

Wykład 2

- Pojęcie składni i semantyki języka
- Podstawowe konstrukcje języka Python

1

Podstawowe pojęcia

Aspekty języka programowania:

- **Syntaktyka** (składnia) - zbiór reguł określający formalnie poprawne konstrukcje językowe
- **Semantyka** - opisuje znaczenie konstrukcji językowych, które są poprawne składniowo

2

Notacja EBNF

Elementy notacji EBNF:

- Symbole pomocnicze (nieterminalne)
- Symbole końcowe (terminalne)
- Produkcje
- Metasymbole
 - $\langle \rangle$ - symbol pomocniczy
 - $::=$ - symbol produkcji
 - $|$ - symbol alternatywy
 - $[]$ - wystąpienie 0 lub 1 raz (EBNF)
 - $\{ \}$ - powtórzenie 0 lub więcej razy (EBNF)



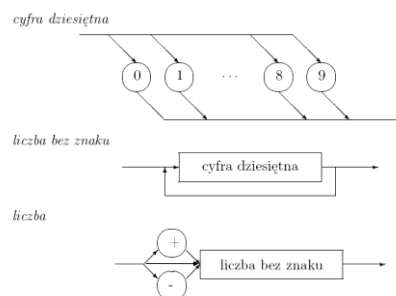
John Backus Peter Naur

Przykład

```
<cyfra dziesiętna> ::= 0|1|2|3|4|5|6|7|8|9
<liczba bez znaku> ::= <cyfra dziesiętna> { <cyfra dziesiętna> }
<liczba> ::= [+|-] <liczba bez znaku>
```

3

Diagramy składniowe



4

4

Gramatyka języka

```
<pgm> ::= <pgmHeading> <pgmDeclarations> <codeBlock> ";"
<pgmHeading> ::= program <pgmIdentifier> ";"
<pgmDeclarations> ::= { <pgmDeclaration> }
<pgmDeclaration> ::= <varDeclaration> | <typeDeclaration> | <procDeclaration>
....
<codeBlock> ::= begin { <statement> } end
<statement> ::= <assignStatement> | <ifStatement> | <whileStatement> | <procCall>
<assignStatement> ::= <variable> "=" <expression> ";"
<ifStatement> ::= if <expression> then <codeBlock> [ else <codeBlock> ] ";"
<whileStatement> ::= while <expression> do <codeBlock> ";"
....
<identifier> ::= <letter> { <letter> | <digit> }
<longint> ::= <digit> { <digit> }
<relOperator> ::= "=" | "<" | ">" | "<=" | ">=" | ">="
<addOperator> ::= "+" | "-" | "*" | "/"
<multOperator> ::= "+" | "-" | "*" | "/" | "div" | "and"
<letter> ::= "A" | ... | "Z" | "a" | ... | "z"
<digit> ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
```

while a>0 do a:=a-1
begin end

5

Język Python

- Stworzony przez holendra Guido van Rossum w 1991 roku
- Open Source
- Interpretowany, interaktywny, zorientowany obiektowo
- Przenośny
- Bogata biblioteka funkcji
- Dobra dokumentacja
- Łatwy do nauki, przejrzysty w zapisie
- Funkcje, moduły, klasy, pakiety
- Dynamiczne typowanie
- Automatyczne zarządzanie pamięcią
- Obsługa wyjątków
- Struktury wysokiego poziomu: zbiory, krotki, listy, słowniki
- Dostępny na wiele urządzeń i systemów operacyjnych

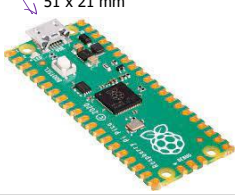
3.10

x = 6
x = True
x = 'ole'

6

Micro Python

Układ	Raspberry Pi RP2040
Rdzeń	ARM Cortex-M0+ Dual-Core 133 MHz
Pamięć SRAM	264 kB
Pamięć Flash	2 MB
Wyposażenie	40-pin / 23 GPIO cyfrowe + 3 piny ADC
Interfejsy	2x UART, 2x I2C, 2x SPI, do 16 kanałów PWM
Napięcie zasilania	5 V (USB)
Wymiary	51 x 21 mm



7

Instrukcja przypisania

<zmienna> = <wyrażenie>

~~!=~~ ~~x+=2~~ ~~x+=2~~ ~~x+=2~~

$x+=2 \Rightarrow x = x+2$

$y = 1/x$

$y = \sin(x)$

$C/C++$

$C = a+b$

28 -32768-32769

```
>>> a = 23
>>> a = "to jest napis"
>>> print(a)
to jest napis
>>> b, c = 6, 'kotek'
>>> print(b, c)
6 kotek
>>> c, b = b, c
>>> print(b, c)
kotek 6
>>> x = y = z = 0
```

