

Etapy riešenia problému – algoritmus

Tvorbou programov sa zaoberá **softvérové inžinierstvo**. Zaoberá špecifikovaním, návrhom, vývojom a údržbou softvéru s využitím poznatkov informatiky, projektového manažmentu a ďalších oblastí.

Vývoj softvéru prebieha v niekoľkých etapách:

- rozbor problému,
- vytvorenie algoritmu (návrh),
- vytvorenie programu (implementácia),
- testovanie a ladenie programu (preverenie funkčnosti a výkonu)

Rozbor problému

Problém je stav, v ktorom jestvuje rozdiel medzi tým, čo v danom momente máme a tým, čo chceme dosiahnuť.

Riešenie problému chápeme ako odstraňovanie rozdielu medzi aktuálnym stavom a tým, čo chceme dosiahnuť.

Rozbor problému spočíva v **presných** odpovediach, **čo** treba riešiť. Treba sformulovať zadanie problému a požiadavky na vznikajúci program (vstupné a výstupné hodnoty, podmienky platnosti programu).

Pozostáva z:

- *rozdelenie problému na menšie podproblémy a tie na elementárne problémy*
- *špecifikácie vstupných dát (štruktúra, formát, množina prípustných hodnôt)*
- *špecifikácie výstupných dát (štruktúra, formát, funkčná závislosť od vstupných dát)*
- *špecifikácie výnimočných situácií (napr. reakcia na chybné vstupné dáta)*

Vytvorenie algoritmu

Jednoznačný zrozumiteľný postup umožňujúci z daných vstupných údajov získať v konečnom čase po vykonaní konečného počtu krokov správne výsledky - riešenie danej úlohy.

Je to presne definovaná, konečná postupnosť príkazov, ktorých vykonanie umožní po konečnom počte krokov získať pre prípustné vstupné údaje zodpovedajúce výstupné údaje.

Pod algoritmom rozumieme predpis na spracovanie, ktorý je formulovaný natoľko presne, že ho môže realizovať mechanické alebo elektronické zariadenie.

Príkladmi algoritmov sú predpisy na sčítanie, odčítanie, násobenie a delenie čísel, Euklidov algoritmus na výpočet najväčšieho spoločného deliteľa. Algoritmy tiež pripomínajú kuchárske recepty, návody pre domácich majstrov a podobne. Tieto však nie sú definované natoľko presne, aby sa dali realizovať pomocou stroja.

Vlastnosti algoritmu

- **jednoznačnosť (determinovanosť)** - v každom momente vykonávania musí byť jednoznačne určené, ktorá činnosť bude realizovaná ako nasledujúca,
- **konečnosť (rezultatívnosť)** - postup skončí v konečnom čase po vykonaní konečného počtu malých krokov, kedy sa získajú správne výsledky,
- **všeobecnosť (hromadnosť)** - postup musí byť použiteľný pre celú množinu vstupných údajov (musí riešiť problém všeobecne a nesmie sa zúžiť na riešenie niektorých špeciálnych prípadov),
- **opakovateľnosť** - viackrát pri rovnakom vstupe musí byť aj rovnaký výstup,
- **zrozumiteľnosť a prehľadnosť.**
- **časová a pamäťová náročnosť**

Algoritmus z hľadiska programovania môže byť vyjadrený

- graficky - pomocou vývojových diagramov alebo štruktúrogramov
- slovne - pomocou prirodzeného jazyka alebo formálneho jazyka,
- matematicky - riešenie sústavy rovníc, matice, determinanty, vzťahy medzi premennými, ...
- programovacím jazykom - Pascal, C#, C, ...

Algoritmizácia

Činnosť na zostavenie algoritmov tak, aby sa jednotlivé elementárne problémy dali vykonať pomocou stroja. Riešenie problému sa však zatiaľ nevzťahuje na žiadny konkrétny stroj alebo systém.

Vytvorenie programu

Program je algoritmus zrealizovaný pomocou, konkrétneho programovacieho jazyka. Podľa požiadaviek sa realizujú dátové štruktúry pre vstupné a výstupné údaje, riadiace štruktúry, ktoré údaje spracúvajú a používateľské rozhranie, pomocou ktorého používateľ program ovláda. Program sa zostavuje pomocou exaktne definovanej sady príkazov operátorov a syntaxe, ktorú je možné previesť do tvaru vykonateľného počítačom.

Ladenie a testovanie

Testovaním zisťujeme, či nie sú v danom programe logické chyby, testovanie väčších programov robíme pomocou trasovacej tabuľky (vloženie kontrolných výpisov - krokovanie).

Proces, počas ktorého zistené chyby hľadáme a odstraňujeme sa nazýva ladenie. Okrem odstraňovania chýb však ladenie slúži aj na optimalizovanie kódu. To znamená, že sa pokúšame dosiahnuť, aby sa program vykonával rýchlejšie, aby spotreboval menej pamäte, alebo aby jeho chod bol plynulý.

Vyladiť môžeme program:

- krokovaním – spúšťanie programu po riadkoch, príkazoch
- kontrolou obsahu premenných
- zastavením programu na zvolenom mieste

Logické chyby:

- zistené počas alebo po skončení behu programu:
vedúce k predčasnemu zastaveniu programu s chybovou správou (run time errors) napr.: delenie nulou, súbor na otvorenie nenájdenny, výpočet väčšieho čísla, aké sme si zadefinovali a pod.
- vedúce k chybným výsledkom,
zacykleniu programu (zastavenie vykonávania programu: Run - Program Reset) a pod.