case "1":`, z cudzysłowami.
`case normalny:`	To przechwycenie wartości do zmiennej, a nie porównanie tekstu. Użyj `case "normalny":`.
Błąd po przeniesieniu case _	Samodzielne `case _:` pasuje do wszystkiego. Musi być ostatnie.

Teraz praca samodzielna: wykonaj zadania z kolejnych dwóch stron. Możesz korzystać ze składni i przykładów z instrukcji.

Podstawa składni: oficjalna dokumentacja Python, 4.7. match Statements; nowości Python 3.10, PEP 634.

ZADANIE SAMODZIELNE 1 / 28-35 MIN / 7 MINUT

Koszt dostawy

Plik: `dostawa.py` | Poziom podstawowy

Napisz program sklepu internetowego, który zapyta o sposób dostawy i wyświetli jej koszt. Klient może wpisać **numer albo nazwę**.

Akceptowany wybór	Komunikat programu
1 lub `kurier`	Koszt dostawy: 15 zł
2 lub `paczkomat`	Koszt dostawy: 10 zł
3 lub `odbior`	Koszt dostawy: 0 zł
Każdy inny wybór	Nieznany sposób dostawy

Wymagania

1. Wyświetl czytelne menu z trzema możliwościami i ich cenami.
2. Pobierz wybór. Obsłuż wielkie litery i spacje na brzegach tekstu.
3. Zastosuj `match...case` i połącz numer z nazwą za pomocą `|`.
4. Dodaj `case _` dla nieznanych opcji.

Używaj nazwy `odbior` bez polskich znaków. Menu i komunikaty możesz wyświetlać z polskimi znakami.

Testy programu

Każdą wartość sprawdź w osobnym uruchomieniu. Zaznacz wynik po wykonaniu testu.

Wpisz w konsoli	Oczekiwany rezultat	Działa?
1	Koszt dostawy: 15 zł	☐
`kurier`	Koszt dostawy: 15 zł	☐
2	Koszt dostawy: 10 zł	☐
PACZKOMAT ze spacjami przed i po nazwie	Koszt dostawy: 10 zł	☐
3, potem `odbior`	W obu testach: Koszt dostawy: 0 zł	☐
9	Nieznany sposób dostawy	☐

Przed przejściem do zadania 2

☐ W każdym case połączyłem numer i nazwę znakiem |.

☐ Przetestowałem również nieznaną opcję.

☐ Zapisałem działającą wersję pliku dostawa.py.

ZADANIE SAMODZIELNE 2 / 35-43 MIN / 8 MINUT

Mini kalkulator

Plik: `kalkulator.py` | Wybór działania

Napisz program, który pobierze dwie liczby oraz znak działania. Za pomocą `match...case` wybierze obliczenie i wyświetli wynik.

Wymagania

1. Pobierz liczby do zmiennych `a` i `b`. Użyj `float(input(...))`, aby móc wykonywać obliczenia.
2. Zapytaj o działanie: `+`, `-` lub `*`.
3. W odpowiednim case oblicz i wypisz wynik. Dla innego znaku wyświetl „Nieznane działanie”.

Przypomnienie o liczbach

Zakładamy, że użytkownik wpisuje poprawne liczby. Ułamki dziesiętne zapisuj z kropką, np. `2.5`. Samego znaku działania nie zamieniaj na liczbę. Zapisuj przypadki jako tekst, np. `case "+":`.

Testy: za każdym razem podaj a = 8 i b = 2

Znak działania	Oczekiwany rezultat	Działa?
+	Wynik: 10.0	☐
-	Wynik: 6.0	☐
*	Wynik: 16.0	☐
?	Nieznane działanie	☐

Dopuszczalny jest też zapis wyniku bez części .0. Dodatkowo sprawdź: a = 2.5, b = 1.5, działanie +. Wynik powinien wynosić 4.0.

Dla szybszych: bezpieczne dzielenie

Dodaj działanie `/`. W jego bloku użyj `if...else`, aby sprawdzić, czy `b == 0`. Dla zera wypisz „Nie można dzielić przez zero”; dla innej wartości oblicz iloraz. Przetestuj 8 / 2 oraz 8 / 0. **To zadanie dodatkowe, jeśli zostanie czas.**

Ostatnie 2 minuty | 43-45 min

☐ Zapisałem pliki: skladnia.py, biletomat.py, dostawa.py i kalkulator.py.

☐ W zadaniach samodzielnych sprawdziłem poprawny i błędny wybór.

☐ Pokazałem nauczycielowi działanie dostawa.py oraz kalkulator.py.

Na wyjście: wyjaśnij w jednym zdaniu, do czego służy `case _`, oraz dlaczego po zwykłym `input()` stosujesz `case "1":`, a nie `case 1:`.