14. 10. 2024

1. Algoritmizace úloh

Základní pojmy:

- Algoritmus = je přesně definovaná konečná posloupnost operací, která pro přípustná vstupní data vede po konečném počtu kroků k požadovaným výsledkům / jednoznačný a konečný popis postupu, který vede k vyřešení daného problému
- Proces = - je konkrétní realizace algoritmu
  - podle jednoho algoritmu může probíhat několik různých procesů
  - je realizován procesorem (člověkem, počítač, ...)

  / - Procesor ⇒ obecně ten (ta), kdo (co) vykonává příkazy algoritmu
  - např. kuchař vaří podle receptu; počítač vykonává program
- Akce = činnost, která má konečné trvání a přesně definovaný účinek
  - popis akce ⇒ příkaz

Co to je algoritmus? - musí být rezultativní (časová složitost)
- musí být konečný ⇒ konečný počet základních příkazů
- musí být jednoznačný (DETERMINISTICKÝ) ⇒ v každém kroku je jednoznačně určeno, jak pokračovat dále → vyloučená náhoda

(základní množina jazyka) → musí být jednoduchý (ELEMENTÁRNÍ) ⇒ jednotlivé příkazy musí být jednoduché a jasné, nelze si je vysvětlit různými způsoby

(obecnost) - hromadnost algoritmu ⇒ lze použít pro řešení obecné úlohy - musí být použitelný pro libovolná vstupní data

Program = zápis algoritmu tak, aby mu rozuměl procesor, který jej bude vykonávat

Algoritmizace = je postup při tvorbě programu pro počítač, kterým lze prostřednictvím algoritmu řešit nějaký problém

Etapy Algoritmizace: 1) Formulace problému
2) Analýza úlohy
3) Vytvoření algoritmu
4) Sestavení programu
5) Odladění programu

- DÉKOMPOZICE = znamená rozdělit složitý problém na menší části, které jsou jednodušší na řešení. Každou část řešíme zvlášť, ale nakonec zohledníme celý problém. (např. psaní knihy) → SHORA DOLŮ (rekurze)
  - menší moduly nebo funkce (např. UI, databáze, výpočetní logika)
- ABSTRAKCE = se zaměřujeme na ignorování detailů, které nejsou důležité, a zaměření se na klíčové prvky problému. Pomáhá řešit problém po částech, bez rozptylování detaily. (např. ovládání auta)
  - přecházíme s abstrakcí ⇒ používáme vyšší úroveň rozhraní a funkcí (nemusíme znát různé detaily)

  → Z DOLA NAHORU → z menších problémů jdeme na větší
- V realitě se používají zároveň ⇒ jdou proti sobě

Způsoby vyjádření algoritmů:

1) Slovní popis
2) Rozhodovací tabulky
3) Diagramy
4) Programovací jazyky

+ další (Vývojový – grafický jazyk)

1. Algoritmizace úloh

14. 10. 2024

1) Slovní popis

- nejméně formální způsob popisu algoritmu
- musí splňovat výše uvedené vlastnosti,
  hlavně nesmí být nejednoznačný
- např.: recepty v kuchařce, pracovní návod na zařízení

3) Diagramy

- Rozhodovací stromy – pro rozhodovací analýzy
- Petriho sítě – pro vícevláknové účely
  - umožňuje modelovat paralelní programy
- Vývojový diagram – obecný nástroj pro zápis algoritmů

4) Programovací jazyk

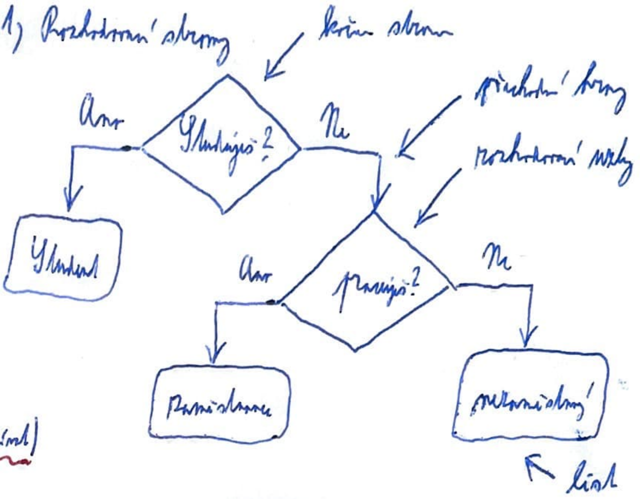
- nejformálnější způsob popisu algoritmu (přesně daný jak to provádět)
- slouží pro řízení počítačů, automatů, robotů, ...
- Kompilovaný ⟹ program se převede do spustitelného binárního (.exe) souboru
  - tak program je rychlejší (C, C++, Pascal)
- Interpretovaný ⟹ program je řádkově vykonáván speciálním programem
  - interpreter
  - Python, Java

2) Rozhodovací tabulky

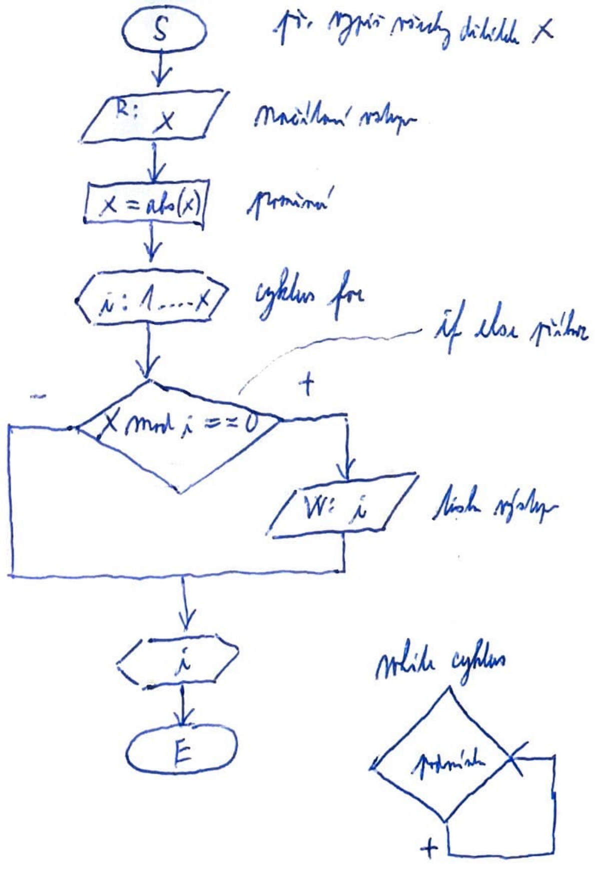
- např. porucha při odstraňování závad na tiskárně

		pravidla			
podmínky	tiskárna netiskne	Y	Y	N	N
	svítí červené světlo	Y	N	Y	N
akce	zkontroluj kabel	.		X	
	zkontroluj náplň	X			
	zásek papíru	X	X		

3.1) Rozhodovací stromy



3.3) Vývojové diagramy UMĚT ZAKRESLIT


S
př. vypiš všechny dělitele X
R: X
načtení vstupu
X = abs(X)
proměnná
i : 1....X
cyklus for
if else příkaz
−
+
X mod i == 0
W: i
tisk výstupu
i
E
while cyklus
podmínka
+