

RECENZIE

Niklaus Wirth, *Systematic programming. An introduction*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey 1973, 164 strán

Autor hneď v úvode hovorí, že jeho cieľom je podať programovanie ako umenie, alebo techniku vytvárania a formulovania algoritmov systematickým spôsobom. Algoritmy sa majú vytvárať ako pevné štruktúry, ktoré sa skladajú z logicky, spoľahlivo a vhodne zostavených konštrukčných blokov.

Rád by som hneď na začiatku zdôraznil, že autorovi sa jeho cieľ podaril a jeho kniha je novátorským a veľmi cenným metodickým prínosom do problematiky vyučovania programovania. Myslím, že táto kniha by bola vhodným základom, alebo by sa dala aj priamo použiť ako učebnica programovania pre stredné školy a, prirodzene, pre úvodné ročníky vysokých škôl.

V knihe nie sú zbytočné detaily, čo býva dosť časté v knihách o programovaní. Cieľom autora je vyabstrahovať základné pojmy a techniky a podať ich jednoduchým spôsobom. To sa mu výborne podarilo. V knihe ide o podstatu vytvárania algoritmov. Existujúca výpočtová technika a programovacie jazyky sú tu v úzadí a používajú sa len na ilustráciu základných pojmov, ako možné realizácie procesorov a programovacích jazykov. Cieľom knihy je vychovať čitateľa, aby postupoval metodicky a systematicky pri vytváraní algoritmov. Tento cieľ sa autor snaží doceliť tým, že demonštruje v knihe problémy a techniky, ktoré sú typické pre programovanie, ale nezávisia od konkrétnej oblasti aplikácií. Žiadne konkrétne označovanie alebo programovací jazyk sa nezdôrazňujú. Autor však upozorňuje, že cieľom základných prednášok z programovania nie je doceliť dokonalosť v používaní všetkých prvkov a zvláštností konkrétneho jazyka. Používaný jazyk má len zachytávať základné a najdôležitejšie pojmy o algoritmoch prirodzeným a ľahko zrozumiteľným spôsobom. Na ilustráciu sa v knihe používa jazyk, ktorý je blízky jazyku **algol 60**.

Aj keď kniha je len úvodom do systematického programovania, je svojím obsahom a spracovaním na vysokej úrovni. Po tejto stránke je to môjím prvá kniha o programovaní, ktorú možno zaradiť vedľa učebníc matematiky, fyziky, chémie atď. Zavádzanie nových pojmov je presvedčivé a elegantné. Autor hneď na začiatku zavádza Floydovu analytickú techniku verifikovania správnosti programov a používa ju dosť systematicky v priebehu knihy. Autor zdôrazňuje aj potrebu verifikovania či výpočet je konečný podľa daného programu. Na vhodných príkladoch ilustruje autor potrebu analýzy algoritmov z hľadiska výpočtovej zložitosti. Programové schémy sa používajú na výklad základných programovacích techník. Štruktúry dát sú zavedené systematicky a elegantne. Značný počet starostlivo vybratých príkladov pomáha jednak pochopiť základné pojmy a techniky, jednak slúži autorovi na ilustráciu tých prvkov, ktoré bezprostredne súvisia s existujúcou výpočtovou technikou.

WIRTH neustále zdôrazňuje, že prítomnosť a využitie štruktúry je základným prostriedkom, ktorý pomáha programátorovi syntetizovať systematicky a udržiavať celkové pochopenie zložitých programov.

Od čitateľa sa predpokladá len znalosť elementárnej matematiky a elementov výrokovej logiky. Niektoré príklady, ktoré však možno vynechať, sú vzaté z analýzy. Predpokladá sa však, že ak sa kniha používa ako učebnica, študenti budú veľmi aktívne používať počítač na

realizáciu rôznych algoritmov. Autor veľmi zdôrazňuje, že pre úspech prednášok z programovania je nevyhnutné, aby študenti mali ľahký prístup k moderným počítačom a aby 15 až 30 minút po odovzdaní programu bol program spracovaný.

