Počítače a algoritimizácia

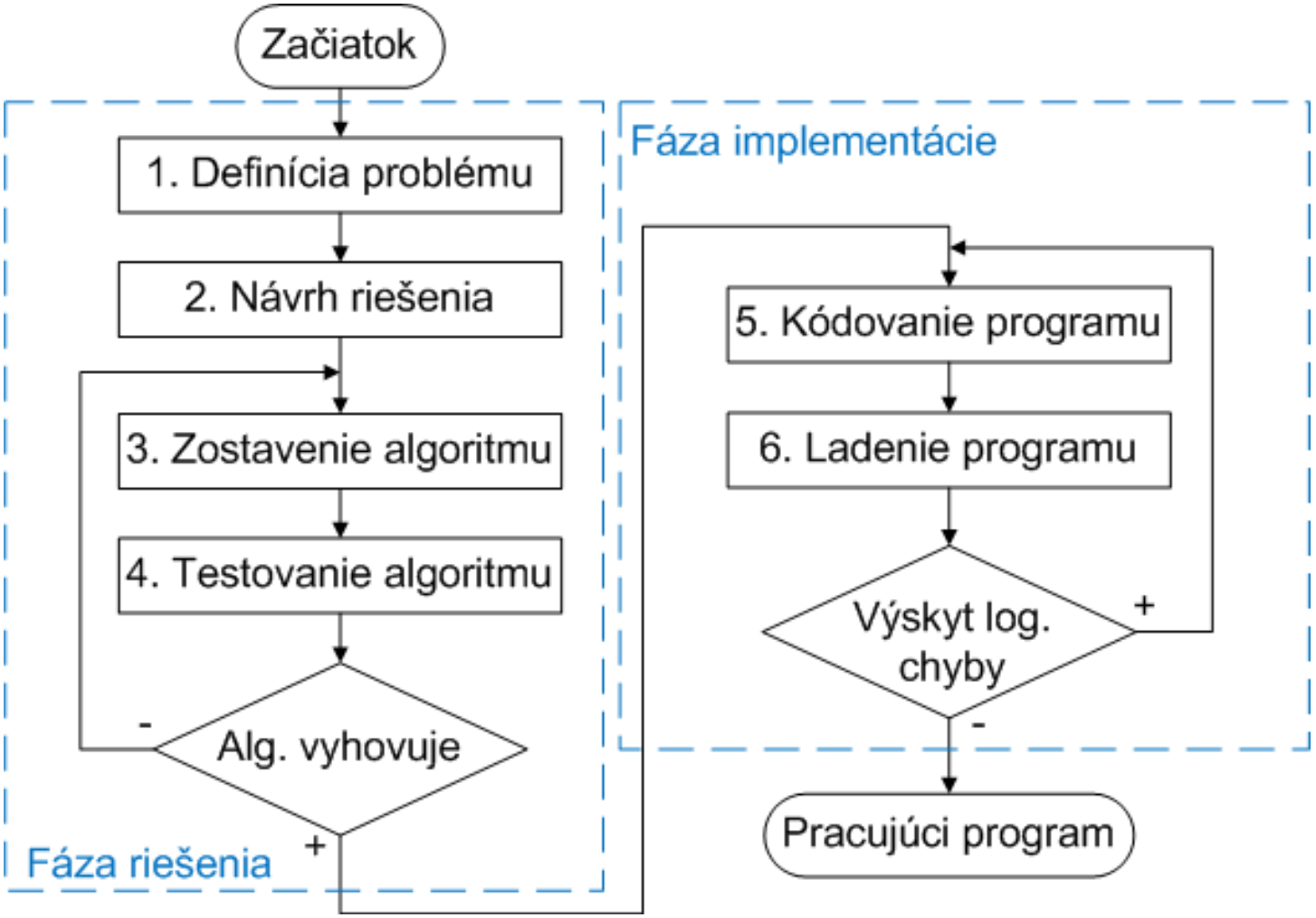
Postup riešenia úlohy na počítači

Prednášajúci:
doc. Ing. Anna Jadlovská, PhD.
doc. Ing. Ján Jadlovský, CSc.

Hlavné etapy postupu riešenia

1. Formulácia úlohy (definícia problému)
2. Analýza úlohy (návrh riešenia)
3. Nájdenie algoritmu výpočtu (zostavenie algoritmu)
4. Overenie správnosti algoritmu
5. Kódovanie programu a preklad
6. Overenie správnosti programu (ladenie programu)

Etapy riešenia úloh na počítači



1. Formulácia úlohy

Pre správne riešenie úlohy na počítači je potrebné definovať, čo máme riešiť (jednoznačná identifikácia úlohy a ujasnenie cieľa).

