

# Počítače a algoritimizácia

## Metódy algoritmizácie a základné zložky algoritmu

**Prednášajúci:**

**doc. Ing. Anna Jadlovská, PhD.**

**doc. Ing. Ján Jadlovský, CSc.**

# Metódy algoritmizácie

## Poznáme 3 metódy algoritmizácie

1. metóda "zhora nadol"
2. metóda "zdola nahor"
3. hybridná metóda

### 1. metóda "zhora nadol"

Algoritmus je zápis postupu, ktorý je použiteľný pre riešenie určitej triedy problémov. Ak postup riešenia rozkladáme na jednoduchšie operácie, až dospejeme k elementárnym krokom, potom tento postup návrhu algoritmu označujeme metóda "zhora nadol".

# Metódy algoritmizácie

## Príklad - Kvadratická rovnica

Na základe navrhnutého algoritmu budeme chcieť napísať program, ktorý bude riešiť rovnice, ktorých koeficienty sú uložené v súbore A. Výsledok sa má uložiť do súboru B.

Najhrubšia formulácia môže mať tvar:

Pokiaľ nenarazíte na koniec súboru A riešte rovnice určené koeficientmi uloženými v súbore A a výsledky zapisujte do súboru B.

Je nutné upresniť význam: "riešte rovnice, určené..." - rozklad na jednoduchšie kroky“

# Metódy algoritmizácie

## Príklad - Kvadratická rovnica

### Algoritmus hrubého riešenia:

- K1. Vykonaj úvodné operácie.
- K2. Pokiaľ nenarazíš na koniec súboru A, opakuj kroky 3-5, potom prejdi na krok 6.
- K3. Prečítaj zo súboru koeficienty  $a$ ,  $b$ ,  $c$ .
- K4. Vyrieš kvadratickú rovnicu s koeficientmi  $a$ ,  $b$ ,  $c$ .
- K5. Zapiš výsledky do súboru B a vráť sa na krok 2.
- K6. Vykonaj záverečné operácie.

Kroky K1, K4, K6 sa dajú spresniť.

Budeme sa zaoberať iba krokom K4 na riešenie KR.

## Metódy algoritmizácie

### Príklad - Kvadratická rovnica

#### Spresnenie kroku K4:

K4a. Vypočítaj  $d = b^2 - 4ac$

K4b. Ak  $d < 0$ , prejdí na K4f , inak pokračuj krokom K4c

K4c. Prirad'  $d = \sqrt{d}$

K4d. Rovnica má korene  $x_{1,2} = \frac{-b \pm d}{2a}$

K4e. Pokračuj krokom K5

K4f. Prirad'  $d = \sqrt{-d}$

K4g. Rovnica má korene  $x_{1,2} = \frac{-b \pm id}{2a}$

$i$  je imaginárna jednotka

Poz. Je potrebné upresniť postup, ak nastane chyba pri otvorení súboru, prípadne obsahuje chybné údaje,... formát vstupných a výstupných údajov.

# Metódy algoritmizácie

## 2. Metóda "zdola nahor"

Pri tejto metóde postupujeme od elementárnych krokov - vytvárame prostriedky, ktoré umožnia vyriešiť daný problém.

## 3. Hybridná metóda

Je kombináciou predošlých dvoch metód (vytváranie programov CASE).

# Základné zložky algoritmov - základné riadiace štruktúry

V algoritmoch sa stretávame s tromi základnými konštrukciami, kt. označujeme:

- a) postupnosť (sekvencia)
- b) vetvenie (selekcia)
- c) cyklus (iterácia)

## a) Postupnosť (sekvencia)

Je najjednoduchšou riadiacou štruktúrou. Je tvorená jedným alebo niekoľkými krokmi, kt. sa vykonávajú práve raz v danom poradí (príklad KR: kroky K4c-K4e).

So sekvenciou sa stretávame pri globálnom popise algoritmu - pri upresnení - zložitejšie konštrukcie!

# Základné zložky algoritmov - základné riadiace štruktúry

## b) Vetvenie (selekcia)

Štruktúra, ktorá v procese vykonávania algoritmu umožňuje vyberať rôzne alternatívy riešenia podľa splnenia či nespĺnenia zadaných podmienok.

Vetvenie môže byť realizované ako:

*I. Preskok*

*II. Dvojcestné vetvenie*

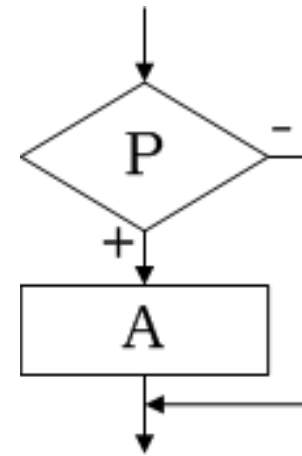
*III. Prepínač*



# Základné zložky algoritmov - základné riadiace štruktúry

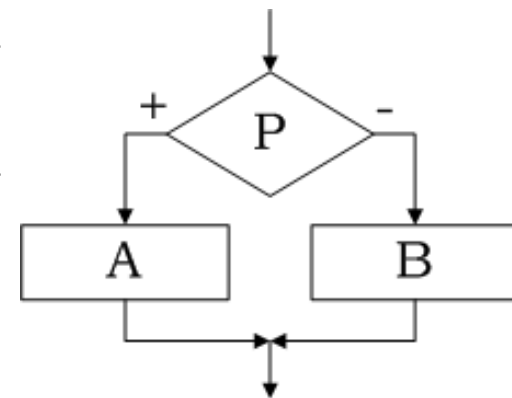
## I. Vetvenie - preskok

Po vyhodnotení podmienky sa uvedená činnosť (sekvencia operácií) buď vykoná alebo sa preskočí (príklad KR - krok K4b)



