Nazwa implementacji: Nauka języka Python – pętla while **Autor:** Piotr Fiorek **Opis implementacji:** Poznanie pętli while()

Pisząc programy, bardzo często zdarzy się, że będziemy chcieli wykonać jakieś zadanie więcej niż jeden raz. Na przykład w naszym kalkulatorze możemy chcieć policzyć więcej niż jedną rzecz, a uruchamianie programu za każdym razem na nowo jest podejściem mało programistycznym.

Poznawanie pętli, bo to właśnie one służą do powtarzania pewnych czynności, zaczniemy od petli „while", czyli „dopóki". Pętla ta tak samo jak instrukcja „if" sprawdza jakiś warunek oraz ma podane instrukcje do wykonania. Można w związku z tym zgadnąć, że pętla „while" wykonuje podane instrukcje, dopóki warunek jest prawdziwy. Czyli sprawdza warunek, wykonuje instrukcje, znowu sprawdza warunek, znowu wykonuje instrukcje i robi to tak długo, dopóki warunek jest prawdziwy. Jeśli warunek będzie fałszywy instrukcje nie zostaną wykonane, a program przejdzie do dalszej części. Oznacza to, że w takim razie warunek powinien zależeć od jakiejś zmiennej, której wartość jest zmieniana albo przez sam program, albo przez nas w taki sposób, aby w pewnym momencie warunek przestał być prawdziwy i żebyśmy kiedyś jednak wyszli z programu. Najprostszy przykład programu z pętlą „while" może wyglądać np. tak:

```
x = 5

while x > 0:

print(x)

x = x – 1
```

Na początku tworzymy zmienną, która będzie sterowała naszą pętlą. Pętla będzie się wykonywała tak długo, jak wartość zmiennej będzie większa niż 0. W treści pętli wyświetlamy wartość zmiennej i co najważniejsze w tym przykładzie zmniejszamy wartość zmiennej o 1. Ostatnie instrukcja jest najważniejsza, ponieważ bez niej wartość zmiennej „x" by się nie zmieniała, a bez tego pętla nigdy się nie skończy. Zatem zmodyfikujmy teraz nasz kalkulator tak, aby po wykonaniu obliczenia pytał użytkownika czy chce wykonać następne i zależnie od tego wychodził z programu, albo pytał o kolejną operację i kolejne cyfry:

```
wyjscie = False

while wyjscie == False:

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 liczne: '))

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

if wybor == 1:

wynik = zmienna1 + zmienna2

elif wybor == 2:

wynik = zmienna1 - zmienna2

elif wybor == 3:
```

1

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.

```
wynik = zmienna1 * zmienna2

elif wybor == 4:

if zmienna2 == 0:

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

exit()

else:

wynik = zmienna1 / zmienna2

elif wybor == 5:

wynik = zmienna1 ** zmienna2

else:

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

exit()

print('Wynik: ', wynik)

pytanie = input('\nWyjść z programu? (t/n): ')

if pytanie == 't':

wyjscie = True

print('Koniec programu')
```

W pierwszej linii deklarujemy nową zmienną pod nazwą „wyjscie”. Jest to zmienna nowego typu, o którym nie mówiłem wcześniej. Takie zmienne nazywamy „bool”. Zmienne tego typu przyjmują tylko dwie wartości – „True” oraz „**False**”, czyli odpowiednio prawda i fałsz. Nazwa typu pochodzi od nazwiska brytyjskiego matematyka George'a Boole'a który ma wielkie zasługi w dziedzinie logiki matematycznej, a z tej właśnie dziedziny wywodzą się wartości prawda oraz fałsz.

Druga linia zawiera warunek pętli i oznacza dokładnie tyle, co „dopóki wartość zmiennej wyjscie jest równa fałsz powtarzaj”. A powtarzać ma wszystko, co jest po dwukropku i jest wcięte względem instrukcji „while”.

Dalej kod jest taki sam jak wcześniej z wyjątkiem czterech ostatnich linijek, które zostały dodane. Trzy pierwsze są w bloku z instrukcjami pętli „while”, a ostatnia jest wykonywana już po wyjściu z pętli, czyli kiedy warunek przestanie być prawdziwy. Ostatniej nie trzeba raczej objaśniać. Pozostałe trzy też powinny być jasne, ale wyjaśnijmy, jaka jest och rola. Pierwsza wczytuje do zmiennej „pytanie” wartość podaną przez użytkownika. „if” sprawdza czy użytkownik wpisał literę „t” (tym razem nie konwertowaliśmy wartości wczytanej z klawiatury, bo chcieliśmy porównać ją z innym napisem, czyli w tym przypadku pojedynczą literą) i jeśli tak to zmienia wartość zmiennej „wyjście” tak, aby przerwać pętlę.

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.

W tym kodzie brakuje jeszcze kontroli tego, jakie litery użytkownik wpisuje na końcu. Co prawda wpisanie litery „t” spowoduje wyjście, a dowolnej innej dalsze działanie programu co jest akceptowalne, ale dla elegancji dodajmy jeszcze kontrolę wpisywanych liter tak, aby tylko litery „t” i „n” były poprawne, a przy wszystkich innych żeby pytanie było zadawane jeszcze raz:

```
wyjscie = False

while wyjscie == False:

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()

else:

wynik = zmienna1 / zmienna2

elif wybor == 5:

wynik = zmienna1 ** zmienna2

else:

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

exit()

print('Wynik: ', wynik)

while True:
```


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.

```
pytanie = input('Wyjść z programu? (t/n): ')

if pytanie == 't':

wyjscie = True

break

elif pytanie == 'n':

break

else:

print('Podałeś błędną literę!')

print('Koniec programu')
```

Jak widać, zmianie uległa tylko część z zadawaniem pytania. Zadawanie pytania zostało całe wzięte w pętlę, ale pętlę nietypową. Warunkiem tej pętli jest po prostu wartość „prawda”. Efekt tego jest taki, że warunek zawsze jest prawdziwy, bo prawda zawsze jest prawdziwa. W związku z tym pętla nie posiada końca. Ze względu na to takie pętle nazywamy pętlami nieskończonymi. Ale przecież my chcemy tylko poznać odpowiedź na pytanie, udzieloną w formie litery „t” lub „n” i jednak wyjść z pętli. Do tego służy instrukcja sterująca pętlą – „break”. Instrukcja ta mówi, żeby natychmiast przerwać wykonanie pierwszej pętli, na którą się natrafi, idąc od „break” w górę. Jest to o tyle ważne, że nasza instrukcja „while True” już przecież znajduje się w pętli którą zaczynamy na samym początku programu. Dlatego należy zawsze pamiętać, wewnątrz której pętli się znajdujemy i działanie, której przerwie użycie instrukcji „break”.

Co do samego kodu to po wprowadzaniu litery

„t”

działanie się nie zmienia poza tym, że natychmiast wychodzimy z pętli. Po podaniu litery

„n”

po prostu wychodzimy z pętli, ponieważ nie musimy zmieniać dotychczasowej wartości zmiennej

”

wyjscie

”

, a podanie dowolnej innej litery spowoduje wyświetlenie tekstu, powrót do początku pętli i ponowne zadanie pytania.

Zadania:

4

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.

- Napisz program wczytujący z klawiatury wartości, a następnie wyświetlający średnią tych wartości, niech program kończy wprowadzanie, kiedy natrafi na cyfrę 0.
- Napisz program, który w zmiennej będzie przechowywał hasło i który po uruchomieniu będzie prosił użytkownika o podanie hasła i nie pozwoli przejść dalej, jeśli użytkownik nie poda właściwego hasła.

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.