Vzhľadom na novátorský prístup k problematike a vzhľadom na to, že považujem knihu za dobrý vzor učebnice, uvediem obsah knihy podrobnejšie. Treba pritom poznamenať, že kniha má len 130 strán textu, vrátane rozpracovaných príkladov, ktorý je vo veľmi prehľadnej úprave, 23 strán cvičení pre čitateľa a dva krátke dodatky: syntaktické diagramy jazyka **pascal** a tabuľka symbolov v ASCII kóde.

V úvodnej kapitole sa zdôrazňuje, že aj keď je programovanie disciplína s mnohými aplikáciami, predsa základné idey vytvárania algoritmov sa dajú vysvetliť a pochopiť bez akejkoľvek referencie na počítače.

V druhej kapitole o základných pojmoch sa vychádza z pojmu akcie, ktorej účinkom je zmena stavu nejakých objektov. Ak je akcia rozložiteľná do častí, hovoríme o *procese* a popis procesu nazývame *programom*. Procesory realizujú procesy podľa programov. Tieto a ďalšie pojmy, ako napr. *algoritmus ako schéma chovania procesu*, *premenná*, *priradovací príkaz* atď., sa ilustrujú na príkladoch algoritmov pre násobenie a delenie prirodzených čísel za predpokladu, že procesor nemá operácie násobenia a delenia. Vysvetľuje sa rozdiel medzi algoritmom a programom.

V ďalšej kapitole 3 je veľmi stručný opis (na 4 stranách) tých *najzákladnejších charakteristík počítačov* — procesor, pamäť, bunka, adresa, uloženie programu do pamäti — a zakódovanie objektov do binárnej formy.

V štvrtej kapitole sú základné pojmy z oblasti *softwaru*: strojový kód, programovacie jazyky vyššieho typu, kompilátor, software.

Prvou rozsiahlejšou kapitolou je piata kapitola, venovaná *jednoduchým programom*. Po zavedení pojmu blokového diagramu a cyklu zdôrazňuje autor potrebu analytických metód verifikovania korektnosti programov. V ďalšom autor uvádza Floydovu techniku a postuluje štyri základné pravidlá pre analytické verifikovanie programov. Autor zdôrazňuje ťažkosti pri používaní Floydovej techniky, najmä pri vytváraní hypotéz v jednotlivých bodoch programov v prípade cyklov a zdôrazňuje význam cyklových invariantov. Použitie analytických metód verifikovania programov je ilustrované na už vytvorených programoch.

Je nesporné, že analytické verifikačné metódy, práve tak ako analýza algoritmov, o ktorej je reč v ďalšom, dosť zvyšujú požiadavky na abstrakčné schopnosti a kombinatorické myslenie u študentov. Súhlasím však s autorom, že táto problematika patrí do základnej prednášky o programovaní, pretože tak verifikovanie, ako aj analýza programov, sú základom pre hlbšie porozumenie algoritmov. Ich praktický význam je značný a bez nich by programátor nemal iný prostriedok ako vlastnú, nedostačujúcu intuíciu.

V ďalšej veľmi stručnej kapitole 6 sa autor zaoberá základnou technikou ako určiť, že výpočet podľa daného programu je konečný.

Kapitola 7 začína stručnou históriou vývoja programovacích jazykov a tento úvod končí zavedením *syntaktických diagramov*, v ktorých je v dodatku, len na ilustráciu, opísaný jazyk **pascal**. Programovací jazyk sa v ďalšom prezentuje ako jazyk, ktorý má prostriedky na opis výrazov, má priradovací a zložený príkaz a príkazy IF, WHILE, REPEAT a CASE. Pre všetky tieto príkazy sú uvedené analytické verifikačné pravidlá. V tomto elementárnom programovacom jazyku sú potom zapísané algoritmy, ktoré sme uviedli už v predchádzajúcich kapitolách, a verifikuje sa ich správnosť. Dva algoritmy pre násobenie celých čísel sú analyzované, so zaujímavým a netriviálnym výsledkom, z hľadiska zložitosti výpočtu. Autor uvádza aj dva algoritmy na delenie a ukázal, že pri použití netradičnej metódy sa dosahujú lepšie výsledky z hľadiska rýchlosti výpočtu.

