



evropský
sociální
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Úvodní list

Název školy	Obchodní akademie, České Budějovice, Husova 1
Číslo šablony/číslo sady	32/03
Poř. číslo v sadě	01
Jméno autora	Jiří Novák
Období vytvoření materiálu	Březen 2013
Název souboru	VY_32_INOVACE_03_IKT_01
Zařazení materiálu podle ŠVP	Informační a komunikační technologie – 3. ročník 6. Algoritmizace
Téma	Algoritmizace – zápis algoritmu
Druh výukového materiálu	Pracovní list (MS Word) Prezentace (MS PowerPoint)
Anotace	Vysvětlení pojmů související s algoritmizací úloh. Cílem je naučit žáky používat vývojové diagramy pro zápis jednoduchých algoritmů. Použité pomůcky: Soubor s pracovním listem, editor s možností vkládání automatických tvarů pro vývojové diagramy (např. MS Word).
Použitý zdroj	Text KOSTOLÁNYOVÁ, K. <i>Algoritmizace a řešení problémů</i> . Ostrava: Ostravská universita v Ostravě, Pedagogická fakulta, 2002, ISBN 80-7042-227-0 Úkoly – vlastní zdroj

Algoritmizace

Algoritmus je posloupnost operací, která řeší daný úkol. Řešení úkolu musíme zapsat v drobných krůčcích, které je určitý programovací nástroj schopen realizovat.

Vlastnosti algoritmu

- **Hromadnost.** Algoritmus musí řešit danou úlohu pro různé vstupní hodnoty (nevypočítá tedy třeba $1 + 2 = 3$, ale $X + Y = Z$, kdy můžeme zadat libovolná čísla X a Y).
- **Determinovanost** (podmíněnost). Všechny operace i jejich návaznosti jsou jednoznačně určeny (definovány), nic nesmí být náhodné, neurčené. Důsledkem také je, že při stejných vstupních hodnotách dostaneme stejný výsledek při prvním běhu algoritmu i při stopadesátém.
- **Rezultativnost** (konečnost). Algoritmus musí celý proběhnout v konečném počtu kroků.

Zápis algoritmu, vývojový diagram

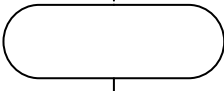
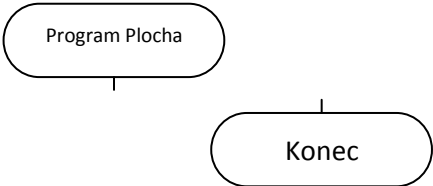
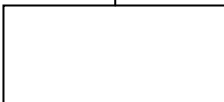
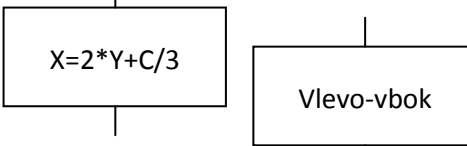
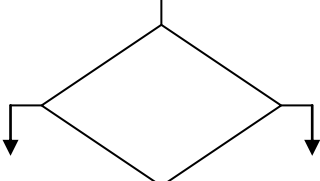
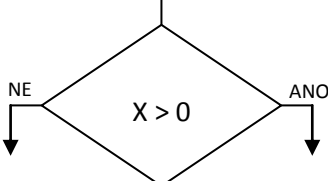
Z postupu vzniku programu víme, že algoritmus zapisujeme pomocí příkazů námi zvoleného programovacího jazyka. To však je přímo možné jen u velmi jednoduchých úloh. Složitější úlohy je zvykem přehledně znázornit do diagramu.

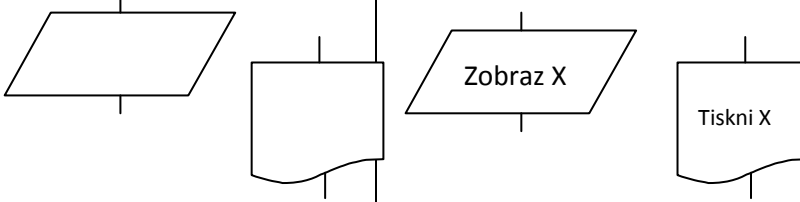
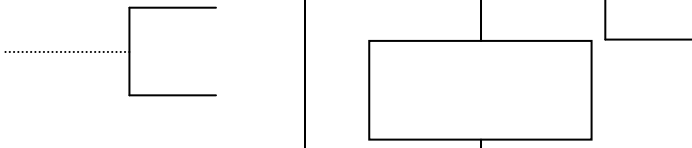
Vývojový diagram

Vývojový diagram je grafické znázornění jednotlivých příkazů, ze kterých se algoritmus skládá a jejich návazností pomocí normalizovaných značek.

Protože jsou značky normalizovány, nemusíme (a nesmíme) si je sami vymýšlet. Důsledkem bude, že našemu zápisu vývojového diagramu bude rozumět každý, kdo tyto značky zná.