8

Operacje wejścia/wyjścia

```
>>> a = input("Jak masz na imię: ")
Jak masz na imię: Marek
>>> print(a)
Marek
>>> b = int(input("Podaj liczbę całkowitą: "))
Podaj liczbę całkowitą: 23
>>> print(b)
23
>>> c = float(input("Podaj liczbę rzeczywistą: "))
Podaj liczbę rzeczywistą: 0.1428
>>> print(c)
0.1428
```

9

Instrukcja skoku warunkowego

if <wyr>:
 <instrukcja>
elif <wyr>:
 <instrukcja>
else:
 <instrukcja>

end = None

if a > 0;
 a = c
 b = b + 1
 print('u')

instr1, instr2, instr3

instr

10

Instrukcja wyboru

```
match <term>:  
  case <pattern_1>:  
    <action_1>  
  case <pattern_2>:  
    <action_2>  
  ...  
  case _:  
    <action-default>
```

Przykład:

```
match dzien_tygodnia:  
  case 0: print("niedziela")  
  case 1: print("poniedziałek")  
  ...  
  case 6: print("sobota")  
  case _: print("zła wartość")
```

if, elif, else

3.10

11

Instrukcja pętli while

while <wyrażenie>:
 <instrukcja>
 ...

Przykład:

```
n = int(input("podaj liczbę"))  
while n!=1:  
    if n%2==0:  
        n = n//2  
    else:  
        n = 3*n+1  
print("koniec")
```

while 0%2: 0=0-1

10, 5, 2, 1

9x

11

11, 34, 13, 52, 19, 26, 13, 40, 20, 10

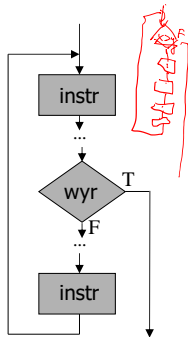
12

Przerywanie pętli - break

Przykład:

Wczytaj ciąg liczb zakończony zerem,
i oblicz ich średnią arytmetyczną

```
suma = 0
licznik = 0
while True:
    a = int(input("podaj liczbę"))
    if a == 0: break
    suma = suma + a
    licznik = licznik + 1
print(suma/licznik)
```



13

Przerywanie pętli - break

Wczytaj ciąg liczb zakończony zerem, którego suma nie
przekracza 100 i oblicz ich średnią arytmetyczną

```
suma = 0
licznik = 0
while suma <= 100:
    a = int(input("podaj liczbę"))
    if a == 0: break
    suma = suma + a
    licznik = licznik + 1
else:
    print('Uwaga suma > 100')
print('średnia ', suma/licznik)
```

while < w1 :
=
=
if < w2 > : break
=
else :
print('ale')

14

Instrukcja pętli for

for <zmienna> in <sekwencja>:
<instrukcja>

...

Przykłady:

```
for x in range(1, 6):
    print(x, "x", "x" = "x*x")
```

```
1 * 1 = 1
2 * 2 = 4
3 * 3 = 9
4 * 4 = 16
5 * 5 = 25
```

range(5) 0 1 2 3 4
range(1,6) 0 1 2 3 4 5
range(1,10,2) 1 3 5 7 9

for x in "Python":
print(x)

P
y
t
h
o
n

15

Wyjątki

try:

kod w którym może wystąpić błąd

except <typ błędu>:

kod wykonywany po wystąpieniu błędu

Przykład:

```
while True:
    try:
        a = float(input("a="))
        b = float(input("b="))
        c = float(input("c="))
        break
    except ValueError:
        print("Podaj 3 liczby !!!")
```

ale

16

Procedury i funkcje

Cel stosowania:

- dekompozycja problemu
- wielokrotne wykonanie
- poziomy abstrakcji
- oddzielna kompilacja
- możliwość użycia rekurencji

17

Procedury i funkcje

def <name> (<arg1>, <arg2>, ...):

<instrukcja>

...

end def

return

return *expression*

procedura

funkcja

18

Procedury i funkcje

```
def nwd(a, b):  
    """Funkcja oblicza największy wspólny dzielnik.  
    Nie jest najszybsza ale działa."""  
    while a != b:  
        if a>b: a=a-b  
        else: b=b-a  
  
    return a  
# end def  
  
a = int(input("podaj a:"))  
b = int(input("podaj b:"))  
c = nwd(a,b)  
if c==1:  
    print("Liczby a,b są względnie pierwsze")
```