**Programování**

**Pozn.:** Abu Abdalah Muhammad Ibn Músa Al-Chvárizmí byl arabský matematik, který žil okolo roku 800 n. l. – od jeho jména je odvozeno slovo **ALGORITMUS**.

**Algoritmizace úlohy, vlastnosti algoritmu**

**ALGORITMUS** je posloupnost operací, která řeší daný úkol. Algoritmus vykonává nějaký **PROCESOR** (tím není myšlen pouze počítač, ale třeba také člověk). **OPERACE** použité v algoritmu musí použitý procesor umět vykonat a řešení úkolu je nutné zapsat pomocí základních operací, které je určitý procesor schopen realizovat.

S tvorbou algoritmu souvisí výběr vhodného programovacího nástroje, tzv. **PROGRAMOVACÍHO JAZYKA**. Na něm závisí, jaké **PŘÍKAZY** a **OPERACE** (funkce, objekty) máte k dispozici pro realizaci svého algoritmu. Aby postup řešení byl opravdu algoritmus, musí mít následující **VLASTNOSTI:**

* **HROMADNOST** – Algoritmus musí řešit danou úlohu pro různé vstupní hodnoty (nevypočítá tedy 1+2=3, ale X+Y=Z, kdy můžete zadat libovolná čísla X a Y).
* **DETERMINOVANOST (PODMÍNĚNOST)** – Všechny operace i jejich návaznosti jsou jednoznačně určeny (definovány), nic nesmí být náhodné.
* **OPAKOVATELNOST** – je důsledkem determinovanosti, při stejných vstupních hodnotách dostanete vždy stejný výsledek.
* **REZULTATIVNOST (KONEČNOST)** – Algoritmus musí celý proběhnout v konečném počtu kroků.

**Vývojový diagram**

**VÝVOJOVÝ DIAGRAM** je grafické znázornění jednotlivých příkazů, ze kterých se algoritmus skládá, a jejich návazností pomocí normalizovaných značek.

Vývojový diagram umožňuje přehledně a obecně zapsat zejména jednoduché algoritmy. U složitějších je potřeba použít nějakou metodu návrhu a pomocí vývojových diagramů zapisovat jednotlivé části celkového řešení.

**METODY NÁVRHU ALGORITMU**

* **METODA OZNAČOVANÁ SHORA DOLŮ** – Problém jako celek rozdělíte na podúlohy. Ty rozdělíte na další podúlohy, až dojdete k operacím, které je možné použít jako elementární příkazy.
* **METODA OZNAČOVANÁ ZDOLA NAHORU** – Postupujete opačně, z elementárních kroků vytváříte větší celky, které nakonec umožní zvládnout požadovaný problém.

Obvyklý postup pak většinou kombinuje obě uvedené metody.

**ZÁKLADNÍ ZNAČKY POUŽÍVANÉ VE VÝVOJOVÝCH DIAGRAMECH:**

|  |  |  |
| --- | --- | --- |
| NÁZEV ZNAČKY | ZNAČKA | PŘÍKLAD ZÁPISU |
| Mezní značka  (začátek a konec programu) |  | Program Plocha  Konec |
| Zpracování  (příkaz, operace, činnost) |  | vlevo v bok  X=2\*Z+C/3  X=500 |
| Větvení |  | NE  ANO  X==0 |
| Vstup a výstup hodnot  Souborové operace, tisk, vstup a výstup dat |  | Tiskni X  Vypiš X  Čti Y |
| Podmínka, komentář |  | Výpočet objemu |

**VÝVOJOVÉ DIAGRAMY ZÁKLADNÍCH PŘÍKAZŮ**

* **PŘÍKAZ (POSLOUPNOST PŘÍKAZŮ, SLOŽENÝ PŘÍKAZ)**  
  - Posloupnost (sekvence) příkazů obsahuje například přiřazovací příkaz ( X=5), nebo příkaz pro výpočet hodnoty a její přiřazení do proměnné (X=Z+5) atd. Můžete samozřejmě použít jen příkazy, které vámi používaný procesor (programovací jazyk) zná.

Příkaz 2

Příkaz 1

Příkaz 150

* **PODMÍNĚNÝ PŘÍKAZ**- Podmíněný příkaz umožňuje větvení programu podle podmínek. Velmi často potřebujete, 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 umožňuje podmíněný příkaz, často realizovaný příkazy IF (jestliže platí), THEN (potom udělej) a ELSE (pokud neplatí, udělej).

Příkazem A může být posloupnost příkazů, může tedy obsahovat stovky příkazů včetně dalších větvení.

Příkaz B není povinný. Pokud podmínka neplatí, nemusí se provést nic.

Podmínka není splněna, proměnná X má hodnotu např. 0.

Podmínka je splněna, proměnná X má hodnotu např. 5.

Příkaz A

Příkaz B

THEN

ELSE

ANO

NE

Podmínka

Např.

X>0

* **CYKLY**- Cyklus představuje opakování příkazů do splnění nějakých podmínek nebo po stanovený počet opakování.  
  - **CYKLUS WHILE … DO** se také označuje jako cyklus s podmínkou na začátku:   
   Cyklus WHILE … DO, dokud (je splněna podmínka), dělej.Cyklus REPEAT … UNTIL, opakuj … dokud (není splněna podmínka)  
  - **CYKLUS FOR** nepotřebuje žádnou podmínku, je však nutné zadat (a tedy předem vědět), kolikrát se musí určitý příkaz provést:

Jakmile je I větší než N, podmínka není splněna a cyklus se ukončí.

Dokud je podmínka splněna, tedy I je menší nebo rovno N (0, 1, 2, …, 30) provádí se příkaz A.

Tento příkaz nastaví hodnoty proměnných I a N, třeba I=0 a N=30.

FOR

Ne

Ano  
DO (Dělej)

Podmínka  
I=<N

Příkaz 2

Příkaz 1

Příkaz A

Podmínka může třeba znít: Je toto poslední záznam?

Příkazem A může být posloupnost příkazů.

REPEAT (Opakuj)

UNTIL  
(Dokud ne)

Ano

Příkaz A

Podmínka

Dokud je podmínka splněna, tedy záznam není poslední, provádí se příkaz A.

Jakmile bude záznam poslední, podmínka nebude splněna a cyklus se ukončí.

Podmínka může v tomto případě třeba znít: Toto není poslední záznam?

Příkaz A

Ano  
DO (Dělej)

Ne

WHILE (Pokud)

Podmínka

* **REKURZE**Rekurze je volání procedury X z procedury X, tedy procedura volá sama sebe.