

# Kategorie znaków

**char** ::= *printable character*

**letter** ::= "a".."z" | "A".."Z" | "\_"

**dig** ::= "0".."9"

**hexdig** ::= "0".."9" | "A".."F"

**selchar** ::= "+" | "-" | "\*" | "/" | "~" | "|" |  
| "," | "<" | ">" | "=" | "&"

# Identyfikatory i symbole

**id** ::= letter[letter|dig]\*

Array  
anArray  
new

---

**bin sel** ::= selchar[selchar]

+  
<=

---

**key sel** ::= id ":"

at:  
ifTrue:

---

**symbol** ::= id | bin sel | key sel[key sel]\*

at:put:  
at:ifAbsent:

---

# Literały

**number** ::= [[dig]+ "r" ] ["-"] [hexDig]+  
["." [hexDig]+] ["e"["-"]][dig]+]

```
13  
-5  
0.17  
16r1C  
333e-2
```

---

**charconst** ::= "\$"char

```
$A  
$7  
$+
```

---

**string** ::= ""[char]\*""

```
'Ala ma kota'  
'10000$ za głowę Dużego Mike''a'
```

---

**symconst** ::= "#"symbol

```
#var  
#Array  
#+  
#at:put:  
#'Ala ma kota'  
#'a^3'
```

---

**array** ::= "(" [number | string | symbol | array | charconst]\* ")"

**arrayconst** ::= "#"array

```
#('mon' 'tue' 'wed' 'thu' 'fri')  
#(1.3 'Ala' $c)  
#(#name (1 2 3) 'string')
```

---

**literal** ::= number | string | charconst | symconst | arrayconst

# Wyrażenia

**unit** ::= id | literal | block | "(" expr ")"

**unaryexpr** ::= unit [id] +

**priexpr** ::= unit | unaryexpr

```
Set new  
'Ala ma kota' printNl  
#(1 2 3 3 3) asSet
```

---

**binmsg** ::= binrel priexpr

**binexpr** ::= priexpr [binmsg] +

```
2 + 2  
$a <= $b  
anArray size + 1
```

---

**pribinexpr** ::= binexpr | priexpr

**keymsg** ::= [keysel pribinexpr] +

**keyexpr** ::= pribinexpr keymsg

```
Array new: 20  
anArray at: pos - 3 put: Set new  
Set new add: #(1 2 3) size
```

---

**msgexpr** ::= unaryexpr | binexpr | keyexpr

**cascade** ::= id | binmsg | keymsg

**expr1** ::= unit | msgexpr [";" cascade]\*

**expr** ::= [id ":@"]\* expr1

```
aSet add: 3; add: 'foo'; add: $k  
x := Array new: 20  
m := n := 1 + anArray size
```

---

**exprs** ::= [expr "."]\* [{"^"} expr]

```
x := Array new: 20.  
x at: 20 put: 'end'.  
^x
```

---

**varsel** ::= ":"id

**block** ::= "[" [[varsel]+ "|" ] exprs "]"

```
[ :x | x printNl ]  
[ :aSet :anElem | aSet add: anElem ]
```

---

# Definicje usług

**temps** ::= "|" [id]\* "|"

**message** ::= id | binsel id | [keysel id]+

**method** ::= message [temps] exprs

**methods** ::= "!"id ["class"] "methodsFor:"  
string "!" [method "!" ]+ "!"

```
! MyClass class methodsFor: 'creation' !
new
    |r|
    r := super new.
    r init.
    ^r
!!
```

```
! MyClass methodsFor: 'initialization' !
init
    var := 0
!!
```