

# Algoritmus a pseudokód

Bc. Katarína Olejková



KATEDRA INFORMATIKY  
UNIVERZITA PALACKÉHO V OLMOUCI

# Obsah

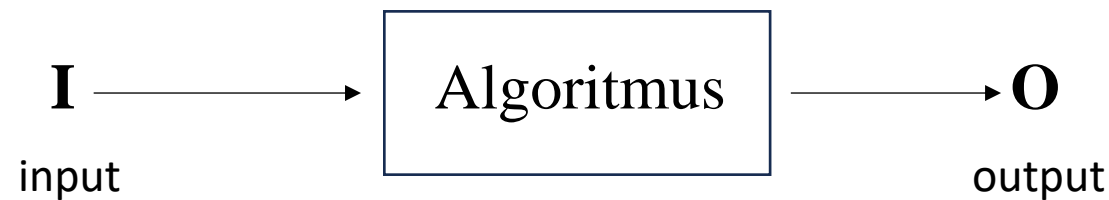
- Zopakovanie termínov
  - Algoritmus
  - Problém
  - Inštrukcia
  - Riešenie problému
- Pseudokód
  - Zápis
  - Príklady

# Algoritmus

- ALGORITMUS JE POSTUPNOSŤ INŠTRUKCIÍ? PRE RIEŠENIE? PROBLÉMU?
  - Táto definícia ale vystihuje podstatu pojmu algoritmus
- Vlastnosti:
  - Konečnosť – algoritmus skončí po prevedení konečného počtu krokov
  - Korektnosť – výstup algoritmu musí byť správny
  - Rezultatívnosť – algoritmus má aspoň jeden výstup
  - Opakovateľnosť – algoritmus má pre rovnaký vstup rovnaký výstup
  - Obecnosť – algoritmus nerieši jeden konkrétny problém ale triedu obdobných problémov
  - Jednoznačnosť – každý krok algoritmu je jednoznačne definovaný

# Problém

- Priradenie, ktoré každému prípustnému vstupu **I** priradí zodpovedajúci výstup **O**
  - Vstup – množina prípustných zadaní
  - Výstup – množina odpovedajúcich riešení
- budeme sa zaoberať problémami, ktoré sú riešiteľné algoritmami



# Inštrukcia

- Jednoznačný zreteľný pokyn
- musí sa dať vykonať mechanicky bez premýšľania
- Základné inštrukcie:
  - Priradenie
  - Aritmetické
  - Logické
  - Podmienené
  - Vstupné/Výstupné
  - Cykly
  - ...

# Riešenie problému

- Algoritmus rieši problém pokiaľ pre každý vstup sa vykonávaním inštrukcií po konečnom počtu krokov dopravuje k výstupu
- Problém: Zistite, či je dané celé číslo kladné
- Vstup: ľubovoľné celé číslo  $n$
- Výstup: áno/nie
- Algoritmus:
  - prečítaj číslo  $n$  zo vstupu
  - Ak  $n > 0$  zapíš na výstup “áno”
  - Inak zapíš na výstup “nie”

# Pseudokód

- Budeme pomocou neho zapisovať algoritmy
- Odľahčená syntax
- Veľmi často používaný (diplomky, učebné texty, vedecké články..)
- Je možné ho prepísať do ľubovlného programovacieho jazyka

# Pseudokód

- Priradenie:  $\leftarrow$

$i \leftarrow 5$

$j \leftarrow i * 2$

$a \leftarrow 'f'$

// písmená budeme značiť pomocou ''

// reťazce budeme značiť pomocou ""

- Aritmetické:  $+$   $-$   $*$   $/$  **mod** (zvyšok po delení)

$x \leftarrow 5$

$y \leftarrow 2$

$p \leftarrow x \bmod y$  // p bude mať hodnotu 1, lebo  $5 / 2 = 2$  zvyšok 1



# Pseudokód

- Logické (true/false): = != < > ≤ ≥ **and or**

c ← 5

c = 5               // true

c != 5             // false

c > 7              // false

c < 6              // true

(c = 5) and (c < 4)   // false

(c > 6) or (c < 7)    // true

# Pseudokód

- Podmiené: **if if-else if-elseif-else**

```
if podmienka1  
    print("podmienka 1 je true")  
    ...  
// ďalší kód
```

```
if podmienka1  
    print("podmienka 1 je true")  
    ...  
else  
    print("podmienka 1 je false")  
    ...  
// ďalší kód
```

```
if podmienka1  
    print("podmienka 1 je true")  
else if podmienka2  
    print("podmienka 2 je true")  
else  
    print("podmienka 1 a podmienka 2 su false")  
// ďalší kód
```

- Pozn. Odsadenie riadkov je dôležité!

