INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Výukový materiál zpracován v rámci projektu

EU peníze školám

Registrační číslo projektu: CZ. 1.07/1.5.00/34.0637

Šablona	III/2
Název	VY_32_INOVACE_39_Algoritmizace_teorie

evropský sociální fond v ČR
EVROPSKÁ UNIE
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Název školy	Základní škola a Střední škola Bohemia s.r.o. Víta Nejedlého 482 Chrudim
Jméno autora	Mgr. Markéta Valentová
Tematický okruh	Informační technologie pro 1. ročník SŠ
Ročník	1. ročník – 23 žáků
Téma	Algoritmizace - teoretická část
Anotace	Materiál je určen na vysvětlení algoritmizace. Jedná se o výukový materiál. Vyučující využívají k jeho prezentaci informační technologie.
Metodický pokyn	Žáci pracují samostatně písemnou formou.
Datum vytvoření	16. 9. 2012

ALGORITMIZACE

Algoritmus

= přesný popis, definující jistý proces, který vede od měnitelných vstupních údajů k žádaným výsledkům.
ZJEDNODUŠENĚ: = algoritmus je jednoznačný a přesný popis řešení problému
= přesný návod a postup, kterým lze vyřešit daný typ úlohy

Vlastnosti algoritmu:

- Determinovanost (předurčenost)
  algoritmus musí být přesný, srozumitelný a jednoznačný, tj. v každém místě je jednoznačně určen další krok a pro stejná vstupní data musí poskytovat stále stejné výsledky.
- Hromadnost
  algoritmus neslouží k řešení jen jedné úlohy, ale je řešením celé skupiny úloh, které se od sebe liší jen vstupními údaji. Vstupní údaje se mohou měnit v určitých mezích.
- Resultativnost (konečnost)
  hledané výsledky musíme získat po konečném počtu kroků, algoritmus musí po konečném počtu kroků skončit.

Některé problémy lze řešit více způsoby - různými algoritmy, které se mohou svým postupem značně lišit. Naší snahou je vybrat pro řešení problému co nejefektivnější algoritmus, který řeší problém v co nejkratším čase, je přehledný a srozumitelný.

Algoritmy můžeme zapisovat :

- slovně
- graficky (pomocí tzv. vývojových diagramů)

Etapy algoritmizace:

1. Formulace problému
2. Analýza úlohy
3. Vytvoření algoritmu
4. Sestavení programu
5. Odladění programu

Formulace problému
V této etapě je třeba přesně formulovat požadavky, určit výchozí hodnoty, požadované výsledky, jejich formu a přesnost řešení.

Analýza úlohy
Při analýze úlohy si ověříme, zda je úloha řešitelná a uděláme si první představu o jejím řešení. Dále zjistíme, zda výchozí hodnoty jsou k řešení postačující a zda má úloha více řešení. Podle charakteru úlohy vybereme nejvhodnější řešení.

Vytvoření algoritmu úlohy
Sestavíme jednoznačný sled jednotlivých operací, které je třeba provést, aby byla úloha správně vyřešena. Algoritmus přesně popisuje postup zpracování daného úkolu, nedává však odpověď na daný problém, ale pouze postup, jak ji získat.

Sestavení programu

Na základě algoritmu řešené úlohy sestavíme program (zdrojový text) v konkrétním programovacím jazyce. Ze zdrojového textu se pomocí překladače do strojového kódu vytvoří spustitelný program. Dá se tedy říci, že dobře provedená analýza úlohy a algoritmizace daného problému je základním předpokladem sestavení programu pro počítač.

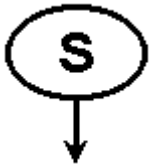
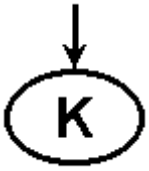
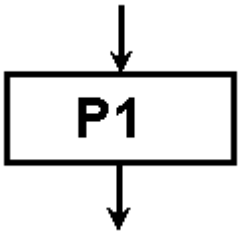
Odladění programu

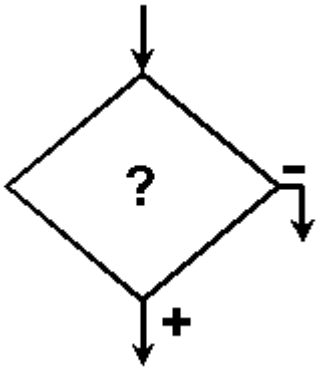
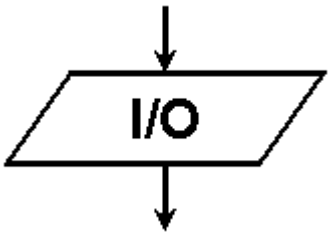
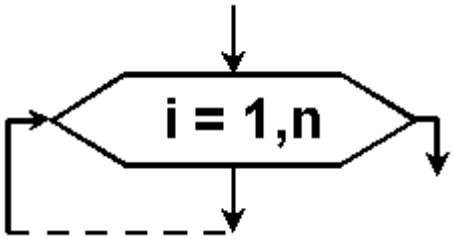
Cílem odladění je odstranění chyb z programu. Nejčastější chyby jsou chyby v zápise, tzv. syntaktické - ty odhalí překladač.
Závažnější jsou logické chyby, vyplývající z nesprávně navrženého algoritmu, nebo chyby, vzniklé špatným předpokladem v etapě formulace nebo analýzy úlohy - ty se projeví nesprávnou činností programu nebo špatnými výsledky - při odstraňování těchto chyb může pomoci ladící program (debugger) umožňující sledování aktuálního stavu proměnných a krokování.
Teprve po odstranění všech druhů chyb můžeme program použít k praktickému řešení úloh.