Nasledujúca kapitola (ôsma) hovorí jednak všeobecne o *typoch dát* a okrem toho podrobne rozoberá *štyri základné skalárne typy dát*. Autor vysvetľuje potrebu a užitočnosť definície typu premennej, uvádza všeobecnú schému na definíciu typu a zdôrazňuje, že sa predpokladá isté usporiadanie medzi prvkami jednotlivých typov dát. Podrobne sú charakterizované štandardné typy: boolean, integer, char a real. Pri type integer zdôrazňuje autor existenciu maximálneho čísla pre daný počítač a ťažkosti s asociatívnym zákonom. Pri type boolean uvádza základné boolovské operácie, základné zákony (de Morganove pravidlo, asociatívny zákon atď.) a relačné operátory. Pri type char sú uvedené niektoré štandardné množiny charakterov, prevodové funkcie a usporiadanie. Typ real chápe autor ako konečnú množinu reprezentantov intervalov reálneho kontinua. Uvádza reprezentáciu čísiel v pohyblivej čiarky a základné axiómy, ktorým musí vyhovovať strojová reprezentácia reálnych čísiel, aby sa s nimi dalo rozumne počítať. Na príkladoch demonštruje autor ťažkosti s asociatívnym a distributívnym zákonom. Autor definuje mieru presnosti počítačovej aritmetiky a opisuje prevodové funkcie medzi typmi real a integer. Na riešení kvadratickej rovnice ilustruje autor problémy numerickej stability.

Originálnym spôsobom je spracovaná kapitola 9 o *programoch založených na rekurentných reláciách*. Autor tu uvádza dve programové schémy založené na príkaze while a ukazuje všeobecné vlastnosti výpočtov podľa týchto schém. Použitie týchto schém ilustruje na nasledujúcich programoch pre výpočet: $N!$, x^{-1} , \sqrt{x} a $\sin x$ pomocou nekonečných radov. V závere autor formuluje pravidlo na čistenie cyklov.

Elegantne je spracovaná nasledujúca kapitola 10 o *zoznamoch ako základnom type štrukturovaných dát*. Každému zoznamu f je automaticky priradená funkcia, označovaná $f^!$, ktorá pracuje ako „buffer“ pri prenose informácie do a zo zoznamu. Autor vymenováva typické zariadenia, ktoré zapamätávajú alebo spracovávajú zoznamy. Matematicky elegantne sú definované operácie put, reset, get, rewrite a predikát eof. Podrobne sa hovorí o spracovávaní textových zoznamov a uvádzajú sa príklady na grafickú reprezentáciu funkcií a na vkladanie znakov na riadenie tlačiarne do textových zoznamov. Autor sa pritom vyhýba obvyklým podrobnostiam o vstupno-výstupných procedúrach, definuje však, ako štandardné, zoznamy input a output.

O *masívoch* sa hovorí v kapitole 11. Autor začína s jednorozmerným masívom a ukazuje, čím sa líši od zoznamu. Zdôrazňuje, že obidvom štruktúram zodpovedajú rôzne typy pamäti. Uvádza programy na hľadanie prvku v usporiadanom masíve a pre skalárny súčin vektorov. Tieto príklady slúžia autorovi ako motivácia na prirodzené zavedenie príkazu for a tento príkaz porovnáva s príkazmi while a repeat. Nasleduje program pre triedenie. V závere kapitoly zavádza autor viacrozmerné masívy, a ako príklad program na násobenie matic.

Aj kapitola 12 o tak zložitej problematike ako sú *podprogramy, procedúry a funkcie*, je pomerne veľmi stručná. Autor vychádza nielen z toho, že účelom procedúr je skrátiť program, ale, čo považuje za významnejšie, díva sa na procedúru ako na prostriedok na rozdelenie a štrukturovanie programov do logicky súvislých a uzavretých zložiek. Podrobne rozoberá problematiku lokálnych a globálnych objektov. Berú sa do úvahy tri typy substitúcie parametrov: substitúcia hodnotou, referenciou a menom. Autor uvádza aj parametrické procedúry a ich použitie ilustruje na Simpsonovej integrácii.

