Nazwa implementacji: Nauka języka Python – wyrażenia warunkowe
Autor: Piotr Fiorek
Opis implementacji: Poznanie wyrażeń warunkowych if – elif - else.

Nasz kalkulator umie już liczyć, ale potrafi przeprowadzać tylko jedno, wybrane działanie i aby np. podzielić dwie liczby, musimy zmieniać kod programu. A gdyby tak nasz program na początku pytał, jaką operację chcemy wykonać i na tej podstawie wybierał, jakie działanie wykonać? W tym celu program będzie musiał na początku zapytać o działanie, które chcemy wykonać, zapamiętać w zmiennej co użytkownik wpisał i na tej podstawie zdecydować co ma zrobić dalej.

Do podejmowania decyzji w programowaniu służy instrukcja warunkowa **„if”,** czyli z angielskiego **„jeśli”**. Posiada ona również opcjonalną, dodatkową część w postaci instrukcji **„else”**, czyli **„w przeciwnym wypadku”**. Część **„else”** nie jest obowiązkowa, ale bardzo często jest przydatna, jeśli chcemy, żeby nasz program sprawdził jakiś warunek i wykonał jakiś kod, jeśli warunek jest prawdziwy lub wykonał inny kawałek kodu, jeśli warunek był fałszywy.

Użycie pary „if” – „else” wygląda w Pythonie tak:

```
if <wyrażenie warunkowe>:

instrukcja 1

instrukcja 2

...

else:

instrukcja 1

instrukcja 2

...
```

Oczywiście zamiast części „<wyrażenie warunkowe>” wpisujemy, to co chcemy, aby nasz program przetestował. Po części wyrażeniu warunkowym musimy wpisać dwukropek, co oznacza, że dalej wypiszemy instrukcje, które mają być wykonane, jeśli warunek jest prawdziwy. Instrukcji może być dowolna ilość, ale wszystkie instrukcje muszą być wcięte względem instrukcji „if”. W ten sposób Python rozpoznaje które instrukcje ma wykonać po sprawdzeniu prawdziwości wyrażenia. Tak samo po instrukcji „else” musimy wstawić dwukropek, a instrukcje muszą być wcięte. Głębokość wcięcia nie ma znaczenia (dobry zwyczaj programowania w Pythonie mówi, żeby używać czterech spacji), ale musi być ono w całym programie zawsze tej samej głębokości. Czyli jeśli raz się zdecydujemy na cztery spacje, to wszystkie wcięcia w programie muszą być robione czterema spacjami.

Zostawmy na razie nasz kalkulator i najpierw pobawmy się instrukcjami „if” – „else” na prostszym kodzie:

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

```
zmienna = input('Podaj cyfrę: ')

zmienna = int(zmienna)

if zmienna > 10:

print('Wpisałeś cyfrę większą niż dziesięć')

else:

print('Wpisałeś cyfrę mniejszą niż dziesięć')

print('Koniec programu')
```

W kodzie po raz kolejny wczytujemy wartość z klawiatury do zmiennej. Następnie przypisujemy do naszej zmiennej wartość tej zmiennej zamienioną na liczbę całkowitą. W ten sposób można na stałe skonwertować zmienną na określony typ, aby nie musieć tego robić za każdym razem, kiedy chcemy użyć zmiennej.

W następnej linii widać użycie instrukcji „if”. W tym przykładzie sprawdzamy czy wartość naszej zmiennej jest większa niż 10. Jeśli jest, wykonywana jest instrukcja zaraz po linii z instrukcją „if”, jeśli natomiast zmienna ma wartość mniejszą niż 10, wykonana zostanie instrukcja wpisana po linii z „else”.

Możecie w ramach eksperymentu usunąć z kodu instrukcję „else” i występującą po niej linijkę, uruchomić program, wpisać cyfrę mniejszą niż 10 i zobaczyć co się stanie.

A co gdybyśmy chcieli, żeby nasz program sprawdzał czy zmienna jest dodatnia, ujemna lub równa zero? W tym celu możemy połączyć ze sobą kilka bloków „if” – „**else**”:

```
zmienna = input('Podaj cyfrę: ')

zmienna = int(zmienna)

if zmienna > 0:

print('Wpisałeś cyfrę dodatnią')

elif zmienna == 0:

print('Wpisałeś zero')

else:

print('Wpisałeś cyfrę ujemną')

print('Koniec programu')
```


KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
FWIOO
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Całość programu nie powinna już być trudna do zrozumienia z wyjątkiem nowej instrukcji „elif”. Jest to sklejenie ze sobą słów „else if”. W ten sposób nasz program wie, że ma sprawdzić czy „zmienna” jest większa od zera i jeśli jest wyświetlić tekst, w przeciwnym wypadku sprawdzić czy jest równa zero i jeśli jest wyświetlić tekst, a w przeciwnym wypadku wykonać ostatni blok kodu. W ten sposób możemy łączyć ze sobą dowolną ilość warunków.