Vývojové diagramy

Jedním z mnoha způsobů znázornění algoritmů jsou vývojové diagramy. Je to grafické znázornění logické struktury řešeného úkolu.
Ve vývojových diagramech se používá několik typů značek, z nichž každé je přiřazen určitý význam. Do těchto značek se vpisují operace nebo skupiny operací, které se mají provést.

Používané značky:

	začátek algoritmu
	konec algoritmu
	blok zpracování (do bloku zapisujeme akce, které se mají provést)

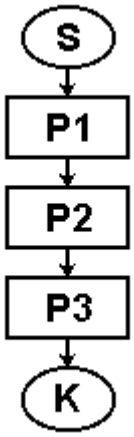
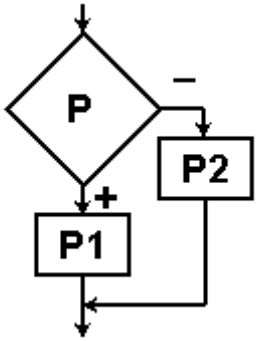
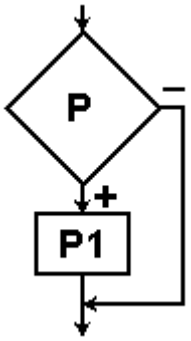
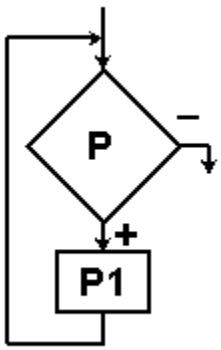
	blok rozhodování (do bloku zapisujeme podmínku)
	blok vstupu nebo výstupu
	blok pro cyklus se známým počtem průchodů
	spojka (pro rozsáhlé diagramy, rozdělené do několika částí)

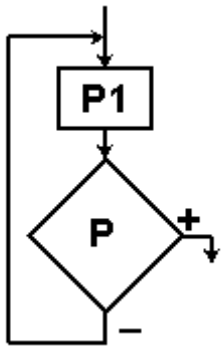
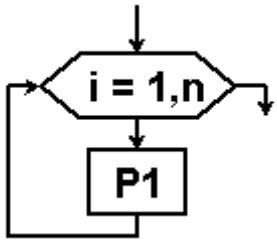
Základní algoritmické konstrukce

Algoritmy lze teoreticky sestavovat libovolně, ale vzhledem k přehlednosti a úmyslně omezeným možnostem programovacích jazyků se musí dodržovat několik zásad:

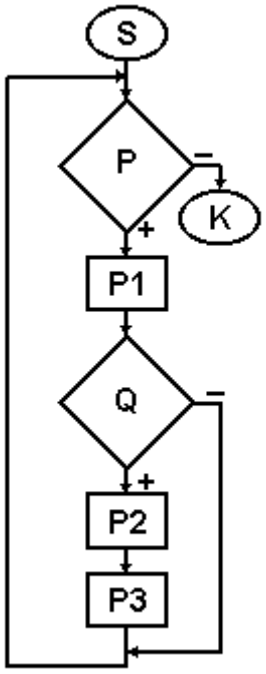
Algoritmus

- má mít jeden začátek a jeden konec
- musí být složen pouze ze základních algoritmických konstrukcí

Název popis	Diagram	Slovní vyjádření
Sekvence je posloupnost příkazů, které se postupně provedou		**Proveď příkazy P1, P2, P3.**
Větvení umožňuje rozdělit program do 2 větví podle toho, zda je nebo není splněna podmínka		**Jestliže platí podmínka P, proveď příkaz P1, jinak proveď příkaz P2.**
Větvení s prázdnou akcí umožňuje provést příkaz jenom tehdy, když je splněna podmínka		**Jestliže platí podmínka P, proveď příkaz P1.**
Cyklus s podmínkou na začátku Když podmínka není na počátku splněna, cyklus nemusí proběhnout ani jednou.		**Dokud platí podmínka P, prováděj příkaz P1.**

Cyklus s podmínkou na konci Tento cyklus musí proběhnout aspoň jednou.		**Opakuj příkaz P1, až do splnění podmínky P.**
Cyklus se známým počtem průchodů Cyklus proběhne v obecném případě n-krát		**Pro i od 1 do n prováděj příkaz P1.** během provádění cyklu řídící proměnná cyklu **i** postupně nabývá hodnot 1, 2, 3,...,n.

Složené algoritmické konstrukce

• Jednotlivé základní algoritmické konstrukce lze do sebe vnořovat. • Místo jednotlivých příkazů v konstrukcích mohou být celé sekvence příkazů.	

Zdroj:
http://www.spsemoh.cz/vyuka/algor/index.htm
http://cs.wikipedia.org/wiki/Algoritmus
http://bech.mzf.cz/pascal.html