# Pseudokód

- Cykly: **for** **while** **do-while**

```
n ← 10
for i ← 0 to n
    print(i)
```

```
n ← 10
i ← 0
while (i < n)
    print(i)
    i ← i + 1
```

```
i ← 0
do
    print(i)
    i ← i + 1
while (i ≤ 10)
```

- Pozn. Odsadenie riadkov je dôležité!

# Pseudokód

- Pole

$A[0..n-1]$     pole A s n prvkami (n – počet prvkov)  
 $A, n$             pole A s n prvkami (n – počet prvkov)

- Indexovanie poľa od nuly

- Príklad:

$A[0..3] \leftarrow \text{"Code"}$

print  $A[0]$     // C

print  $A[1]$     // o

$A[2] \leftarrow \text{'m'}$

print  $A[2]$     // m

print  $A[3]$     // e

A

n = 4

index:

0	1	2	3
C	o	d	e

C	o	m	e
---	---	---	---

# Pseudokód

- Indexovanie poľa

- Príklad:

$A[0..2] \leftarrow [2, 4, 6]$

$A[0] \leftarrow A[1]$

$A[2] \leftarrow A[1]$

print A[0] // 4

print A[1] // 4

print A[2] // 4

A

n = 3

index:    0       1       2

2	4	6
4	4	6
4	4	4

# Pseudokód

- Zápis na výstup

$i \leftarrow 5$

`print("Hello World")`      `// Hello World`

`print(i)`      `// vráti 5 ako text`

- Návratová hodnota

$i \leftarrow 5$

`return i`      `// vráti 5 ako číslo`

# Príklad 1 - Zapište algoritmus JeKladne(n) pomocou pseudokódu

- Problém: Zistite, či je dané celé číslo kladné
- Vstup: ľubovoľné celé číslo **n**
- Výstup: “áno”/ “nie”

- Algoritmus:

- Ak **n** > 0 zapíš na výstup “áno”
- Inak zapíš na výstup “nie”

- Riešenie:

```
1. JeKladne(n)
2.   if n > 0
3.       print(“áno”)
4.   else
5.       print(“nie”)
```

## Príklad 2 – Navrhните algoritmus SudeNeboLiche(n) a zapíšte ho pomocou pseudokódu

- Problém: Zistite, či je dané celé číslo sudé alebo liché
- Vstup: ľubovoľné celé číslo **n**
- Výstup: “sudé”/“liché”

- Riešenie:

```
1. SudeNeboLiche(n)
2.   if  $n \bmod 2 = 0$ 
3.       print(“sudé”)
4.   else
5.       print(“liché”)
```



## Príklad 3 – Navrhните algoritmus Signum(n) a zapíšte ho pomocou pseudokódu

- Problém: Vypočítajte funkciu signum pre dané celé číslo
- Vstup: ľubovoľné celé číslo **n**
- Výstup: 1/0/-1

Signum(n) = 1 pre  $n > 0$   
              0 pre  $n = 0$   
             -1 pre  $n < 0$

- Riešenie:

```
1. Signum(n)
2.   if n > 0
3.       return 1
4.   else if n = 0
5.       return 0
6.   else
7.       return - 1
```

## Príklad 4 – Navrhните algoritmus Sucet(n) a zapíšte ho pomocou pseudokódu

- Problém: Pre zadané celé číslo  $n$  vypočítajte súčet všetkých čísel od 1 do  $n$
- Vstup: ľubovoľné celé číslo  $n$
- Výstup: súčet

- Riešenie:

```
1. Sucet(n)
2.   sum ← 0
3.   for i ← 1 to n
4.     sum ← sum + i
5.   return sum
```

Príklad 5 – Navrhните algoritmus  $\text{SucetPole}(A[0..n-1])$  a zapíšte ho pomocou pseudokódu

- Problém: Pre zadané pole  $A[0..n-1]$  vypočítajte súčet všetkých jeho prvkov
- Vstup: Pole  $A[0..n-1]$  (pole  $A$  s  $n$  prvkami)
- Výstup: súčet

Príklad 6 – Navrhnite algoritmus `AritmetickyPrumer(A[0..n-1])` a zapíšte ho pomocou pseudokódu

- Problém: Pre zadané pole  $A[0..n-1]$  vypočítajte aritmetický priemer jeho prvkov
- Vstup: Pole  $A[0..n-1]$  (pole  $A$  s  $n$  prvkami)
- Výstup: priemer