

Špecifikácia ScaPy 2.0

ScaPy je programovací jazyk inšpirovaný programovacími jazykmi Scala a Python. ScaPy je silno typový procedurálny programovací jazyk so syntaxou veľmi podobný jazyku Scala s prvkami Pythonu.

Syntax

- **znak**: [a..zA..Z]
- **číslo**: [0..9]
- **identifikátor**: [znak][znak, číslo]*

Inak povedané identifikátor je string, ktorý obsahuje znaky a písmená a musí začínať písmenom.

Identifikátory sú používané ako názvy premenných, funkcií alebo tried. Musí byť rôzny od kľúčových slov.

- príklad identifikátorov

```
a, b4, c4c
```

Zoznam kľúčových slov

- **if**
- **else**
- **for**
- **true**
- **false**
- **while**
- **and**
- **or**
- **not**
- **val**

- **var**
- **def**
- **return**

Podporované dátové typy

- **Boolean**, boolean hodnota (true alebo false)
- **Char**, znak
- **String**, reťazec znakov
- **Int**, 32-bitový integer
- **Float**, desatinné číslo
- **Array**, pole

Operátory

- **+, -, *, /, %**
- **+=, -=, *=, /=, %=**
- **==, !=, <, >, <=, >=**
- **and, or, not**
- **=**

Deklarácia konštánt

- statická deklarácia

```
val <názov konštanty> : <typ> = <hodnota>
```

- dynamická deklarácia

```
val <názov konštanty> = <hodnota>
```

- príklady

```
val pi : Float = 3.14
```

```
val ch = 'a'
```

Deklarácia premenných

Podobne ako pri konštantách:

- statická deklarácia

```
var <názov> : <typ> = <hodnota>
```

- dynamická deklarácia

```
var <názov premennej> = <hodnota>
```

- príklady

```
var i : Int = 0  
  
var s = "Hello word"
```

Polia

- statická deklarácia

```
val <názov premennej> : Array[<Typ>] = Array[<typ>](<veľkosť poľa>)
```

- dynamická deklarácia

```
val <názov premennej> = Array[<typ>](<veľkosť poľa>)
```

- čítanie v poli, (indexácia od 0)

```
<názov premennej> [index v poli]
```

- zápis v poli

```
<názov premennej> [index v poli] = <hodnota>
```

- všetky prvky v poli majú hodnotu po deklarácii podľa typu:

- typ: **Int**, hodnota: 0
- typ: **Char**, hodnota: “
- typ: **Float**, hodnota: 0.0
- typ: **Boolean**, hodnota: false

- príklady

```
val pole = Array[Int](4)

pole[0] = 1

printInt(pole[0]) # 1
printInt(pole[2]) # 0

val pole2 = Array[Char](2)

pole2[0] = 'a'
pole2[1] = 'h'

printString(pole2) # ah
```

Cykly a riadenie toku

- **while** príkaz

```
while(){
# telo cyklu
}
```

- **for** príkaz

```
for (<názov premennej> <- <hodnota> until <hodnota>){
# telo cyklu
}
```

- premenná vo **for** nemusí byť predtým deklarovaná

- príklad

```
for (i <- 0 until 10) printInt(i)  
# 0 1 2 3 4 5 6 7 8 9
```

- **if** príkaz

```
if (<podmienka>) {  
  # ak je podmienka pravdivá, riadenie pokračuje tu  
}  
else {  
  # ak podmienka nie je pravdivá, riadenie pokračuje tu  
  # else je nepovinná časť  
}
```

- príklad

```
var b = 0  
while (b <= 10){  
  for (i <- 0 until 5){  
    if (i == b) {  
      printInt(i)  
    }  
    else {  
      printInt(b)  
    }  
  }  
  b += 1  
}
```

Funkcie a procedúry

- deklarácia funkcie

```
def <názov funkcie> (<názov arg0>: <typ>, ... ,<názov argn>: <typ>): <Typ náv  
  # telo funkcie  
  return <návratová hodnota>  
}
```

- deklarácia procedúry

```
def <názov procedúry> (<názov arg0>: <typ>, ... ,<názov argn>: <typ>) = {  
    # telo procedúry  
}
```