Podľa zložitosti problému môže byť definícia problému slovná alebo s využitím formálneho aparátu.

Formulácia úlohy – úprava úlohy na taký tvar, ktorý je vhodný pre počítač.

Formulácia má spĺňať nasledujúce podmienky:

a) diskrétnosť údajov a operácií
b) konečnosť údajov a operácií
c) obmedzenia dané počítačom

2. Analýza úlohy – návrh riešenia

Zisťuje sa, či úloha je riešiteľná, či má jedno alebo viac riešení, načrtávajú sa možnosti riešenia a rozhoduje sa o druhu metód.

Kroky analýzy:

a) vykonať dôkladný rozbor úlohy (problému)

b) špecifikovať vstupy (štruktúra, formát, množinu prípustných hodnôt)

c) špecifikovať výstupy (štruktúra, formát, funkčná závislosť na vstupných hodnotách)

d) výber metódy, dekompozíciu úlohy na čiastkové problémy

3. Nájdenie algoritmu výpočtu - algoritmizácia

Realizácia úlohy na počítači predpokladá poznať presný postup výpočtu – algoritmus.

Algoritmus je presný predpis definujúci výpočtový proces, ktorý vedie od meniteľných východzích údajov až k správnym odpovedajúcim výsledkom (Markov).

Algoritmus sa skladá z jednotlivých výpočtových krokov, ktoré sú zapísané v presne určenom poradí, ich počet musí byť konečný.

Vlastnosti algoritmov

Každý algoritmus má 3 základné vlastnosti:

a) Hromadnosť – algoritmus definuje výpočet celej triedy úloh rovnakého typu, pričom sa líšia od seba vstupnými údajmi

b) Determinovanosť – algoritmus je presný a jednoznačný, čo znamená, že vždy je určený nasledujúci krok jednoznačným spôsobom. Výpočet sa dá vždy opakovať (pre rovnaké vstupné údaje dostaneme vždy rovnaký výsledok).

c) Rezultatívnosť – udáva, že pre prípustné vstupné hodnoty získame výsledok po konečnom počte výpočtových krokov.

Nájdenie algoritmu riešiaceho daný problém nazývame algoritmizácia (pri algoritmizácii sa vyžaduje schopnosť logického myslenia).

Pozn. Pojem algoritmizácia pochádza od arabského matematika M.M.Abdallah al Chorezmiho (8/9 st. n.l.)

Algoritmus

Vstup pojmu algoritmus do novodobej terminológie zaznamenala monografia ruského matematika Markova *Teória algoritmov*.

Neformálne vymedzenie pojmu algoritmus:

I. mať problém, ktorý chceme riešiť

a) presne definovaný

b) v priebehu riešenia nemenný

c) rozpoznateľné riešenie

II. mať nástroj na riešenie problému

a) nehmotný - matematický aparát

b) hmotný - počítač

Príklad – aritmetický priemer

1. Príklad:

   Navrhnite algoritmus pre výpočet aritmetického priemeru z daných troch čísel a overte jeho správnosť.

2. Riešenie:

   1. vstupné premenné: *A1 - prvé číslo*
      A2 - druhé číslo
      A3 - tretie číslo
   2. výstupné premenné: *PR - aritmetický priemer (výsledok)*
   3. pomocná premenná: *C*

Príklad – aritmetický priemer

3. Zostavenie algoritmu (slovný popis algoritmu)

K1. načítanie prvého čísla	*čítaj A1*
K2. odpamätaj prvé číslo do premennej C	*C ←A1*
K3. načítanie druhého čísla	*čítaj A2*
K4. vykonaj súčet prvého a druhého čísla a ulož do premennej C	*C ← C+A2*
K5. načítanie tretieho čísla	*čítaj A3*
K6. pričítanie tretieho čísla k výsledku a odpamätanie do C	*C ← C+A3*
K7. výpočet aritmetického priemeru	PR ← C/3
K8. KONIEC	

Príklad – aritmetický priemer

4. Overenie správnosti algoritmu

Spočíva v otestovaní algoritmu na vhodnej množine vstupných údajov (A1=1, A2=-4, A3=0).

Vhodným prostriedkom pre testovanie je simulačná tabuľka (viď cvičenia).

K1	čítaj A1	A1=1
K2	C ←A1	C = 1
K3	čítaj A2	A2 = -4
K4	C ← C+A2	C = 1-4 = -3
K5	čítaj A3	A3 = 0
K6	C ← C+A3	C = -3 + 0 = -3
K7	PR ← C/3	PR = -3/3 = -1
K8	KONIEC	

Príklad – aritmetický priemer

5. Kódovanie programu a preklad

Predstavuje prepis algoritmu do programovacieho jazyka (C-jazyk).

