

Obsah

Obsah

1 Program přednášek	1
2 Podmínky zápočtu	2
3 Co je algoritmus?	2
3.1 Trocha historie	2
3.2 Definice algoritmu	3
3.3 Vlastnosti algoritmu	3
4 Zapisování algoritmů	4
4.1 Slovní popis	4
4.2 Vývojový diagram	4
4.3 Pseudojazyk	5
4.4 Programovací jazyk	5
5 Vývojové diagramy	6
6 Algoritmus, program, programovací jazyk	6
6.1 Příklady algoritmů	7

1 Program přednášek

Program předmětu

1. Úvod, algoritmus, program, programovací jazyk.
2. Paměť počítače, jednoduché datové typy, interpretace čísel, struktura programu v jazyce Pascal.
3. Znakové a řetězcové typy, funkce pro práci s řetězci.
4. Řídící struktury – podmínky, cykly.
5. Strukturované programování – procedury, funkce.
6. Rozklad problému na podproblémy, rekurze.
7. Práce se soubory.

Program předmětu

8. Strukturované datové typy.
9. Správa datových struktur na heapu, ukazatele.
10. Modulární programování – unity.
11. Složitější datové struktury – fronta, zásobník, ...
12. Funkce v externích jednotkách (CRT, DOS, GRAPH).
13. Rezerva
14. Rezerva, informace o zkoušce.

2 Podmínky zápočtu

Podmínky zápočtu

1. Povinná docházka.
 - Jsou povolené 3 neomluvené neúčasti.
 - Jako omluvu účasti na cvičení je možné uznat pouze kontrolku od lékaře (ne pouhé potvrzení).
 - Jako účast je považována „aktivní účast“ – při neaktivitě je cvičení neomluvené (a neomluvitelné).
2. Úspěšné absolvování 2 průběžných písemných prací.
3. Vypracování semestrální práce.

3 Co je algoritmus?

3.1 Trocha historie

Trocha historie

- أبو عبد الله محمد بن موسى الخوارزمي ابو جعفر
 - Abú Abd Alláh Muhammad Ibn Músá al-Chórezmí Abú Dža'far
 - Významný perský matematik (cca. 780-840)
- دنہا باسحب قيرفتل و عمال باتك
 - Kitáb al-džám'a wa-l-tafríq bil-hisáb al-hindi
 - Z této knihy jsou arabské číslice a hlavně nula

- **قالب اقل و رب جلا باسح**

- Hisáb *al-džabr wa-l-muqábala*
- Původ slova algebra

- Jméno polatinštěno na Al-Gorizmi, později Algoritmi. Odtud odvozeno slovo algorismus (provádění aritmetiky pomocí arabských číslic), které se zkomolilo na algorithmus a nakonec algoritmus.

3.2 Definice algoritmu

Algoritmus

Algoritmus je postup řešení určité třídy úloh tvořený seznamem jednoznačných příkazů, který pro každou přípustnou kombinaci vstupních dat zaručuje požadovaný výsledek po provedení konečného počtu kroků.

3.3 Vlastnosti algoritmu

Vlastnosti algoritmu

- Konečnost – důležité Každý algoritmus musí skončit v *konečném* počtu kroků. Postupy, které tuto podmínku nesplňují, se mohou nazývat *výpočetní metody*.
- Determinismus – důležité Každý krok algoritmu musí být jednoznačně a přesně definován; v každé situaci musí být naprosto zřejmé, *co a jak se má provést*, jak má provádění algoritmu pokračovat.
- Obecnost – méně důležité Algoritmus neřeší jeden konkrétní problém (např. „jak spočítat 3×7 “), ale obecnou třídu obdobných problémů (např. „jak spočítat součin dvou celých čísel“).
- Vstup Algoritmus obvykle pracuje s nějakými *vstupy*, veličinami, které jsou mu předány před započítím jeho provádění, nebo v průběhu jeho činnosti. Vstupy mají definované množiny hodnot, jichž mohou nabývat.
- Výstup Algoritmus má alespoň jeden *výstup*, veličinu, která je v požadovaném vztahu k zadaným vstupům, a tím tvoří odpověď na problém, který algoritmus řeší.
- Efektivita Obecně požadujeme, aby algoritmus byl *efektivní*. Požadujeme, aby každá operace požadovaná algoritmem, byla dostatečně jednoduchá na to, aby mohla být alespoň v principu provedena v konečném čase pouze s použitím tužky a papíru.

