

Algoritmizace

V každém okamžiku ví procesor počítače přesně, co má vykonat. Pojem procesor se v souvislosti s algoritmy označuje objekt (např. stroj i člověk), který vykonává činnost popisovanou algoritmem. Abychom mohli formulovat algoritmus, musíme znát konkrétní procesor, protože na jeho povaze závisí elementární kroky.

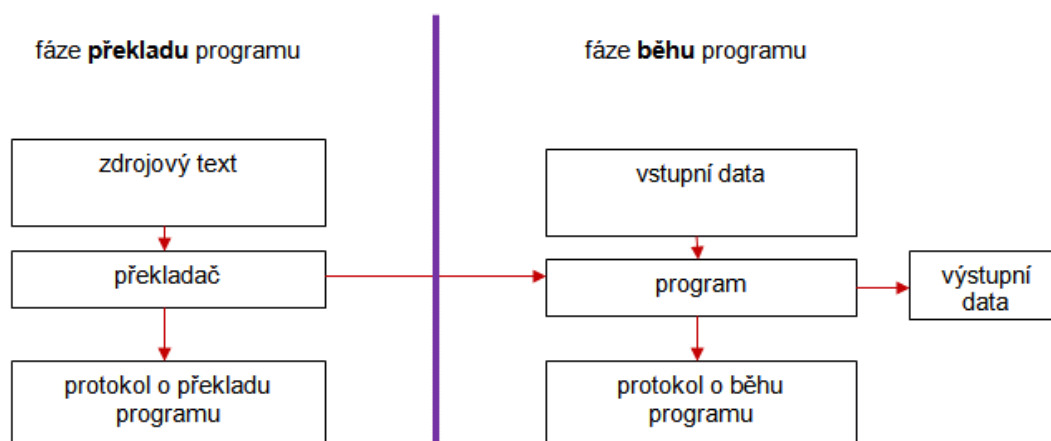
Děj, který trvá konečně dlouho a má přesně definovaný účinek, nazýváme **akce**. Každá akce se projeví změnou stavu objektů, kterých se týká. Každou akci můžeme popsat pomocí pojmů nějakého jazyka nebo prostřednictvím soustavy obecných ustálených předpisů. Popis akce se nazývá příkaz. **Příkaz** je tedy předpis pro provedení akce, přičemž akce je jednotlivá část výpočetního procesu. Pokud jednotlivé akce následují po sobě, nazýváme tento proces sekvenčním procesem. Posloupnost příkazů nazýváme **program**. Procesor je objekt, který vykonává akce na základě programu.

Díky programům (software) mohou uživatelé počítač ovládat. Pojem program používáme pro algoritmy formulované tak, že je může vykonávat pouze nějaký konkrétní procesor. Rozdíl mezi obecným algoritmem a programem je v tom, že program musí do nejmenších detailů vyhovovat principům nějakého programovacího jazyka.

Programovací jazyk si můžeme představit jako souhrn vyjadřovacích prostředků pro zápis programů pro určitý procesor. Při definování programovacího jazyka definujeme jednak syntaxi jazyka (pravidla jak mohou vypadat programy v tomto jazyce), a dále sémantiku (pravidla, která každému správně utvořenému programu přiřadí jeho význam).

Je-li program sestavený správně, vydá po proběhnutí správná výstupní data.

Ve většině případů je pro vykonání programu zapotřebí „prostředníka“ - překladač zpracuje program jako svoje vstupní data, provede kontrolu správnosti syntaxe a vyhovuje-li syntaxi jazyka, program přeloží. Musíme tedy rozlišit fázi překladu (pracuje překladač) a fázi běhu (pracuje program).



Obrázek 1: Přeložení programu překladačem



evropský
sociální
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost

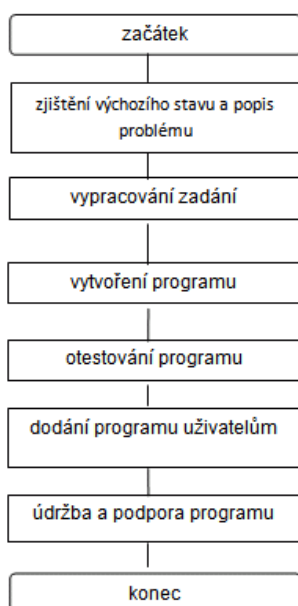
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Algoritmizace

Algoritmizace je postup, při kterém vytváříme program pro řešení nějakého problému. Můžeme ji rozdělit do etap:

