Wrocław, 14 października 2022

Programowanie **w j**ęzyku **P**ython — Laboratorium 3

Instrukcje sterujące - instrukcja warunkowa, pętle

prowadzący: K. Kluwak

Instrukcje warunkowe

Instrukcje warunkowe umożliwiają podzielenie programu na bloki, których realizacja zależy od spełnienia tzw. warunków logicznych. Dzięki temu wynik działania programu może zależeć od wprowadzanych do niego wartości. Działanie instrukcji warunkowych można obrazowo porównać do zwrotnic na torach kolejowych - w zależności od ustawienia zwrotnicy pociąg (podobnie jak program) pojedzie "w lewo" lub "w prawo" na schemacie algorytmu naszego programu.

Słowa kluczowe:

- `if` - za pomocą tej instrukcji rozpoczynamy blok warunku
- `elif` - pozwala uwzględnić alternatywne warunki do sprawdzenia
- `else` - obsługuje sytuacje "wszystkie inne" niż te, zdefiniowane przy insktrukcjach if oraz elif.

W języku Python składnia instrukcji warunkowych jest następująca:

```
# Składnia instrukcji warunkowych:     #     <instrukcje>                    # --- lub ---

# if (warunki logiczne):               # --- lub ---                        # if (warunki logiczne):
#    <instrukcje>                                                           #    <instrukcje>
                                       # if (warunki logiczne):            # elif (warunki logiczne):
# --- lub ---                          #    <instrukcje>                   #     <instrukcje>
                                       # elif (warunki logiczne):          # ......
# if (warunki logiczne):               #     <instrukcje>                  # else:
#    <instrukcje>                      # ......                            #     <instrukcje>
# else:
```

Zad 2.17 *Napisz program, który wczyta liczbę wprowadzoną przez użytkownika, a następnie dokona klasyfikacji: liczba ujemna, liczba dodatnia, liczba równa zeru.*

Zad 2.18 *Napisz program, który sprawdza, czy wprowadzona przez użytkownika liczba jest parzysta czy nie oraz wyświetla w konsoli stosowny komunikat. Podpowiedź: Parzystość można sprawdzić za pomocą operacji modulo, która zwraca resztę z dzielenia (symbol %).*

Zad 2.19 *Trójmian kwadratowy wyraża się wzorem: . Napisz program, w którym użytkownik wprowadza współczynniki trójmianu kwadratowego (a, b, c), a następnie sprawdza, czy trójmian o podanych współczynnikach ma 2, 1 czy 0 rozwiązań w zbiorze liczb rzeczywistych.*

Uzupełnienie matematyczne: Należy sprawdzić znak delty: . Jeżeli:

- trójmian ma 2 pierwiastki,
- trójmian ma 1 pierwiastek,
- trójmian nie ma pierwiastków rzeczywistych.

Zad 2.20 *Napisz program, który przyjmie od użytkownika dwie wartości, a następnie wskaże wartość większą. Do rozwiązania zadania wykorzystaj instrukcje warunkowe.*

Zad 2.21 *Napisz program, który przyjmie od użytkownika dwie wartości, a następnie wskaże wartość mniejszą. Do rozwiązania zadania wykorzystaj instrukcje warunkowe.*

Zad 2.22 *Napisz program, który przyjmie 3 wartości, a następnie wskaże wartość najmniejszą i największą. Do rozwiązania zadania wykorzystaj instrukcje warunkowe.*

Zad 2.23 *Napisz program, który przyjmie 6 wartości, a następnie wskaże wartość najmniejszą i największą. Do rozwiązania zadania wykorzystaj instrukcje warunkowe.*

Zad 2.24 *Napisz program, w którym utworzysz i nadasz dowolne wartości liczbowe trzem zmiennym, natomiast program wypisze te wartości od najmniejszej do największej (wykona sortowanie). Do rozwiązania zadania wykorzystaj instrukcje warunkowe.*

Zad 2.25 *Napisz program do obliczania współczynnika BMI. Użytkownik podaje swoją wagę (w kilogramach) i wzrost (w metrach). Następnie program wykonuje obliczenia współczynnika zgodnie ze wzorem: Jeżeli wynik mieści się w przedziale (18.5-24.9) program wyświetla komunikat "waga jest prawidłowa". Jeżeli wynik jest poniżej wartości 18.5 to wyświetla komunikat "niedowaga". Jeżeli z kolei wynik jest powyżej wartości 24.9 to program wyświetla komunikat "nadwaga".*