4 Zapisování algoritmů

Zapisování algoritmů

- Jak se dá zapsat algoritmus?
 - Slovní popis
 - Vývojový diagram
 - Program v pseudojazyku
 - Program v programovacím jazyku

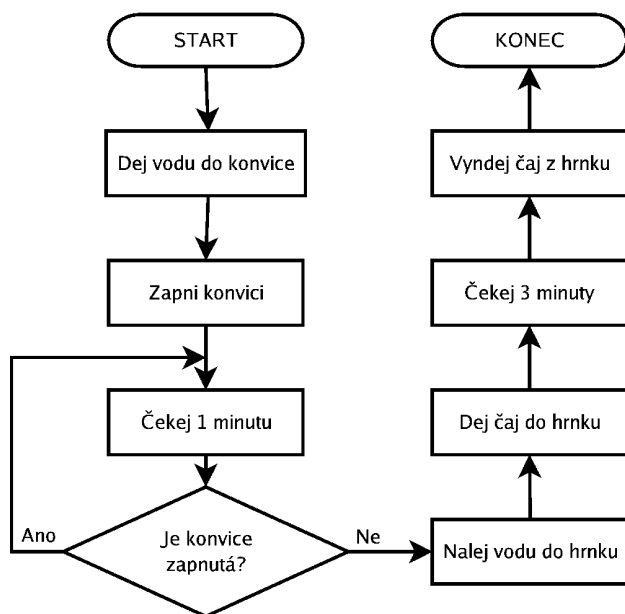
4.1 Slovní popis

Algoritmus pomocí slovního popisu

1. Dej vodu do konvice
2. Zapni konvici
3. Čekaj 1 minutu
4. Pokud je konvice zapnutá, pokračuj bodem 3
5. Nalej vodu z konvice do hrnku
6. Dej čaj do hrnku
7. Čekaj 3 minuty
8. Vyndej čaj z hrnku
9. KONEC

4.2 Vývojový diagram

Algoritmus pomocí vývojového diagramu



4.3 Pseudojazyk

Algoritmus v pseudojazyku

```

FUNKCE uvar_caj:
    dej_vodu_do_konvice;
    zapni_konvici;
    DOKUD je_konvice_zapnuta DELEJ
        cekej(1);
    KONEC_DOKUD
    nalej_vodu_do_hrnku;
    dej_caj_do_hrnku;
    cekej(3);
    vyndej_caj_z_hrnku;
KONEC_FUNKCE
  
```

4.4 Programovací jazyk

Algoritmus v programovacím jazyku

```

FUNCTION uvar_caj;
BEGIN
    dej_vodu_do_konvice;
    zapni_konvici;
    WHILE je_konvice_zapnuta DO
  
```

```

BEGIN
    cekej(1);
END;
nalej_vodu_do_hrnku;
dej_caj_do_hrnku;
cekej(3);
vyndej_caj_z_hrnku;
END;

```

5 Vývojové diagramy

Trochu více o vývojových diagramech

- Vývojové diagramy jsou standardizovaný prostředek pro zápis algoritmů (ČSN 36 9030).
- Používají několik symbolů v kombinaci se slovním popisem.
- Některé používané symboly:
 - Spojka, mezní značka, spojnice
 - Zpracování
 - Rozhodování
 - Příprava/modifikace
 - Vstup/výstup dat
 - Ruční vstup, interní paměť, paměť s přímým přístupem (disk, ...)
 - Zobrazení

6 Algoritmus, program, programovací jazyk

Program × algoritmus

- Algoritmus = postup práce
- Program = posloupnost příkazů
- Program realizuje (implementuje) algoritmus, algoritmus je jeho neoddělitelnou součástí.

Programy a programovací jazyky

- Program je předpis pro provedení určitých akcí (algoritmu) zapsaný v programovacím jazyku.
- Jazyky se dají rozdělit na:

- strojově orientované (strojový jazyk, assembler)
- vyšší jazyky (imperativní, neimperativní)
- Imperativní jazyky (C, Pascal, Java, Basic)
 - Údaje mají formu datových objektů - proměnné, konstanty
 - Program obsahuje deklarace, příkazy, komentáře, ...

Syntaxe × sémantika

- *Syntaxe* = skladba
 - souhrn pravidel udávajících přípustné tvary dílčích konstrukcí a celého programu
- *Sémantika*
 - význam jednotlivých konstrukcí, které jsou zapsané podle pravidel udaných syntaxí
- Syntaktická pravidla jsou „formou“ programu a dají se jednoznačně zapsat.
- Sémantika je „obsah“ programu.

6.1 Příklady algoritmů

Příklady algoritmů

- Kuchařský recept (není regulérní algoritmus)
- Přecházení přechodu
- Výpočet zeměpisné polohy
- Výpočet faktoriálu
- Hledání nejmenšího společného dělitele
- Zjištění integrálu funkce *není algoritmus*
- Zjištění derivace funkce
- Třídění čísel
- Výpočet přesné hodnoty odmocniny *není algoritmus*