Nasledujúcu kapitolu venuje *transformáciám* na rôznych číselných reprezentáciách. Sú tu uvedené programy pre vstup a výstup celých čísiel, ktoré prevádzajú daný zoznam čísiel na celé číslo a naopak. Ďalej je uvedený algoritmus pre vstup čísiel x , $0 \leq x \leq 1$, a pre transformáciu reálnych čísiel v pohyblivej čiarky z jednej číselnej sústavy do inej.

Predposledná kapitola obsahuje tri programy na spracovanie abecednočíslícových textov použitím zoznamov a masívov: program, ktorý do daného abecednočíslícového zoznamu vsunie symboly pre tlačiareň tak, že každý riadok bude obsahovať maximum informácií a slová v texte sa nerozdelia; program, ktorý v danom texte z nahradí výskyt podslova x slovom y ; posledný príklad ilustruje základy syntaktickej analýzy a udáva metódu, ako k danému regulárnemu výrazu automaticky zostrojíte program, ktorý určí, či dané slovo je v jazyku reprezentovanom týmto regulárnym výrazom.

Cieľom poslednej kapitoly (15) je na podrobne prepracovaných príkladoch ilustrovat systematický prístup k vytváraniu programov. Autor sa tu zmieňuje jednak o metóde „zhora-nadol“, pri ktorej sa postupne algoritmus detailizuje, až kým sa nedostaneme na úroveň používaného programovacieho jazyka, a aj o metóde „zdola-nahor“, pri ktorej sa vychádza z používaného programovacieho jazyka a postupne sa skupiny príkazov sústreďujú do primitívnych procedúr, z nich sa vytvárajú procedúry vyššieho typu atď., až kým sa nedostaneme k výslednému programu. Autor stručne uvádza, za akých okolností je ktorý prístup výhodnejší. Prirodzene, že základnou metódou je metóda zhora-nadol. Systematický prístup k vytváraniu programov ilustrujú ďalšie štyri problémy: riešenie systému lineárnych algebraických rovníc eliminačnou metódou; určenie najmenšieho celého čísla, ktoré sa dá dvoma spôsobmi vyjadriť ako súčet dvoch tretích mocnín celých čísel ($x = a^3 + b^3 = c^3 + d^3$); určenie prvých n prvočísel a príklad na heuristický algoritmus, ktorý má generovať takú postupnosť N symbolov z abecedy (0, 1, 2), že žiadne dve susedné podpostupnosti nie sú rovnaké. Autor pritom nezanedbáva ani analýzu zostrojených algoritmov z hľadiska zložitosti výpočtu. Táto kapitola je najdlhšia a reprezentuje 20 % knihy. Kapitola súčasne ilustruje, že vytváranie algoritmov je zložitý proces. Príklady volil autor dobre. Snáď len druhý príklad je dosť ťažko zrozumiteľný a vo výslednom algoritme je aj dosť jasná chyba: za desiaty riadokom, ktorý má tvar „end“, treba vsunúť príkazy označené na s. 136 ako (15.30).

Z obsahu knihy vidieť, že napriek malému počtu strán pokrýva pozoruhodné množstvo materiálu. Je ale jasné, že o celom rade ďalších pojmov a techník možno diskutovať, či by mali alebo nemali byť v takejto základnej učebnici. Malý rozsah knihy ponecháva v tomto smere veľa priestoru pre vyučujúceho.

Záverom by som ešte raz doporučil Wirthovu knihu ako učebnicu programovania.

Preklad Wirthovej knihy má vyjsť vo vydavateľstve ALFA.

Jozef Gruska,

Tibor Šalát, *Nekonečné rady*, Academia (Cesta k vědě 23), Praha, 1974, 244 strán, Kčs 18,—

Šalátova kniha bola rozobratá za niekoľko dní. Je to v československej literatúre prvé skutočne hodnotné dielo o nekonečných radoch.