Zad 2.26 *Napisz program, który przyjmie od użytkownika długości trzech boków trójkąta, a następnie sprawdzi czy z podanych boków możliwe jest utworzenie jakiegokolwiek trójkąta. Wyświetl stosowny komunikat.*

Zad 2.27 *Napisz program, który przyjmie od użytkownika długości trzech boków trójkąta, a następnie wyświetli komunikat, czy z takich boków można zbudować trójkąt prostokątny.*

Zad 2.28 *Utwórz program, w którym użytkownik może dokonać wyboru czy chce obliczyć pole czy obwód koła. Następnie program pobierze od użytkownika długości promienia, wykona obliczenia i wyświetli wynik na ekranie.*

Zad 2.29 *Napisz program umożliwiający wykonywanie działań prostego kalkulatora. Program powinien pracować na zmiennych rzeczywistych. Użytkownik wprowadza pierwszą liczbę, znaku działania (+, -, *, /) i drugą liczbę, natomiast program oblicza i wyświetla wynik działania na ekranie (w konsoli).*

Zad 2.30 *Napisz program do nauki tabliczki mnożenia. Program losuje liczbę z przedziału (1-10), wykonuje mnożenie i pyta użytkownika o wynik. Następnie użytkownik wprowadza wynik, a program udziela odpowiedzi czy jest poprawny czy nie. Podpowiedź: do losowania można wykorzystać funkcję randint z modułu random.*

Zad 2.31 *Napisz program, który będzie samodzielnie losować dwie liczby całkowite (0-100) i znak operacji (+, -, *), a następnie wyświetli utworzone równanie i zada pytanie o wynik. Użytkownik udziela odpowiedzi, a komputer ocenia czy odpowiedź jest dobra czy zła i wyświetla ją na ekranie (w konsoli).*

Zad 2.32 *Napisz program, który będzie dokonywać oceny hałasu na podstawie wprowadzonej wartości poziomu natężenia dźwięku. Przyjmij następujące zakresy: poziomu natężenia dźwięku pomiędzy: 0 a 50 dB - niski hałas, 50 a 100 dB - średni hałas, powyżej 100 dB - wysoki hałas.*

Zad 2.33 *Napisz program do wystawiania ocen. Uczniowie pisali sprawdzian, na którym maksylanie można było zdobyć 50 pkt. Użytkownik wprowadza uzyskaną liczbę punktów, a program zwraca ocenę jaką powinien uzyskać. Progi ocen są następujące: 0-39% - ndst, 40-49% - dop, 50-69% -dst, 70-84% - db, 85-99% bdb, 100% - cel.*

Zad 2.34 *Napisz program do sprawdzania długości tekstu dla recenzentów prac dyplomowych. Recenzja musi składać się conajmniej z 500 znaków (bez spacji). Użytkownik podaje tekst, a pogram zwraca informację ile posiada znaków, ile posiada znaków bez spacji oraz czy spełnia wymogi formalne dotyczące liczby znaków bez spacji.*

Pętle

Pętla umożliwia wykonanie danego fragmentu kodu więcej niż raz. Jeden cykl pętli jest nazywany iteracją.
Najważniejsze pętle to:

- pętla `for` - posiada wbudowany „licznik" (iterator)
- pętla `while` - domyślnie nie posiada „licznika" ale można go dodać.

Pętla for

Pętla for - składnia dla dowolnych obiektów:

```
# for i in range(liczba):                                          # lista = [8, 3, 4, 6, 1]
#     <instrukcje>                    # for i in range(poczatek, koniec, skok):
                                      #     <instrukcje>                    # for element in lista:
# for i in range(poczatek, koniec):   Pętla for dla typów sekwencyjnych:    #     <instrukcje>
#     <instrukcje>
```

```
# for indeks, element in enumerate(lista):slownik = {"k1" : "v1", "k2" : "v2"} #     <instrukcje>
#     <instrukcje>
                                        # for key in slownik:
```

Zad 2.35 *Napisz program, który wyświetli 10 razy na ekranie numer wynonanej iteracji oraz napis: "Ala ma kota".*

Zad 2.36 *Napisz program, który z przedziału liczb od 1-1000 wyświetli liczby podzielne przez 13. Podpowiedź: podzielność możemy sprawdzać korzystając z operacji modulo (obliczanie reszty z dzielenia, symbol %).*