1. Formulace problému – musíme přesně formulovat požadavky, určit výchozí hodnoty, požadované výsledky, formu a přesnost řešení.
2. Analýza úlohy – ověřujeme, zda je úloha řešitelná, zjišťujeme, zda výchozí hodnoty jsou k řešení postačující a zda má úloha více řešení. Vybereme nejvhodnější řešení.
3. Vytvoření algoritmu - sestavíme sled jednotlivých operací, které je třeba provést, aby byla úloha správně vyřešena.
4. Sestavení programu - sestavíme zdrojový text v konkrétním programovacím jazyce, ze kterého se pomocí překladače do strojového kódu vytvoří spustitelný program.
5. Odladění programu - cílem je odstranění chyb v programu. Syntaktické chyby odstraní překladač, logické chyby se projeví špatnou činností programu a chybnými výsledky. Pro jejich ladění a odstranění se používá debugger (ladicí program).

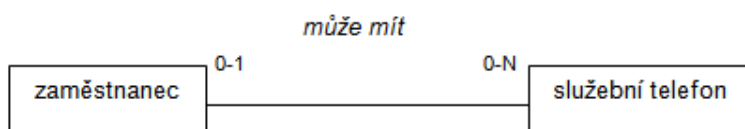
Analýzu daného problému provádí analytik - na základě analýzy vytvoří postup řešení např. ve formě formě vývojového diagramu. Podle tohoto vývojového diagramu napíše programátor v konkrétním programovacím jazyce program.



Obrázek 2: Činnosti při vývoji programu

Programy často řeší velice složité problémy, proto se pro jejich vyjádření dají využít kombinace slov a obrázků (určitý typ strukturovaného záznamu informací, můžeme se setkat s DFD diagramem, E-R Diagramem nebo Use Case diagramem).

Ilustrace č. 3 představuje situaci, kdy zaměstnanec podniku může mít nebo nemusí služební telefon a každý služební telefon může být nebo nemusí být zapůjčen zaměstnancem.



Obrázek 3: Diagram ukládání dat do počítače

Algoritmus

Algoritmus je základní matematický pojem. Původ slova algoritmus se datuje do začátku 9. století, kdy perský matematik Muhammad ibn Musa al-Khwarizmi sepsal v jedné z knih postupy pro počítání s čísly.

Protože se jedná o pojem, nemůžeme **algoritmus** definovat, musíme pro vyjádření použít opis. Algoritmus je složen z kroků a zajišťuje, že ze vstupních dat získáme požadovaná výstupní data.

Algoritmus musí splňovat následující vlastnosti:

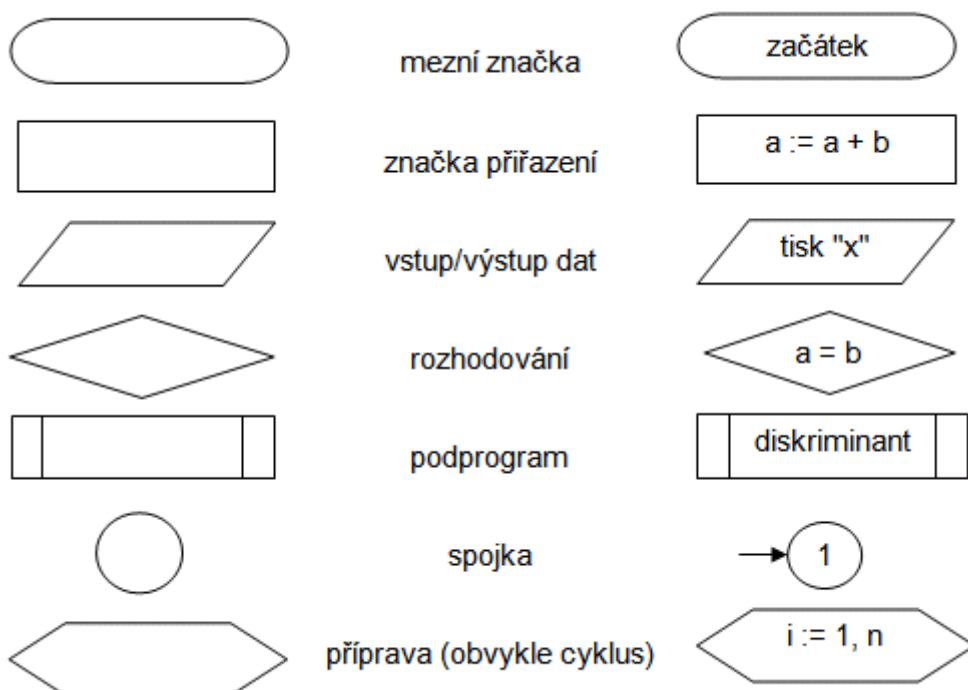
- Hromadnost – předpokládáme různé množiny vstupních hodnot v rámci předem definovaného oboru, např. řešení rovnice $ax + b = 0$ (nesprávně: $2 \times 9 = 16$). Při popisu algoritmu v programovacím jazyce se tato vlastnost projeví tak, že vstupy do algoritmu jsou označeny symbolickými jmény.
- Determinovanost - zaručuje, že při každém použití algoritmu dostaneme pro tytéž vstupní hodnoty stejný výsledek. Tzn., že kroky a jejich návaznosti musí být jednoznačně definované. Každý krok je přechod z jednoho stavu algoritmu do jiného, přičemž každý stav je přesně určen zpracovávanými daty. Tím, jak data v konkrétním kroku vypadají, musí být přesně dáno, který krok přesně následuje.
- Konečnost – každý algoritmus, který vychází z přípustných vstupních hodnot, musí za konečnou dobu skončit. Přitom požadujeme, aby algoritmus skončil „v rozumném čase“ - tj., v čase, kdy nám výsledek výpočtu bude k něčemu.
- Rezultativnost – algoritmus, který vychází z přípustných hodnot, vede ke správnému výsledku.