Prvá zo štyroch kapitol má prípravný charakter a obsahuje potrebné poznatky o funkciách, číslach a metrických priestoroch. Ťažisko knihy je v najrozsiahlšej druhej kapitole (122 strán), v ktorej, popri všeobecnej teórii nekonečných radov a súčínov s reálnymi aj komplexnými členmi, nájde čitateľ niektoré špeciálne a novšie výsledky (prerovnanie radov, vlastnosti čiastočných radov a iné). Tretia kapitola hovorí o funkcionálnych postupnostiach a radoch. Študujú sa rôzne druhy konvergencie (bodová, rovnomerná, kvázirvnomerná). V súvislosti s tým sa dokazuje Weierstrassova veta

o aproximácii spojitaj funkcie a v cvičeniach sa uvádza príklad spojitaj funkcie bez derivácie. Ďalej sa skúmajú potenčné rady a Taylorov rad. Posledná kapitola, s nadpisom „Základy teórie sumovateľnosti radov“, obsahuje niektoré zovšeobecnenia pojmu limity postupnosti so zameraním na metódy sumovateľnosti divergentných radov.

Kniha má moderné poňatie. Materiál je účelne vybraný so zreteľom na zaujímavé súvislosti a aplikácie. Hlavný text dopĺňa množstvo starostlivo vybraných cvičení. Na štýle knihy sa prejavuje autorova vedecká kvalifikácia a veľké pedagogické umenie. Kniha je predovšetkým vynikajúcou učebnicou. Vzhľadom na to bude užitočná veľmi širokému okruhu čitateľov, počnúc študentmi matematiky všetkých zameraní a foriem štúdia.

Kniha vyšla v nepatrnom náklade 1500 výtlačkov. V ďalších vydaniach bude treba upraviť úvahy o polomere konvergenencie logaritmického a binomického radu, lebo metóda použitá v príkladoch d) a e) na s. 207 až 208 dáva len polovicu intervalu konvergenencie (na túto skutočnosť recenzenta upozornil L. Kollár). Na s. 163 má byť $f_n(x)$ namiesto $f(x)$ a v definícii funkcie g_n na s. 199 je tlačová chyba (správne má byť $g_n(z) = a_n(z^n - z_1^n) / (z - z_1)$).

Ladislav Kosmák

Václav Medek a kolektív, *Matematická terminológia*, SPN, Bratislava 1975, 144 strán, Kčs 15,—

V r. 1975 vyšla v SPN malá knižka s názvom *Matematická terminológia*, ktorú spracoval kolektív slovenských matematikov pod vedením prof. RNDr. VÁCLAVA MEDEKA. Je to akýsi slovník výrazov, symbolík a hesiel používaných v školskej matematike. Dlhé roky žiadna podobná kniha, ktorá by zjednocovala matematickú terminológiu, u nás nevyšla. Toto vydanie zaplňa vzniknutú medzeru a súčasne spĺňa aj ďalší cieľ — uvádza mnoho nových pojmov, ktoré sa vďaka modernizácii matematiky v stále väčšej miere používajú aj v školskej matematike. Všetci učitelia matematiky by si mali osvojiť túto jednotnú terminológiu, aby nedochádzalo k nedorozumeniam z viacznačnosti a preto, aby sa pestovala čistota matematického jazyka. Bolo by vhodné dôsledne používať názvy ako napr. karteziánsky súčin (namiesto doteraz častého kartézsky), rovnica s n neznámymi (doteraz rovnica o n neznámym), Diofantovská rovnica (doteraz Diofantická), afinitná priamka (doteraz niekedy afínna) atď.

V súlade s Pravidlami slovenského pravopisu (SAV 1970) možno používať názvy sčítanie i sčítovanie čísiel, rovníc atď.

Stojí za povšimnutie, že *Matematická terminológia* prisudzuje číslu 1 v elementárnej aritmetike názov jeden (mužský rod), zatiaľ čo v geometrii názov jedna (ženský rod).

V *Matematickej terminológii* čitateľ nájde okrem uvedeného aj stručné definície a vlastnosti niektorých matematických pojmov a operácií, ako aj platné vzorce a iné.

Knižka *Matematická terminológia* je poučná a potrebná, preto by mala mať svoje miesto v knižnici každého učiteľa matematiky.

Eva Nyulassyová