Program v príslušnom jazyku nazývame zdrojový program, ktorý tvorí vstup pre prekladač (kompilátor).

Kompilátor – súbor programov pre počítač, ktorý prijíma ako vstupné dáta program v problémovo-orientovanom jazyku a ako výstup vytvára počítačovo-orientovaný kód, (lexikálna analýza, syntaktická analýza – odstránenie formálnych chýb)

Algoritmus versus program

program - postupnosť príkazov (chránený autor. zákonom

algoritmus - postup práce (dá sa patentovať)

Program realizuje algoritmus, algoritmus je jeho nutnou súčasťou.

Príklad – aritmetický priemer

6. Overenie správnosti programu (a algoritmu)

Spočíva v testovaní programu na vhodnej množine vstupných údajov, pre ktoré poznáme výsledky.

Ladenie programu – odstránenie logických chýb

Ladiace prostriedky (debuger) – umožňujú sledovať priebeh výpočtu s medzivýsledkami.

Algoritmické jazyky

Použitie ľudskej reči k jednoznačnému a presnému vyjadreniu algoritmu - v praxi robí problémy - pre popis algoritmov (výpočtových) boli vyvinuté jazyky určené k vyjadrovaniu vymedzeného okruhu vybraných činností.

Algoritmické jazyky predstavujú súhrn prostriedkov a pravidiel, ktoré umožňujú vyjadrovať výpočtové algoritmy:

a) vývojové diagramy
b) štruktúrogramy
c) rozhodovacie tabuľky
d) programovacie jazyky

Algoritmické jazyky

a) Vývojové diagramy

Predstavujú normou definované symbolické značky a pravidlá pre ich používanie, ktoré slúžia pre jednoznačné grafické vyjadrovanie výpočtových operácií a postupov.

Zobrazujeme postupnosť krokov riešenia úlohy - popis algoritmu Λ podklad pre zostavenie programu pre počítač.

b) Štruktúrogramy

Sú obdobou VD - nie sú definované normou. Predstavovali pokus "zhutniť" grafickú interpretáciu výpočtového postupu. Neujali sa v praxi - praktické využitie bezvýznamné.

Algoritmické jazyky

c) Rozhodovacie tabuľky

Boli definované pre algoritmizáciu úloh so zložitým rozhodovaním v oblasti hromadného spracovania údajov.