Jak można zauważyć z działania programu Python przestaje sprawdzać kolejne możliwości, po pierwszej która okaże się prawdziwa.

Jak również widać porównanie w Pythonie, wykonujemy poprzez użycie dwóch znaków „=”. Matematyczne wyrażenie „nie równe” (≠) w Pythonie zapisujemy jako „!=”.

Mamy już w tej chwili wszystkie potrzebne narzędzia i możemy wrócić do naszego kalkulatora. Zmodyfikowany kod może wyglądać np. tak:

```
print('Jaką operację chcesz wykonać?\n', '1) dodawanie\n', '2) odejmowanie\n', '3) mnożenie\n', '4) dzielenie\n', '5) potęgowa

wybor = int(input('Wpisz cyfrę operacji: '))

zmienna1 = float(input('Podaj 1 liczną: '))

zmienna2 = float(input('Podaj 2 liczną: '))

if wybor == 1:

wynik = zmienna1 + zmienna2

elif wybor == 2:

wynik = zmienna1 - zmienna2

elif wybor == 3:

wynik = zmienna1 * zmienna2

elif wybor == 4:

wynik = zmienna1 / zmienna2

elif wybor == 5:

wynik = zmienna1 ** zmienna2

else:

print('\nWybrałeś nieistniejącą opcję!\n')

exit()

print('Wynik: ', wynik)
```


KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
FWIOO
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Co nowego jest w tym programie? Funkcja „print” ma podane więcej parametrów niż do tej pory, ale jak mówiłem na początku, może ona przyjmować dowolną ilość parametrów do wyświetlenia. Ale jeśli się uważnie przyjrzycie tekstom w parametrach funkcji „

print

” zauważycie tajemnicze kombinacje „\n” na końcu napisów. Jest to oznaczenie mówiące komputerowi, że w tym miejscu należy wstawić nową linię. Same znaki „

\n

” nie są wypisywane.

Jeszcze jedną zmianą względem dotychczasowych programów jest sposób konwersji danych wpisywanych z klawiatury. W tym programie konwertujemy na typ „float” od razu to, co zwraca funkcja „input” i dopiero tak zmienioną wartość przypisujemy do zmiennej.

W bloku „else” widać jeszcze nową funkcję „exit”, która nie musi przyjmować żadnych parametrów i która powoduje wyjście z programu. W tym wypadku chcemy wyjść z programu, ponieważ użytkownik wpisał cyfrę nie przypisaną do żadnej operacji.

Bloki „if” – „else” można jeszcze zagnieżdżać wewnątrz siebie, aby przeprowadzać dalsze sprawdzenia. W naszym kalkulatorze możemy to wykorzystać np. do sprawdzania czy użytkownik nie próbuje dzielić przez zero:

```
print('Jaką operację chcesz wykonać?\n', '1) dodawanie\n', '2) odejmowanie\n', '3) mnożenie\n', '4) dzielenie\n', '5) potęgowa

wybor = int(input('Wpisz cyfrę operacji: '))

zmienna1 = float(input('Podaj 1 liczną: '))

zmienna2 = float(input('Podaj 2 liczną: '))

if wybor == 1:

wynik = zmienna1 + zmienna2

elif wybor == 2:

wynik = zmienna1 - zmienna2

elif wybor == 3:

wynik = zmienna1 * zmienna2

elif wybor == 4:

if zmienna2 == 0:

print('\nDzielenie przez zero jest zabronione!\n')

exit()
```


KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

```
else:

wynik = zmienna1 / zmienna2

elif wybor == 5:

wynik = zmienna1 ** zmienna2

else:

print('\nWybrałeś nieistniejącą opcję!\n')

exit()

print('Wynik: ', wynik)
```

W tym przypadku można ewentualnie pominąć instrukcję „

else

”, ponieważ jeśli „**zmienna2**” jest równa zero to dzielenie i tak nie zostanie wykonane, bo najpierw wyjdziemy z programu z powodu funkcji „exit”. Czyli część z dzieleniem można by zapisać jako:

```
elif wybor == 4:

if zmienna2 == 0:

print('\nDzielenie przez zero jest zabronione!\n')

exit()

wynik = zmienna1 / zmienna2
```

Warto jeszcze na koniec wspomnieć, że w instrukcję „if” można wpisać więcej niż jeden warunek. W takim przypadku wszystkie wpisane warunki należy ze sobą połączyć słowami „and” lub „or” oznaczającymi logiczne działania, czyli odpowiednio koniunkcję i alternatywę.

Zadania:

- Napisz program, który sprawdza czy wpisana liczba jest liczbą parzystą i wypisująca informację czy jest czy nie.
- Wczytaj z klawiatury trzy zmienne i wypisz je po kolei od największej do najmniejszej (bonus: program powinien uwzględniać możliwość, że liczby są równe).
- Napisz program wczytujący trzy zmienne i sprawdzające czy z odcinków o takich długościach da się zbudować trójkąt.

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.