

Podstawy Programowania ELEMENTY PROGRAMU i TYPY DANYCH

Michał Bujacz

bujaczm@p.lodz.pl

B9 „Lodex” 207

**godziny przyjęć: środy i
czwartki 10:00-11:00**

<http://www.eletel.p.lodz.pl/bujacz/>



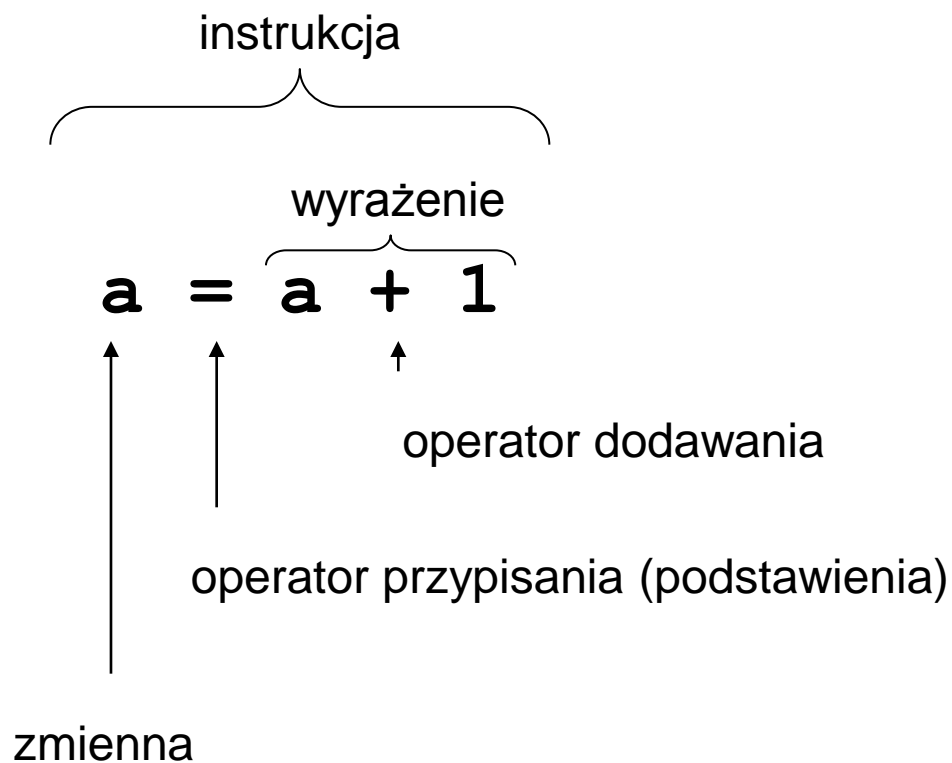
Pytania weryfikacyjne:

- Wiedza deklaratywna vs imperatywna
- 6 kroków pracy programisty
- Sposoby projektowania programów
- Klasyfikacje języków programowania
- Poziom/generacja języka programowania
- Kompilator vs. interpreter
- Paradygmaty programowania

Słownictwo

- Instrukcja (Instruction)
- Zmienna (Variable)
- Słowo kluczowe (Keyword)
- Wyrażenie (Expression)
- Operator (Operator)
- Komentarz (Comment)

Przykładowa instrukcja



Instrukcja

- Program komputerowy to ciąg instrukcji w języku programistycznym
- Podstawowe typy instrukcji:
 - czynne (zmieniają wartości zmiennych, wywołują kolejne instrukcje)
 - bierne (deklaracje zmiennych, specyfikacje funkcji/obiektów)

Zmienna

- Nazwa (identyfikator) przypisana do danych zapisanych w pamięci komputera
- Większość języków programistycznych posiada wiele typów danych które określa się na stałe dla danej zmiennej (JAVA, C)
- Niektóre języki (np. Python, Pearl, PHP, JavaScript) nie wymagają stałego deklarowania typu, gdyż może on się zmieniać w trakcie działania programu (typowanie dynamiczne)

Nazwy zmiennych

- **mogą** zawierać duże i małe litery cyfry, `_`, w niektórych językach również `$`
- **nie mogą** zaczynać się od cyfry
- nie mogą pokrywać się ze słowami kluczowymi danego języka (reserved keywords)

<code>and</code>	<code>del</code>	<code>for</code>	<code>is</code>	<code>raise</code>
<code>assert</code>	<code>elif</code>	<code>from</code>	<code>lambda</code>	<code>return</code>
<code>break</code>	<code>else</code>	<code>global</code>	<code>not</code>	<code>try</code>
<code>class</code>	<code>except</code>	<code>if</code>	<code>or</code>	<code>while</code>
<code>continue</code>	<code>exec</code>	<code>import</code>	<code>pass</code>	<code>yield</code>
<code>def</code>	<code>finally</code>	<code>in</code>	<code>print</code>	

Słowa kluczowe: Python ~30, JAVA ~50, C++ ~80-150

Typy danych

- Liczbowe – całkowite i zmiennoprzecinkowe
- Znakowe (char)
- Tekstowe
- Tablicowe
- Wyliczeniowe (listy)
- Wskaźnikowe
- Strukturalne
- Logiczne (boolean)
- Klasy (typ obiektowy)