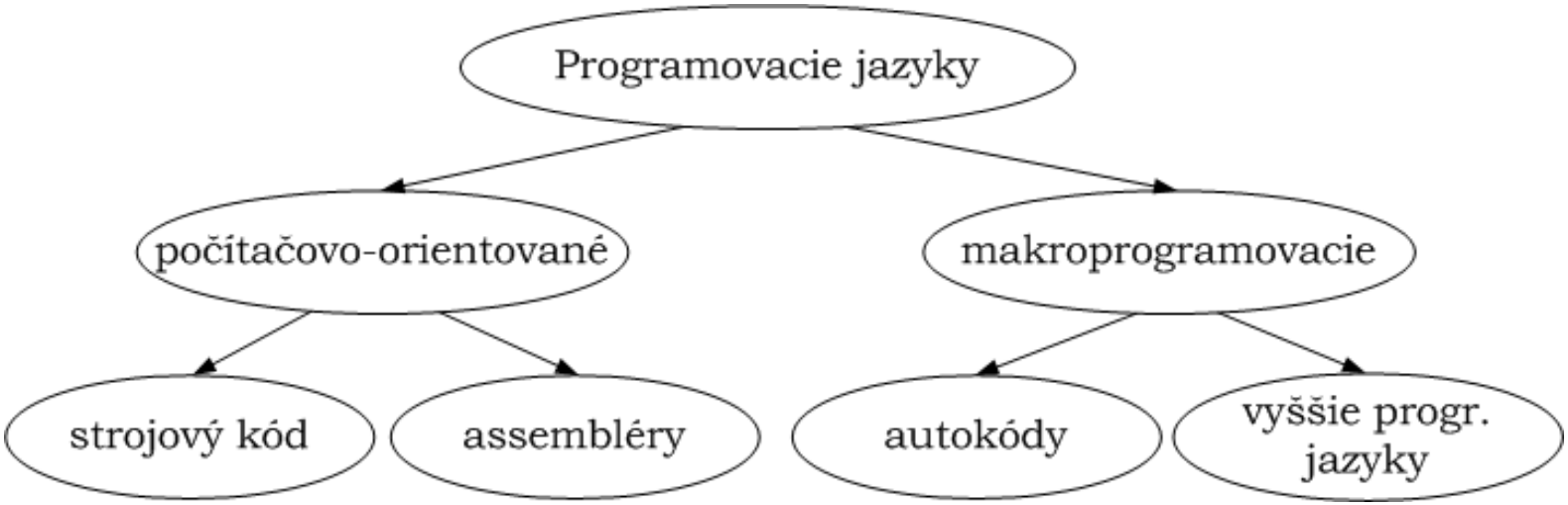
d)Programovacie jazyky

Sú dôsledne formalizované algoritmické jazyky, určené pre zápis algoritmu pre počítač.

Zápis algoritmu v programovacom jazyku – program.

Programovací jazyk vyhovuje vždy určitým pravopisným (syntaktickým) pravidlám s dopredu dohodnutým významom (sémantikou).

Programovacie jazyky



[C-jazyk, Pascal, Fortran, ...]

Assemblér – jazyk symbolických adries

Vyššie programovacie jazyky

Majú výrazne prepracovanú syntax a sémantiku.

Zahŕňajú v sebe zložité programované konštrukcie (štruktúry a objekty).

Kvôli všeobecnosti nie sú tieto jazyky viazané na konkrétny typ počítača *(jeden príkaz ~ 4-10 strojovým inštrukciám).*

Vývoj:

- prírodné vedy (Fortran, Algol 60)
- spracovanie hromadných údajov (RPG,COBOL)
- zobecnenie (PL/1, Algol68, ada)

Programovacie jazyky vyš. typu (Pascal-výuka programovania, tzv. štruktúrované programy.

Simulačné jazyky (SIMULA,CSMP, Matlab/Simulink).

Vývojové diagramy

VD je symbolický algoritmický jazyk, ktorý sa používa pre názorné zobrazenie algoritmu, t.j. znázorňuje logickú stavbu programu pre systém spracovania informácií (tok údajov).

Tento jazyk je tvorený:

a) značkami s presným významom (sémantika – slovník)
b) pravidlami ako značky používať vo vzájomnej súvislosti (syntax – gramatika)

Predstavuje grafické znázornenie logickej štruktúry riešeného problému

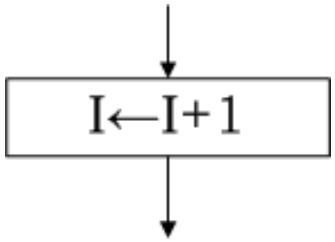
Vývojový diagram:

- rieši problém vo všeobecnej rovine bez zreteľa na vlastnosti počítača a programovacieho jazyka
- rieši problém – priamy vzťah k počítaču a hlavne k možnostiam a vlastnostiam pr. jazyka.

Symboly vývojových diagramov

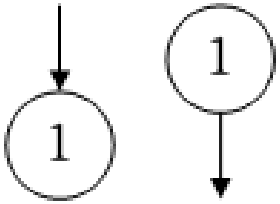
Príkaz priradenia

Priradenie hodnoty výrazu na pravej strane do premennej na ľavej strane. Typ premennej a výrazu musia byť kompatibilné.
Čítame: Premennej I priraď hodnotu výrazu I+1



Spojka - pokračovanie

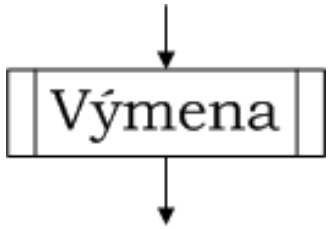
Predstavuje prechod z jednej časti VD na inú časť



Symboly vývojových diagramov

Vopred definovaná činnosť (podprogram)

Samostatný úsek programu – predstavuje skupinu činností, ktoré sú špecifikované inde a v danom VD nie sú rozpracované.



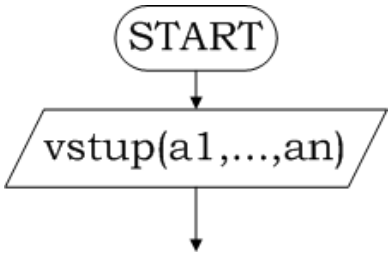
Medzná značka

Označenie medzného bodu vývojového diagramu (začiatok, koniec)

Symboly vývojových diagramov

Príkaz vstupu

Tvar vstup(a1,..,an) - zo vstupného zariadenia sa prečíta n hodnôt a postupne sa priradia premenným a1,..,an.



Príkaz výstupu

Tvar výstup(b1,..,bn) – vytlačia sa výstupné údaje b1,..,bn