

algoritmizácia a programovanie

Algoritmus.

Návod, ktorý určuje postup vedúci k riešeniu danej úlohy, nazývame **ALGORITMUS**.

- Je základným pojmom informatiky. Je to zrozumiteľný, jednoznačný, presný opis postupu riešenia danej úlohy a úloh rovnakého typu.
- Je to logický sled pravidiel, ktorý vyrieši danú úlohu (pravidlo = inštrukcia, príkaz).
- Je to postup, realizáciou ktorého získame v konečnom čase zo zadanych vstupných údajov správne výsledky.
- Je to predpis na získanie určitého výsledku pomocou postupnosti krokov, ktoré vykonávame v dopredu stanovenom poradí.
- Je to postupnosť krokov, ktoré definujú postupnú transformáciu vstupných údajov na výstupné.

ALGORITMUS je presne definovaná konečná postupnosť pravidiel, ktorých realizácia pre prípustné vstupné údaje nám umožňuje získať po konečnom počte krokov príslušné správne výstupné údaje.

Vlastnosti – požiadavky na algoritmus:

- **elementárnosť** – postup je zložený z jednoduchých krokov, ktoré sú pre vykonávateľa (počítač, nemysliace zariadenie, človeka) zrozumiteľné,
- **determinovanosť** – postup je zostavený tak, že v každom momente jeho vykonávania je jednoznačne určené, aká činnosť má nasledovať, alebo či sa už postup skončil,
- **rezultatívnosť** – výpočet dáva po konečnom počte krokov výsledok, a to **správny výsledok** pri riešení ľubovoľnej zo skupiny úloh, pre ktorú bol vytvorený – algoritmus musí riešiť zadaný problém
- **konečnosť** – výpočet (činnosť vykonávaná podľa algoritmu) vždy skončí po vykonaní konečného počtu krokov,
- **hromadnosť** – algoritmus je použiteľný na celú množinu prípustných vstupných údajov,
- **efektívnosť** – výpočet sa uskutočňuje v čo najkratšom čase a s využitím čo najmenšieho množstva prostriedkov (časových i pamäťových).

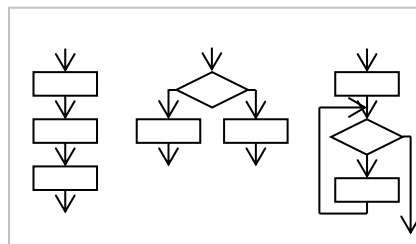
Tvorca algoritmu – človek

Realizátor – vykonávateľ – človek alebo počítač. Pri počítači je veľmi dôležitý presný a jasný postup, pretože počítač nemá všeobecné znalosti ako človek. Počítač vie vykonať len také príkazy, ktoré má naprogramované

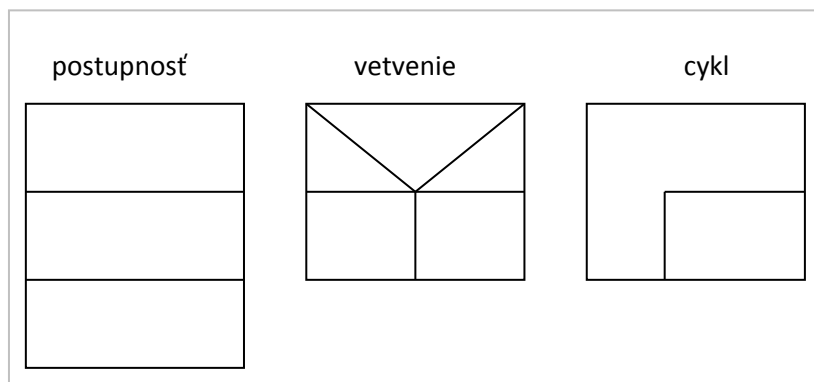
Spôsoby zápisu algoritmov:

1) Graficky orientované

- vývojové diagramy, znázorňujú rôzne typy príkazov rôznymi tvarmi blokov



- štruktúrogramy, znázorňujú rôzne typy príkazov konštrukciami:



- obrázkové jazyky.

2) Textovo orientované

- programovacie jazyky,
- slovný zápis algoritmu v prirodzenom jazyku,

Etapy algoritmicizácie úloh

Ak chceme riešiť akúkoľvek úlohu na počítači, treba ju rozdeliť na celý rad prípravných prác – etáp. Algoritmicizácia úloh má tri základné etapy:

- formulácia úlohy
- analýza úlohy
- zostavenie riešacieho algoritmu

Formulácia úlohy

Prvým predpokladom, aby sme danú úlohu mohli riešiť na počítači, je jej jasná a jednoznačná formulácia, ujasnenie cieľa, ktorý pri riešení príslušnej úlohy sledujeme. Treba formulovať problém, napr. matematickými prostriedkami.