Zad 2.37 *Napisz program wykonujący odliczanie do wystrzału rakiety. Na koniec odliczania program powinien wyświetlić komunikat "BUM!!!".*

Zad 2.38 *Napisz program, który przyjmie od użytkownika zdanie, a następnie przeliteruje je w domyślnej i odwróconej kolejności.*

Zad 2.39 *Napisz program obliczający dowolną potęgę zadanej liczby całkowitej (bez użycia operatora **). Użytkownik podaje liczbę całkowitą podnoszoną do potęgi oraz potęgę (liczba całkowita). Program wykonuje obliczenia i wyświetla odpowiedź.*

Zad 2.40 *Napisz program obliczający silnię z zadanej liczby. Użytkownik podaje liczbę całkowitą. Program wykonuje obliczenia i wyświetla odpowiedź. Uwzględnij przypadek 0!=1.*

Zad 2.41 *Napisz program obliczający zadany wyraz ciągu Fibonacciego. Użytkownik podaje numer wyrazu, program wykonuje obliczenia i wyświetla wynik.*

Zad 2.42 *Napisz program z wykorzystaniem pętli, w którym użytkownik poda ile liczb chce dodać do siebie, program pobierze od niego dokładnie tyle liczb, obliczy ich sumę i wyświetli ją na ekranie.*

Zad 2.43 *Napisz program, który przyjmie liczbę o dowolnej liczbie cyfr i wyświetli na ekranie sekwencję jej cyfr w formie słownej (np. 112 powinno dać w wyniku: jeden jeden dwa).*

Pętla while

Pętla while opiera się na istnieniu i spełnieniu pewnego warunku logicznego. Pętla while(warunek) nie posiada licznika. Jest wykonywana tak długo, jak długo jest spełniony jej warunek. Jego obecność umożliwia wprowadzenie licznika, ale również pozwala na stworzenie pętli nieskończonej.

Składnia pętli while:

```
# while(warunek):
#     <instrukcje>
```

Dodatkowe instrukcje:

- `continue` - pominięcie instrukcji i przejście do kolejnego cyklu pętli
- `break` - przerwanie działania pętli (wyjście)

Zad 2.44 *Rozbuduj program do nauki tabliczki mnożenia o menu które pojawi się po każdej próbie i pozwoli użytkownikowi wybrać czy chce grać dalej czy nie, oraz będzie zliczać jego punkty (dodawać punkty za dobre odpowiedzi i odejmować za złe) a na końcu gry będzie zwrcać statystyki gracza (ile miał pytań, ile odpowiedzi było poprawnych a ile nie, ile zdobył punktów, jaki był procent poprawnych odpwiedzi a jaki procent odpwiedzi niepoprawnych).*

Zad 2.45 *Napisz program, który będzie przyjmować od użytkownika liczby i będzie umieszczać je w liście. Etap wprowadzania liczb kończy się, kiedy użytkownik wprowadzi cyfrę 0. Następnie program obliczy i wyświetli sumę i średnią ze wprowadzonych liczb.*

Zad 2.46 *Napisz program do organizacji bazy klientów. W programie użytkownik podaje imię, nazwisko i numer telefonu klienta. Dane są zapisywane do listy. Program powinien posiadać menu D-odaj, P-okaż, K-oniec.*

Zad 2.47 *Sito Erastotenesa: Utwórz program, który przyjmie liczbę naturalną i sprawdzi czy jest pierwsza czy nie.*

Zad 2.48 *Utwórz grę, w której komputer losuje liczby z przedziału 1-100. Gracz ma odgadnąć co to za liczba poprzez wprowadzanie kolejnych odpowiedzi. Jeżeli podana przez gracza liczba jest za mała, otrzymuje komunikat "podałeś za małą liczbę". Jeżeli za duża - "podałeś za dużą liczbę". Jeżeli równa wylosowanej liczbie - gracz otrzymuje komunikat "Gratulacje" i gra zostaje zakończona.*

Zad 2.49 *Rozbuduj program losujący dwie liczby i znak działania (+, -, *) tak aby po każdym cyklu użytkownik mógł podjąc decyzję, czy chce grać dalej, czy chce zakończyć. Program zbiera informacje na temat dobrych i złych odpowiedzi użytkownika. Na koniec gry wyświetla na ekranie wynik i statystyki gracza (liczba dobrych odpowiedzi, liczba złych odpowiedzi, procent dobrze udzielonych odpowiedzi).*

Zad 2.50 *Napisz grę w kółko i krzyżyk człowieka z komputerem. W grze obowiązują klasyczne zasady. Układ planszy powinien być wymiaru 3x3.*

Zad 2.51 *Napisz program, który przyjmie od użytkownika pewną ilość liczb (uzytkownik decyduje ile), a następnie dokona ich sortowania. Zadanie wykonać nie korzystając z metody sort (należy samodzielnie zaprojektować algorytm sortowania).*

Wyrażenia listowe

Lista składana - popularna funkcjonalność języka Python - umożliwia szybkie tworzenie i modyfikowanie list.