- príklad

```
def fib(a: Int): Int = {  
    if ((a == 0) or (a == 1)) return 1  
    if (a == 2) return 2  
    return fib(a-1) + fib(a-2)  
}  
  
for (i <- 0 until 10){  
    printInt(fib(i))  
}  
  
# 1 1 2 3 5 8 13 21 34 55
```

Komentáre

- za komentár sa označuje riadok, ktorý začína symbolom #
- príklad

```
# toto je komentar
```

Input/Output

- **printFloat** príkaz, vypíše desatinné číslo na STDOUT

```
printFloat(<hodnota>)
```

- **println** príkaz, vypíše číslo na STDOUT

```
printInt(<hodnota>)
```

- **readInt** príkaz, číta číslo zo STDIN

```
<nazov premennej> = readInt()
```

- **printString** príkaz, vypíše string(pole charov) na STDOUT

```
printString(<hodnota>)
```

- **printString2** príkaz, vypíše string(pole charov) s danou dĺžkou na STDOUT

```
printString(, )
```

- **readString** príkaz, číta string zo STDIN

```
<nazov premennej> = readString()
```

- príklad


```
printInt(5)

var line = readString()
printString(line)
```

Používanie externých funkcií

- import externej funkcie

```
import <názov funkcie>(<názov arg0>: <typ>, ... ,<názov argn>: <typ>) : <Typ>
```



- príklad

```
import myrandom():Int

printInt(myrandom()) # 47
```

Sémantika

- je zakázané do deklarovanej premennej priraďovať iný typ ako bola deklarovaná
- príklad

```
var i = 10
i = 1.0 # Error
```

- priradiť hodnotu do konštanty
- príklad

```
val a = 4
a = 5 # Error
```

Príklady

- hello.scapy

```
printString("Hello World!")
```

- primes.scapy


```

def check(prime :Int): Boolean = {
  for (i <- 2 until prime){
    if ((prime % i) == 0){
      return false
    }
  }
  return true
}

var n = readInt()
var i = 2

while (i <= n){
  if (check(i)) printInt(i)
  i += 1
}

```

- sort.scapy

```

var n = readInt()
val pole = Array[Int](n)

for (i <- 0 until n) pole[i] = readInt()

# bubble sort
for (i <- 0 until n){
  for (j <- i+1 until n){
    if (pole[i] > pole[j]){
      var tmp = pole[i]
      pole[i] = pole[j]
      pole[j] = tmp
    }
  }
}

for (i <- 0 until n) printInt(pole[i])

```

- reverz.scapy

```

var a = readString()
var length = getLengthString(a)

val reverz = Array[Char](length)

var ind1 = length - 1
var ind2 = 0
while (ind1 >= 0){
    reverz[ind2] = a[ind1]
    ind2 += 1
    ind1 -= 1
}

printString2(reverz, length)

```

- graphs.scapy

```

val n = readInt()
val m = readInt()

val pole = Array[Int](n)

for (i <- 0 until n){
    pole[i] = -1
}

def findset(pole: Array[Int], x: Int): Int = {
    var i = x
    var k = x
    while(pole[i] >= 0){
        i = pole[i]
    }

    while(pole[x] >= 0){
        k = pole[x]
        pole[x] = i
        x = k
    }

    return x
}

def union(pole: Array[Int], x: Int, y: Int){
    val sx = findset(pole, x)
    val sy = findset(pole, y)

```

```

        if (sx != sy){
            if (pole[sx] < pole[sy]){
                pole[sx] += pole[sy]
                pole[sy] = sx
            }
            else{
                pole[sy] += pole[sx]
                pole[sx] = sy
            }
        }
    }

def printArray(pole: Array[Int], n: Int){
    for(i <- 0 until n){
        printInt(pole[i])
    }
}

for (i <- 0 until m){
    val x = readInt()
    val y = readInt()
    union(pole, x, y)
}

var components = 0
for (i <- 0 until n){
    if (pole[i]<0){
        components += 1
    }
}

if (components == 1){
    printString("YES")
}
else {
    printString("NO")
}

```