## II. Vetvenie - dvojcestné

Po vyhodnotení log. podmienky sa vykoná jedna z uvedených činností. Ak je pod. splnená, potom sa vykoná sekvencia príkazov tvoriacich činnosť A, v opačnom prípade sa vykoná sekvencia príkazov B. Po ukončení niektorej z alternatív sa pokračuje v algoritme. Viaccestné vetvenie - vnorenie dvojcestných vetvení.



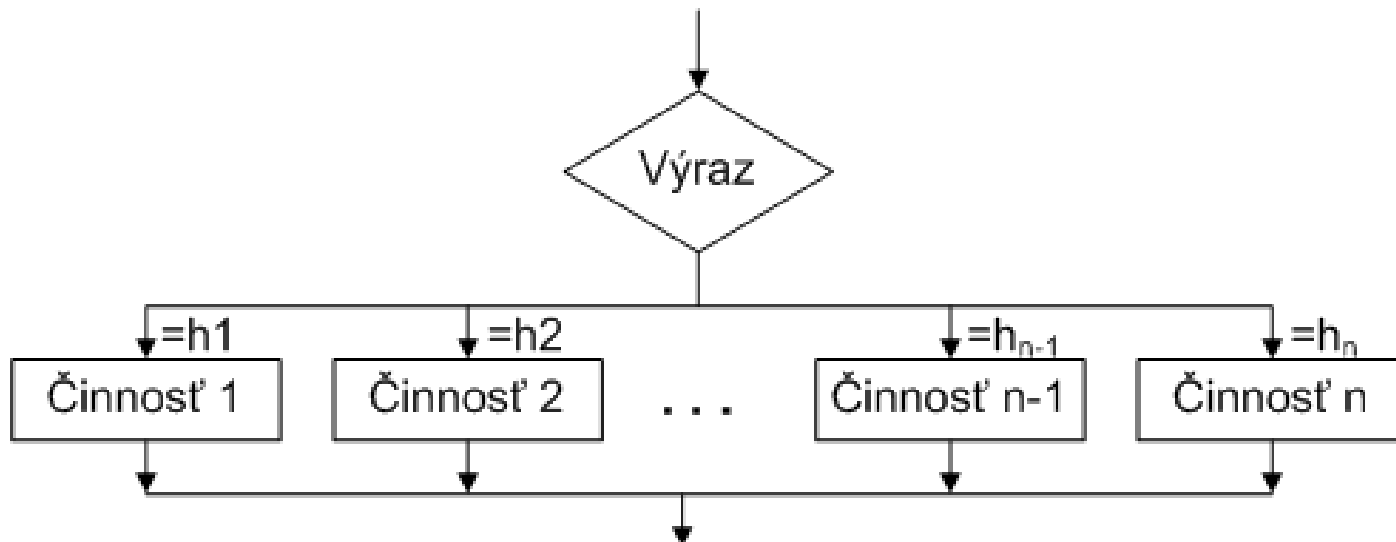
# Základné zložky algoritmov - základné riadiace štruktúry

## III. Vetvenie - prepínač

Je modifikáciou viaccestného vetvenia.

Pri viaccestnom vetvení - každá z vetiev je špecifikovaná pre celú množinu hodnôt podmienky.

Pri prepínači - je každá vetva určená pre konkrétnu hodnotu výrazu uvedeného v podmienke.



# Základné zložky algoritmov - základné riadiace štruktúry

Použitie vetvení znamená zaradenie rozhodovacieho procesu do vykonávania algoritmu a súvisí s vyhodnotením podmienky.

## Príklad - Kvadratická rovnica

VD - rozhodovanie



obr. Hrubý VD riešenia kvadratickej rovnice

# Základné zložky algoritmov - základné riadiace štruktúry

## c) Cyklus (iterácia)

Predstavuje časť algoritmu, ktorá sa opakuje pokiaľ je splnená podmienka opakovania.

Cyklus sa skladá z podmienky opakovania a tela cyklu.