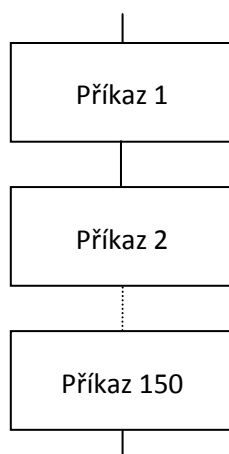
Základní značky používané ve vývojových diagramech

Název značky	Značka	Příklad zápisu
Mezní značka (začátek a konec programu)		
Zpracování (příkaz, operace, činnost)		
Větvení		

Vstup a výstup hodnot Tisk	
Poznámka	

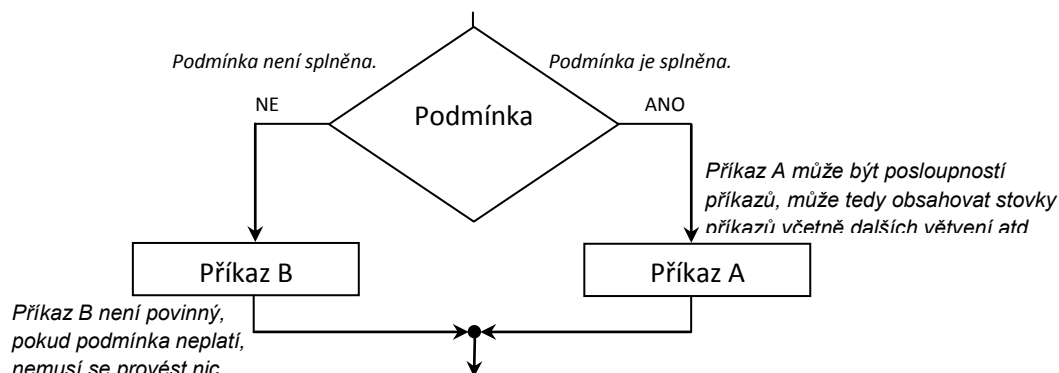
Posloupnost příkazů (složený příkaz)

Posloupnost (sekvence) příkazů obsahuje např. přiřazovací příkaz ($X=5$), nebo příkaz pro výpočet hodnoty ($X = Z + 5$) atd., prostě jakékoliv zpracování dat. Můžeme samozřejmě použít jen příkazy, které námi používaný programovací jazyk zná.



Podmíněný příkaz, větvení programu podle podmínek

Velmi často potřebujeme, aby se při splnění nějaké podmínky vykonal příkaz A a při jejím nesplnění příkaz B (nebo se nevykonal příkaz žádný a program pokračoval dál). To nám umožní podmíněný příkaz.

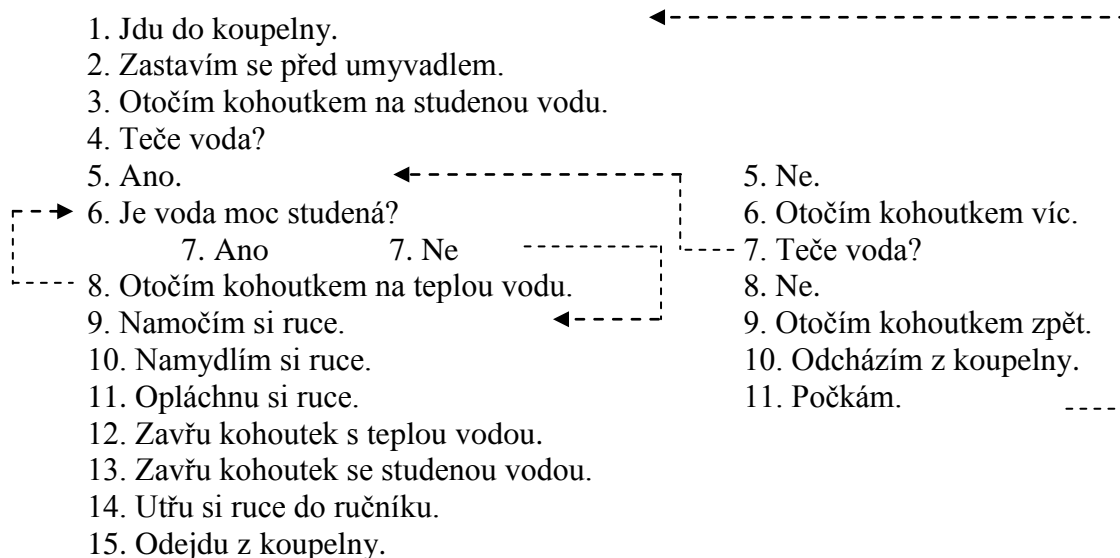


Úkoly

1. Navrhněte algoritmus na umytí rukou. (Nepoužívejte vývojové diagramy.)
2. Navrhněte algoritmus, který vypíše větší číslo ze dvou zadaných. (Využijte jen vývojových diagramů.)

Řešení úkolů

Úkol 1.



Poznámka:

Uvedený algoritmus je značně zjednodušený. Nejsou brány v úvahu další situace, které mohou nastat. (Např.: teče jen studená voda, voda je příliš horká, chybí mýdlo, schází ručník, v koupelně je tma apod.) Mohou nastat i různé naprosto neočekávané situace, které by měl algoritmus také řešit: praskne vodovodní trubka, bude zemětřesení, ...

Úkol 2.