Systemy liczbowe

- Dziesiętny
- Binarny (bit, bajt, słowo)
- Szesnastkowy
- Kolejność bajtów: Big endian / little endian (grubokońcowe/cienkokońcowe)

System dziesiętny

◦ 1234

$$= 1 * 10^3 + 2 * 10^2 + 3 * 10^1 + 4 * 10^0$$

System binarny

◦ 1101

$$= 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0$$

1 Bajt = 8 bitów



2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

Liczby ujemne

- Sign bit

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
+/-	64	32	16	8	4	2	1

- Two's complement (invert bits, add 1)

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
-128	64	32	16	8	4	2	1

Szesnastkowy (hexadecimal)



2^3	2^2	2^1	2^0
8	4	2	1

Hexadecimal	Binary	Decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

o F201(hex)
 $= 15 \cdot 16^3 + 2 \cdot 16^2 + 0 \cdot 16^1 + 1 \cdot 16^0$ (dec)
 $= 1111\ 0010\ 0000\ 0001$ (bin)

Kolejność bajtów w pamięci

- Wartość 0x4A3B2C1D w pamięci pod adresem 100
- **Big-endian** (grubokońcowość)

	100	101	102	103	
...	4A	3B	2C	1D	...

- **Little-endian** (cienkokońcowość)

	100	101	102	103	
...	1D	2C	3B	4A	...

Liczby całkowite (integer, long)



- W zależności od ilości bitów, różny górny limit
- Przykłady z Javy:

JAVA Integral Types	Memory Storage	Approximate Range
<code>int</code>	4 bytes	+/- 2 billion
<code>short</code>	2 bytes	+/- 32,767
<code>long</code>	8 bytes	+/- 9.2 x 10 ¹⁸
<code>byte</code>	1 byte	-128 to 127

- Python standardowo używa aż 8 bajtów na zwykły **int** (maksymalnie $2^{63} - 1 = 9223372036854775807$)
- Automatycznie zmienia na **long** (36 bajtów) gdy ten limit zostaje przekroczony i dynamicznie zmienia rozmiar

Liczby zmiennoprzecinkowe (float, double)



- Rozbite na dwa elementy:

$$a^b$$

- a - mantysa (mantissa)
- b - wykładnik (exponent)

- Pewna ilość bajtów na jedno i drugie
- W Pythonie **float** ma 8 bajtów - 1 bit znak, 11 bitów wykładnik, 52 bity mantysa

JAVA DATA TYPES



Groups	Variable Type	Storage	Approximate Range
Integrals	<code>int</code>	4 bytes	+/- 2 billion
	<code>short</code>	2 bytes	+/- 32,767
	<code>long</code>	8 bytes	+/- 9.2 x 10 ¹⁸
	<code>byte</code>	1 byte	-128 to 127
Floating-point	<code>float</code>	4 bytes	7-8 digits
	<code>double</code>	8 bytes	16-17 digits
Character	<code>char</code>	2 bytes	65,536
Boolean	<code>boolean</code>	1 bit	<code>true</code> and <code>false</code>

Zmienne znakowe - ASCII



Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

OPERATORY

- Arytmetyczne (+,-,/....)
- Jednoargumentowe (Unary) (!,~, -, ++...)
- Przypisania (Assignment) (=, +=...)
- Porównania (Relational) (==, !=, <=...)
- Logiczne (and, or not) i bitowe (&, |, ~)
- Przynależności (in)

<i>ARYTMETYCZNE</i>		<i>JEDNOARGUMENTOWE</i>	
+	Addition	-	Arithmetic negation
-	Subtraction (also unary minus)	!	Logical negation
*	Multiplication	++	Increment
/	Division	--	Decrement
%	Modulus (remainder from division)		
**	Exponent		
//	Floor division		

Styl

- Zbiór wytycznych w języku programistycznym nie wpływających na działanie programu
- Wpływa na czytelność kodu
- Nazewnictwo zmiennych
- Wcięcia kodu, spacje, komentarze
- PEP 8 – obowiązujący styl Pythona

Styl PEP 8 – wcięcia i linijki

- Wcięcia na 4 spacje
- Bez tabulatorów (chyba że edytujemy plik w którym już ich używano)

- Puste linie :
 - Pomędzy klasami 2
 - Pomędzy metodami, funkcjami, atrybutami 1

- Znaków w linii 72

Styl PEP 8 - spacje

- Spacje
 - po przecinku, nie przed
 - dookoła operatorów
- Brak spacji
 - dookoła operatorów jednoargumentowych
 - przed otwarciem nawiasu lub klamry
 - Puste linie

Styl PEP 8 - komentarze

- Komentarz na początku pliku informuje o użytym kodowaniu ASCII (np. US-ASCII lub UTF-8)
- Blok dokumentacyjny na początku skryptu
- Komentarze zazwyczaj po #
- Każda funkcja udokumentowana po nagłówku:

```
def funkcja_add(self):  
    " " " Opis funkcji ...  
    " " "
```

Styl PEP 8 – nazewnictwo

- Stałe - **WIELKIMI_LITERAMI**
- Klasy - **ZWielkichLiter** (jak w JAVA)
- Zmienne - **male, krotkie,**
- Zmienne lokalne tymczasowe - **a, b, c**
- Funkcje - **podkresl_jesli_konieczne**