Poznáme 3 druhy cyklu:

*I. cyklus s podmienkou na začiatku (v jazyku C: while, for)*

*II. cyklus s podmienkou na konci (príkaz jazyka C: do-while)*

*III. cyklus s daným počtom opakovaní (aritmetický cyklus) (v jazyku C: for)*

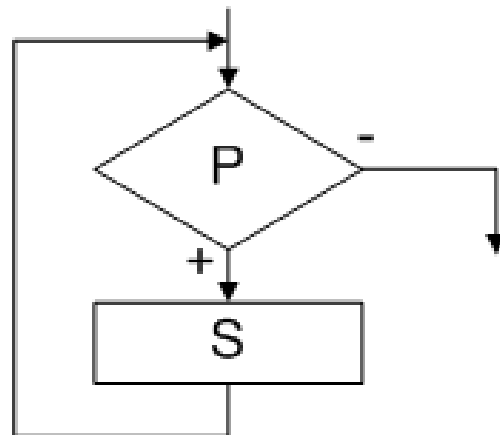
I. a II. sú logické cykly

V príklade riešenia kvadratickej rovnice tvoria kroky 2-5 cyklus s podmienkou na začiatku.

# Základné zložky algoritmov - základné riadiace štruktúry

## I. Cyklus s podmienkou na začiatku

1. Vyhodnotenie podmienky P
2. Ak je podmienka P splnená vykonajú sa operácie dané štruktúrou S (telo cyklu) a nasleduje návrat na začiatok cyklu
3. Ak podmienka P nie je splnená cyklus končí.

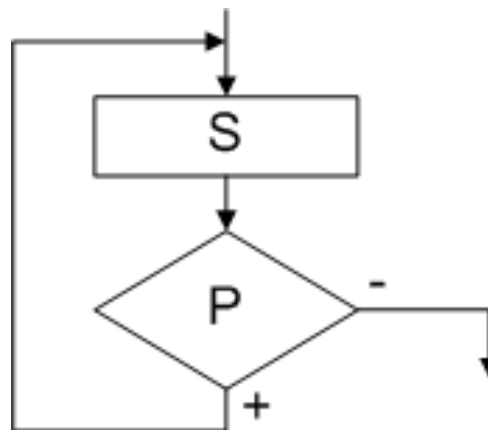


Pozn. cyklus sa nemusí vykonať ani raz

# Základné zložky algoritmov - základné riadiace štruktúry

## II. Cyklus s podmienkou na konci

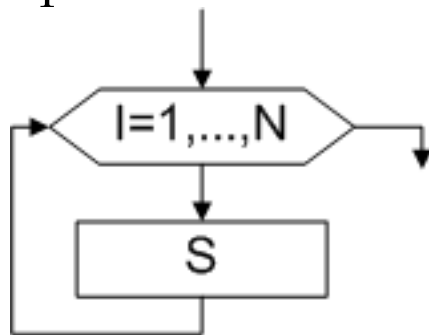
1. cyklus začína svoju činnosť operáciami v sekvencii S
2. ak podmienka P je splnená nasleduje návrat na bod 1 (začiatok tela cyklu)
3. ak podmienka P nie je splnená vykonávanie (logického) cyklu je ukončené (cyklus sa vykoná vždy aspoň raz oproti prípadu I.)



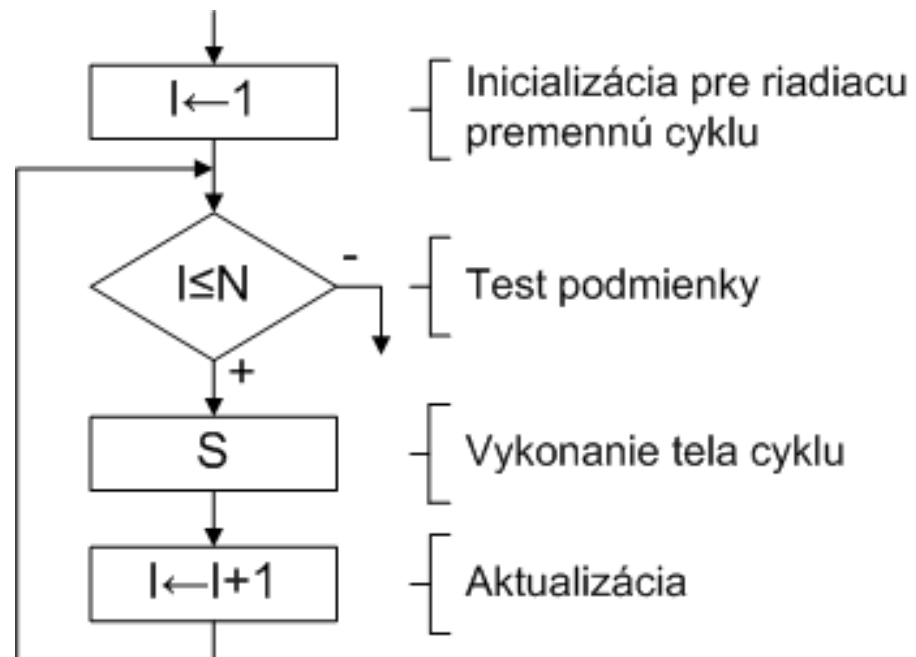
# Základné zložky algoritmov - základné riadiace štruktúry

## III. Cyklus s daným počtom opakovaní

Poznáme nasledujúce 2 typy cyklu s daným počtom opakovaní:



Aritmetický cyklus s modifikačným blokom



Logický cyklus (while)