

Algoritmizace

David Weber

SPŠE Ječná

23. září 2023

Algoritmizace

Algoritmizace

Chci něco spočítat, jak na to?

Algoritmizace

Chci něco spočítat, jak na to?

- Rozmyslím si postup výpočtu

Algoritmizace

Chci něco spočítat, jak na to?

- Rozmyslím si postup výpočtu
- Provedu výpočet podle vymyšleného postupu

Algoritmizace

Chci něco spočítat, jak na to?

- Rozmyslím si postup výpočtu
- Provedu výpočet podle vymyšleného postupu

Potřebuji nutně rozumnět postupu?

Algoritmizace

Chci něco spočítat, jak na to?

- Rozmyslím si postup výpočtu
- Provedu výpočet podle vymyšleného postupu

Potřebuji nutně rozumnět postupu?

⇒ Nepotřebuji, důležitá je jeho správnost!

Algoritmus

Algoritmus

Co je to algoritmus?

Přesný návod či postup, kterým lze vyřešit daný typ úlohy.

Algoritmus

Algoritmus

V užším slova smyslu se algoritmem rozumí takové postupy, které mají určité vlastnosti.

Algoritmus

V užším slova smyslu se algoritmem rozumí takové postupy, které mají určité vlastnosti.

Vlastnosti algoritmu

- **Elementárnost (diskrétnost).** Algoritmus se skládá z konečného počtu jednoduchých (elementárních) kroků.

Algoritmus

V užším slova smyslu se algoritmem rozumí takové postupy, které mají určité vlastnosti.

Vlastnosti algoritmu

- **Elementárnost (diskrétnost).** Algoritmus se skládá z konečného počtu jednoduchých (elementárních) kroků.
- **Konečnost (finitnost).** Algoritmus musí skončit v konečném počtu kroků.

Algoritmus

V užším slova smyslu se algoritmem rozumí takové postupy, které mají určité vlastnosti.

Vlastnosti algoritmu

- **Elementárnost (diskrétnost).** Algoritmus se skládá z konečného počtu jednoduchých (elementárních) kroků.
- **Konečnost (finitnost).** Algoritmus musí skončit v konečném počtu kroků.
- **Obecnost (hromadnost).** Algoritmus neřeší jeden konkrétní problém, ale obecnou třídu obdobných problémů (např. neřeší jen „kolik je $2 \cdot 2$ “, ale obecně součin libovolné dvojice čísel $a \cdot b$).

Algoritmus

V užším slova smyslu se algoritmem rozumí takové postupy, které mají určité vlastnosti.

Vlastnosti algoritmu

- **Elementárnost (diskrétnost).** Algoritmus se skládá z konečného počtu jednoduchých (elementárních) kroků.
- **Konečnost (finitnost).** Algoritmus musí skončit v konečném počtu kroků.
- **Obecnost (hromadnost).** Algoritmus neřeší jeden konkrétní problém, ale obecnou třídu obdobných problémů (např. neřeší jen „kolik je $2 \cdot 2$ “, ale obecně součin libovolné dvojice čísel $a \cdot b$).
- **Determinovanost.** Po každém kroku lze jednoznačně určit, který následuje.

Algoritmus

V užším slova smyslu se algoritmem rozumí takové postupy, které mají určité vlastnosti.

Vlastnosti algoritmu

- **Elementárnost (diskrétnost).** Algoritmus se skládá z konečného počtu jednoduchých (elementárních) kroků.
- **Konečnost (finitnost).** Algoritmus musí skončit v konečném počtu kroků.
- **Obecnost (hromadnost).** Algoritmus neřeší jeden konkrétní problém, ale obecnou třídu obdobných problémů (např. neřeší jen „kolik je $2 \cdot 2$ “, ale obecně součin libovolné dvojice čísel $a \cdot b$).
- **Determinovanost.** Po každém kroku lze jednoznačně určit, který následuje.
- **Správnost.** Algoritmus řeší danou úlohu, tj. pro přípustná data vydá správný výsledek a nesprávná vstupní data zamítne.

Záznam algoritmu

Záznam algoritmu

Každý záznam (popis) algoritmu musí být

Záznam algoritmu

Každý záznam (popis) algoritmu musí být

- **Srozumitelný.** Je jasné, co a jak algoritmus řeší.

Záznam algoritmu

Každý záznam (popis) algoritmu musí být

- **Srozumitelný.** Je jasné, co a jak algoritmus řeší.
- **Přehledný.** Ze záznamu je algoritmus rychle uchopitelný.

Záznam algoritmu

Každý záznam (popis) algoritmu musí být

- **Srozumitelný.** Je jasné, co a jak algoritmus řeší.
- **Přehledný.** Ze záznamu je algoritmus rychle uchopitelný.
- **Jednoznačný.** Každý krok musí být jednoznačně popsán (vágní popisy jsou nežádoucí).

Záznam algoritmu

Každý záznam (popis) algoritmu musí být

- **Srozumitelný.** Je jasné, co a jak algoritmus řeší.
- **Přehledný.** Ze záznamu je algoritmus rychle uchopitelný.
- **Jednoznačný.** Každý krok musí být jednoznačně popsán (vágní popisy jsou nežádoucí).
- **Stručný.** Neuvádíme zbytečné detaily (matoucí).