- Opakovatelnost – pokud algoritmus použije pokaždé stejné vstupní údaje, musí dospět vždy ke stejnému výsledku.

Pro zápis algoritmu se používá:

1. Přirozený jazyk – tj. slovní popis.
2. Grafické znázornění – obvykle vývojové diagramy.
3. Pseudojazyk – speciální jazyk.
4. Programovací jazyk.

Slovní popis algoritmu bývá doplněn grafickými ilustracemi, přesto je nepřesný. Častěji se setkáváme s vyjádřením algoritmu pomocí vývojového diagramu. K zakreslení vývojového diagramu se používají standardizované symboly (norma ČSN ISO 5807 Zpracování informací. Dokumentační symboly a konvence pro vývojové diagramy toku dat, programu a systému, síťové diagramy programu a diagramy zdrojů systému). Nejčastěji používané symboly ukazuje ilustrace č. 4.



Obrázek 4: Symboly pro vývojové diagramy toku dat



evropský
sociální
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Sestavení algoritmu

Algoritmus je sestavený na základě tří struktur: posloupnost (sekvence), větvení (rozhodování) a cykly (opakování).

Posloupnost (větvení)

Jedná se o řadu za sebou jdoucích a vzájemně navazujících kroků, jejichž pořadí je předem přesně dané, žádný krok nemůže být vynechán, např. výpočet aritmetického průměru ze dvou čísel, výpočtu objemu válce.

Posloupnost se v algoritmu vyskytuje samostatně nebo jako součást složitější struktury.

Větvení (rozhodování)

Větvení se použije v situaci, kdy je třeba některé kroky vynechat, nahradit nebo přidat. Větvení je složeno ze tří částí, přičemž první část – otázka – je vždy povinná. Na otázku existuje kladná nebo záporná odpověď. Druhá část je krok, který se provádí v případě kladné odpovědi, třetí část je krok, který se provádí v případě záporné odpovědi.

Obsahuje-li algoritmus všechny tři kroky, jedná se o úplné větvení, např. Jednotlivé kroky, které se mají vykonat po otázce, se značí symboly „+“ a „-“ nebo slovy „ano“ a „ne“.

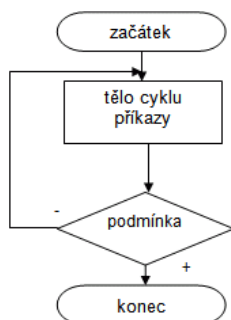
Při neúplném větvení chybí v algoritmu krok pro kladnou nebo pro zápornou odpověď (vyskytuje se častěji), např. výpočet absolutní hodnoty.

Ve vnořeném větvení je krok pro kladnou (zápornou) odpověď tvořený dalším větvením, např. hledání menšího z čísel.

Cykly (opakování)

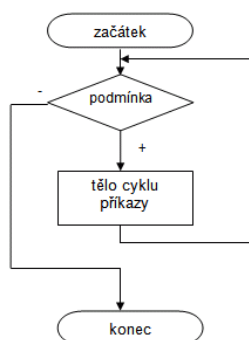
Pokud se v algoritmu vyskytne stav, kdy je nutné situaci opakovat, mluvíme o cyklu. V závislosti na konkrétní podmínce je počet opakování znám již dopředu nebo se opakování provádí do doby, než nastane konkrétní situace.

