

Maturitní témata 2020/21

Otázka č. 16 – Počítačové sítě a programování

Algoritmizace:

- přesný pracovní postup, který se používá při tvorbě programu, jehož prostřednictvím lze řešit nějaký konkrétní problém

- algoritmizaci lze rozdělit do několika kroků:

- 1) Formulace problému - v této části je potřeba přesně formulovat požadavky, určit výchozí hodnoty, požadované výsledky, jejich formu a přesnost řešení, tvůrce algoritmu musí dokonale rozumět řešenému problému jinak nemůže algoritmus sestavit
- 2) Analýza úlohy - ověření 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 zvolíme nejvhodnější a nejjednodušší řešení
- 3) Vytvoření algoritmu úlohy - sestavení jednoznačného sledu jednotlivých instrukcí (příkazů) které je potřeba provést aby byl úkol správně vyřešen, algoritmus přesně popisuje postup zpracování daného úkolu, nedává však odpověď na daný problém, ale pouze jak danou odpověď získat
- 4) Sestavení programu - na základě algoritmu řešené úlohy sestavíme program, základním předpokladem pro zpracování programu je dobře provedená analýza úlohy a algoritmizace
- 5) Odladění programu - odstranění chyb z programu, nejčastěji se jedná o syntaktické chyby (v zápise), horší jsou logické chyby které vyplývají z nesprávně navrženého algoritmu, nebo chyby které vzniknou špatným předpokladem v etapě formulace nebo analýzy úlohy → projeví se nesprávnou činností programu nebo špatnými výsledky

Vlastnosti algoritmu:

- za algoritmus se považují postupy které splňují:

- 1) Elementárnost - algoritmus se skládá z konečného počtu jednoduchých (elementárních) kroků
- 2) Konečnost - každý algoritmus musí skončit v konečném počtu kroků, počet kroků může být libovolně velký ale pro každý jednotlivý vstup musí být konečný
- 3) Obecnost - algoritmus neřeší jeden konkrétní problém ale obecnou třídu obdobných problémů, má širokou množinu možných výstupů
- 4) Determinovanost - algoritmus je determinovaný, pokud za stejných podmínek (pro stejné vstupy) poskytuje stejný výstup, tato vlastnost je požadována u velké části úloh, existují však úlohy kdy je naopak vyžadována náhodnost (vrh kostkou, generování hesel)

- 5) Determinismus - každý krok algoritmu musí být jednoznačně a přesně definován, v každé situaci musí být naprosto zřejmé co a jak se má provést
- 6) Výstup - algoritmus má aspoň jeden výstup, veličinu, která je v požadovaném vztahu k zadaným vstupům, a tím tvoří odpověď na problém který algoritmus řeší
- 7) Opakovatelnost - algoritmus vede vždy ke stejným výsledkům jsou-li zadána stejná data

Způsoby zápisu algoritmu:

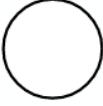

- Slovní vyjádření - všemožné návody na použití různých výrobků (návod na použití DVD přehrávače), nejméně přehledný
- Matematický zápis - vhodný tam kde je možné zapsat problematiku pomocí matematických vztahů, je jednoznačný, ve většině případů je příliš stručný
- Rozhodovací tabulky - v případech kdy se v úloze vyskytuje několik možností a vlastní řešení je pro každou možnost jednoduše popsatelné, např. rozvrh hodin pro určitou konkrétní třídu, zápis je jednoznačný, přehledný a srozumitelný, nehodí se pro každý typ úlohy
- Strukturogramy - úspornější znázornění algoritmu, jsou kombinací grafického a textového popisu, je tvořen obdélníkovou tabulkou, kde do řádků zapisujeme postup, přehlednější způsob znázornění, lze ho aplikovat i na složitější problémy, pracnost konstrukce, malé možnosti pozdějších úprav

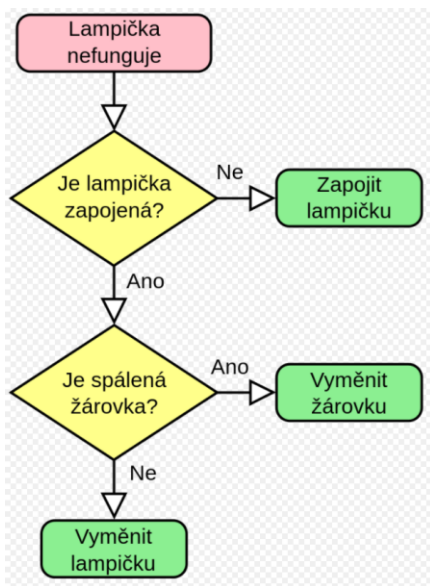
Řešení kvadratické rovnice			
Čti a, b, c			
Spočti diskriminant $D=b*b-4*a*c$			
Je $D>0$?			
	Ano	Ne	
	$x1=(-b+sqrt(D))/(2*a)$	Je $D=0$?	
	$x2=(-b-sqrt(D))/(2*a)$		Ano
	Tisk 2 kořenů: $x1, x2$		Ne
		$x1=(-b+sqrt(D))/(2*a)$	Tisk:
		Tisk 1 kořen: $x1$	Nemá v Řešení.

- Vývojové diagramy - k zachycení řešení se používají některé specifické grafické symboly, do kterých je zapsán slovní text

Značka	Popis
Mezní značka	
	Mezní značka představuje vstup z vnějšího prostředí do programu nebo výstup do vnějšího prostředí, např. začátek nebo konec programu (popř. podprogramu).
Vstup nebo výstup	
	Vstup – znázorňuje načtení dat potřebných pro činnost programu. Výstup – znázorňuje zobrazení výstupů na zobrazovacím zařízení.
Zpracování	
	Zpracování znázorňuje činnost programu, během níž dochází k přeměně dat (např.: matematickým či logickým operacím, k přiřazení hodnoty).

Značka	Popis
Rozhodování	
	Rozhodovací blok slouží k rozvětvení programu na základě podmínky uvedené uvnitř.
Příprava	
	Příprava označuje přípravnou fázi programu, nejčastěji se používá pro zahájení cyklu s pevným počtem opakování.
Podprogram	
	Podprogram nebo předem definovaná činnost znázorňuje samostatnou část algoritmu, která je zpracována (popsána) v jiné části programu.

Značka	Popis
Spojka	
	Spojka umožňuje spojit dvě části vývojového diagramu, které nebylo možné nakreslit souvisle. Spojky na konci přerušení a na začátku pokračování musí být označeny stejným číslem.
Poznámka	
	Poznámka nebo komentář slouží ke zkrácenému slovnímu popisu jednotlivých úkonů ve vývojovém diagramu.



- Program v programovacím jazyce - algoritmus je zapsaný v jazyce, kterému počítač rozumí a umí z něho vytvořit strojový kód, je to jediná forma které rozumí člověk i PC, tato forma se nedá ničím nahradit ani obejít, málo názorná a přehledná

Časová a paměťová složitost:

- nástroj kterým lze porovnat efektivitu a rychlost provádění jednotlivých algoritmů, určuje operační náročnost algoritmu, slouží ke zjištění praktické použitelnosti algoritmů a programů
- časová složitost se odvíjí od počtu provedených operací, matematická funkce která vyjadřuje závislost spotřeby času pro zpracování dat na vstupních datech
- paměťová (prostorová) složitost se odvíjí od velikosti datových struktur které algoritmus využívá, matematická funkce která vyjadřuje závislost spotřeby paměťového prostoru pro zpracování dat na vstupních datech