Způsoby zadání algoritmu

Způsoby zadání algoritmu

- 1 Slovním vyjádřením \Rightarrow např. tzv. **pseudokódem**.

Způsoby zadání algoritmu

- ❶ Slovním vyjádřením \Rightarrow např. tzv. **pseudokódem**.

Příklad pseudokódu

1. Vytvoř proměnnou x
2. Pro $i = 1, 2, \dots, 10$ opakuj:
3. Do x ulož hodnotu $x + i$
4. Vypiš hodnotu proměnné x

Způsoby zadání algoritmu

- ❶ Slovním vyjádřením \Rightarrow např. tzv. **pseudokódem**.

Příklad pseudokódu

1. Vytvoř proměnnou x
2. Pro $i = 1, 2, \dots, 10$ opakuj:
3. Do x ulož hodnotu $x + i$
4. Vypiš hodnotu proměnné x

\Rightarrow Pozor, zde je obzvlášť třeba dbát na jednoznačný zápis!

Způsoby zadání algoritmu

- ❶ Slovním vyjádřením \Rightarrow např. tzv. **pseudokódem**.

Příklad pseudokódu

1. Vytvoř proměnnou x
2. Pro $i = 1, 2, \dots, 10$ opakuj:
3. Do x ulož hodnotu $x + i$
4. Vypiš hodnotu proměnné x

\Rightarrow Pozor, zde je obzvlášť třeba dbát na jednoznačný zápis!

- ❷ Smluveným grafickým vyjádřením (diagramem)

Způsoby zadání algoritmu

- ❶ Slovním vyjádřením \Rightarrow např. tzv. **pseudokódem**.

Příklad pseudokódu

1. Vytvoř proměnnou x
2. Pro $i = 1, 2, \dots, 10$ opakuj:
3. Do x ulož hodnotu $x + i$
4. Vypiš hodnotu proměnné x

\Rightarrow Pozor, zde je obzvlášť třeba dbát na jednoznačný zápis!

- ❷ Smluveným grafickým vyjádřením (diagramem)

Druhy diagramů

- Plošný strukturogram (tzv. „blokáč“)
- Vývojový diagram

Prvky algoritmu

Prvky algoritmu

- **Sekvence.** Jednotlivý krok, vykoná se vždy.

Prvky algoritmu

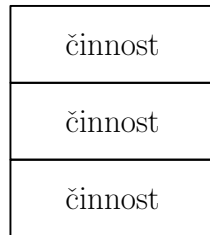
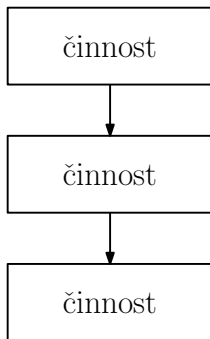
- **Sekvence.** Jednotlivý krok, vykoná se vždy.
- **Selekce.** Větvení programu, volba pokračování dle stanovené podmínky.

Prvky algoritmu

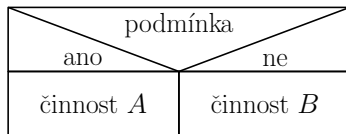
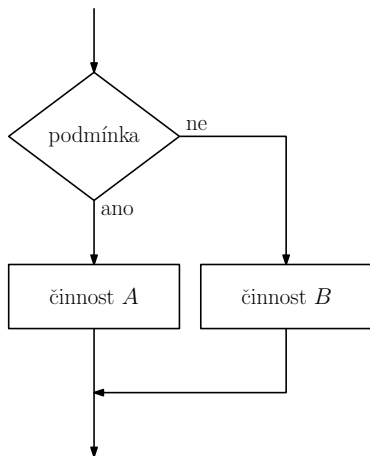
- **Sekvence.** Jednotlivý krok, vykoná se vždy.
- **Selekce.** Větvení programu, volba pokračování dle stanovené podmínky.
- **Iterace.** Cyklus, tj. opakované provádění určité posloupnosti příkazů, dokud je splněna stanovaná podmínka.
 - Iterace *s testem na začátku*
 - Iterace *s testem na konci*

Sekvence

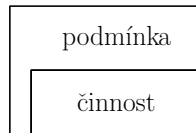
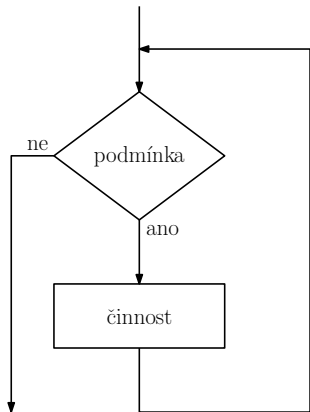
Činnosti jsou vykonávány v pořadí od shora dolů.



Selekce

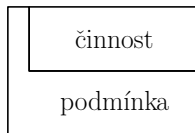
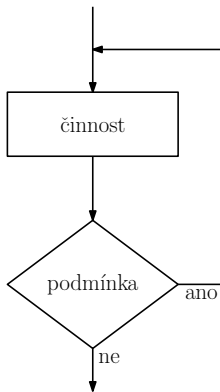


Iterace s testem na začátku



Iterace s testem na konci

Oproti iteraci s testem na začátku je zde zaručeno, že tělo iterace se vykoná **alespoň jednou**.



Dotazy?