Cyklus s výstupní podmínkou (na konci): nejprve se provedou příkazy, pak se kontroluje podmínka. Příkazy těla cyklu se opakují do doby, než je podmínka splněna (ilustrace č. 5).



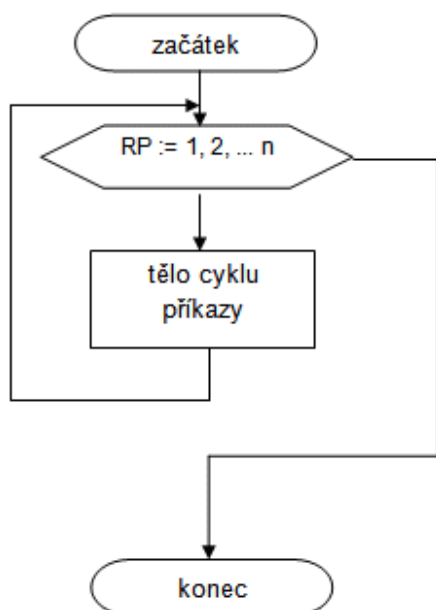
Obrázek 5: Cyklus s podmínkou na konci

Cyklus se vstupní podmínkou (na začátku): nejdříve se vyhodnotí podmínka, v případě kladné odpovědi se vykonají příkazy a provádění se vrátí k vyhodnocení podmínky. Cyklus bude ukončen v okamžiku, kdy podmínka bude vyhodnocena jako false (ilustrace č. 6). Má-li podmínka hodnotu false již při prvním vyhodnocení, nebude se cyklus provádět ani jednou.



Obrázek 6: Cyklus s podmínkou na začátku

Cyklus s řídicí proměnnou: lze jej použít, je-li dán počet opakování explicitně a nezávisí na činnosti prováděné v těle cyklu. Cyklus řídí řídicí proměnná. Na začátku se do řídicí proměnné uloží počáteční hodnota. Je-li hodnota řídicí proměnné menší nebo rovna koncové hodnotě, provedou se následující činnosti: vykoná se tělo cyklu, provede se návrat na začátek cyklu, hodnota řídicí proměnné se zvýší o 1, je-li hodnota řídicí proměnné menší nebo rovna koncové hodnotě, bude se činnost opakovat, cyklus se ukončí v okamžiku, kdy bude hodnota řídicí proměnné vyšší než koncová hodnota.



Obrázek 7: Cyklus s řídicí proměnnou



evropský
sociální
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Zdroje a prameny:

- [1] TRETEROVÁ, Eliška: Návrh a vývoj algoritmů. Ostravská univerzita v Ostravě, 2003. ISBN 80-7042-854-6.
- [2] PŠENČÍKOVÁ, Jana: Algoritmizace. Computer Media, 2009. ISBN 978-7402-034-6.
- [3] VRBÍK, V. Programování 1. 1. vyd. Plzeň: ZČU v Plzni, 2000. ISBN 80-7082-663-0.
- [4] VRBÍK, Václav. Algoritmy – řešené příklady. Pedagogické centrum Plzeň, 2002. ISBN 80-7020-103-7.
- [5] Úvod do algoritmizace a programování. In: ŠTEFAN, Radim. Oa-poruba.cz [online]. 2006. vyd. [cit. 2012-09-22]. Dostupné z: http://www.oa-poruba.cz/vp/prg/soubory/uvod_do_programovani.pdf
- [6] HRUŠKA, Ondřej. Základy algoritmizace a programování: Elektronický kurz pro střední školy. 2007. Dostupné z: http://algoritmizace.asp2.cz/algo/index_egen.html
- [7] DIVIŠ, Jozef. Algoritmizace. SPŠE Mohelnice. Dostupné z: <http://www.spsemoh.cz/vyuka/algor/index.htm>
- [8] Základy algoritmizace. In: VIRIUS, Miroslav. [online]. [cit. 2012-09-22]. Dostupné z: <http://www.rudisweb.wz.cz/dokumenty/algoritmizace.pdf>
- [9] ČSN ISO 5807. Zpracování informací. Dokumentační symboly a konvence pro vývojové diagramy toku dat, programu a systému, síťové diagramy programu a diagramy zdrojů systému. 1995.

Autorem materiálu a všech jeho částí, není-li uvedeno jinak, je Bohuslava Čezíková.