Analýza úlohy

V tejto etape je potrebné nájsť algoritmus riešenia úlohy. Zisťuje sa, či úloha je riešiteľná, či má jedno alebo viac riešení, načrtávajú sa možnosti riešenia.

Zostavenie riešacieho algoritmu

Po správnej formulácii a analýze úlohy nasleduje etapa činnosti, v ktorej sa popíše logika a postup riešenia úlohy. Výsledkom tejto etapy je riešiaci algoritmus. Do tejto etapy už môžeme zahrnúť aj naprogramovanie úlohy v konkrétnom programovacom jazyku.

Pod pojmom **programovanie** rozumieme činnosť, pomocou ktorej sa uskutočňuje prevod nápadu riešenia do formy vhodnej pre spracovanie na počítači. Výsledkom tejto činnosti je program.

Program je algoritmus v takej forme, ktorej rozumie počítač, t. j. program je zápis algoritmu v niektorom programovacom jazyku.

KONTROLA SPRÁVNOSTI ALGORITMU:

1. **TABUĽKA** — do ktorej zapisujeme presné hodnoty premenných, ktoré sa v programe vyskytujú, tak ako sa počas vykonávania jednotlivých krokov programu menia
2. **LADENIE PROGRAMU** — algoritmus prepísaný do programovacieho jazyka a opakovane spúšťaný na počítači vždy s novými údajmi. K jednotlivým vopred pripraveným skupinám vstupných údajov musia byť pripravené aj výsledky, t.j. predpokladané výstupy.
3. **INVARIANTY** — matematické dokazovanie platnosti algoritmu

Chyby v programoch môžu byť:

- syntaktické – nesprávny zápis
- logické – program nedáva správny výsledok
- run time - program „spadne“ počas vykonávania

Programovací jazyk je kompromis:

Programátor v ňom vie ľahko a dobre navrhovať algoritmy, počítač mu zas dokáže porozumieť vďaka špeciálnemu programu, ktorý sa nazýva **prekladač-kompilátor**. Je to program, ktorý prekladá zápis algoritmu z programovacieho jazyka do tvaru vykonateľného pre počítač (binárneho kódu).

Každý programovací jazyk má kľúčové slová, príkazy, rezervované slová a podobne. Rôzne programovacie jazyky majú rôznu syntax (odlišný zápis príkazov).

Čo je ťažšie?

1. *Vymyslieť algoritmus alebo*
2. *prepísať hotový algoritmus do konkrétného programovacieho jazyka?*

Ak ste si pomysleli, že 1. tak máte pravdu.

Programovací jazyk je len zoznam príkazov, ktoré v každom jazyku majú rôznu syntax, v každom jazyku je príkaz vstupu, výstupu, priradenie, vetvenie, cyklus a podobne. To sa dokáže naučiť každý. Ale vymyslenie algoritmu je v programovaní práve tá najťažšia etapa

Prečo sa budeme učiť programovací jazyk **Pascal v prostredí Delphi**? Bol vyvinutý práve na učiace sa účely. Žiak sa v ňom naučí základné programátorské konštrukcie, celú logiku programovania, veľmi jednoduchým efektívnym spôsobom. Kto zvládne Pascal, nebude mať v budúcnosti problém prejsť veľmi rýchlo aj k inému programovaciemu jazyku. Uvedomte si algoritmus bude rovnaký, len syntax bude iná, asi trochu náročnejšia.

Úlohy:

1. Kde všade sa v bežnom živote stretávame s algoritmami?
2. Navrhnite algoritmus, formulujte ho slovne, ako sa z vrátnice dostanem do riaditeľne.

3. Evička chce ísť na pizzu. Má našetrené peniaze. Popíšte slovne, ako bude postupovať pri výbere pizze. Nakreslite pomocou vývojového diagramu.
4. Janičko má vedro, ktoré má objem 10 litrov. Z domu k potoku je 98 metrov. Chce naplniť 150 litrový sud vodou. Popíšte slovne, ako má postupovať! Nakreslite pomocou vývojového diagramu.
5. Pomocou vývojového diagramu skúste načrtnúť riešenie nejakého matematického problému.
6. Pomocou vývojového diagramu načrtnite postup začiatku hry „človeče nehnevaj sa“, postavenie panáčka na štart.

.