Składnia: [wyrażenie + kontekst]

Wyrażenie określa działanie jakie powinno być wykonane na każdym elemencie listy, kontekst określa które elementy listy należy wybrać (może zawierać dowolną liczbę słów kluczowych `for`, `if`, itd.).

Przykład: Wyrażenie

```
[x for x in range(10)]
```

Utworzy nam listę elementów x o wartościach od 0 do 10.

Zad 2.52 *Za pomocą listy składanej utwórz listę zawierającą elementy od 0 do 100.*

Zad 2.53 *Za pomocą listy składanej utwórz sekwencję krotek zawierających pary liczb x, y w przedziale od 0 do 10 (dot. zarówno zmiennej x i y).*

Zad 2.53 *Korzystając z listy zarobków pracowników w firmie oraz z listy składane, zwróć tych pracowników, którzy zarabiają powyżej 5000 zł.*

```
pracownicy = [("Jan Kowalski", 3200), ("Mteusz Nowak", 4500),
("Anna Kubaczyk", 5200), ("Katarzyna Nowacka", 5100), ("Władysław Marciniak", 4800)]
```

Zad 2.54 *Korzystając z listy składanej napisz program, która zwróci listę liczb parzystych od o do 100.*

Zad 2.55 *Korzystając z listy składanej napisz program, który zwróci listę kwadratów liczb od 1 do 10.*

Zad 2.56 *Korzystając z listy słów ["ALA", "MA", "KOTA"] i z konstrukcji listy składanej, zamienić wszystkie słowa w liscie na słowa zawierające małe litery.*

Grafika żółwia (turtle graphics)

Grafika żółwia to interesujący moduł graficzny języka Python, który umożliwia w relatywnie prosty sposób wizualizować wyniki działań równań matematycznych i instrukcji programistycznych.

Importu modułu dokonujemy tak samo jak w przypadku innych, Możemy skorzystać np. z instrukcji:

- import turtle
- from turtle import *

Następnie musimy utworzyć obiekt `Turtle()`, którym dalej będziemy sterować za pomocą instrukcji języka Python.

Przegląd najważniejszych instrukcji i metod:

- `Turtle()` - konstruktor obiektu
- `speed()` - szybkość rysowania
- `forward() / fd()` - ruch do przodu
- `backward() / bk()` - ruch do tyłu

- `right() / rt ()` - obrót w prawo
- `left() / lt()` - obrót w lewo
- `goto() /setpos() / setposition()` - przejście do pozycji
- `pendown()  pd()  down()` - przyłóż pisak
- `penup()  pu()  up()` - podnieś pisak
- `pensize()  width()` - rozmiar pisaka
- `pencolor()` - kolor pisaka
- `showturtle()  st()— ` - pokaż żółwia
- `hideturtle()  ht()— ` - ukryj żółwia
- `isvisible()` - sprawdź czy żółw jest widoczny
- `shape()` - zmień kształt żółwia
- `resizemode()` - zmień rozmiar
- `shapesize()  turtlesize()` - zmień rozmiar

Zad 2.57 *Utwórz program, który narysuje kwadrat, którego każdy bok będzie posiadać inny kolor*

Zad 2.58 *Utwórz program, w którym zainicjowane zostanie obiekt typu Turtle i wykona rysunki trzech figur (kwadrat, pięciokąt, sześciokąt) w różnych miejscach okna w różnych kolorach.*

Zad 2.59 *Utwórz program, w którym korzystając z metody „setpos" narysujesz funkcję sinus i cosinus.*

Zad 2.60 *Utwórz program, w którym korzystając z metody „setpos" narysujesz okrąg.*

Zad 2.61 *Błądzenie chaotyczne: Utwórz program, w którym obiekt typu Turtle() będzie wykonywać krok naprzód, a następnie obracać się o wylosowany kąt (0, 90, -90, 180) i w pętli wykonywać błądzenie chaotyczne.*