

2 Vývojové diagramy

Vývojový diagram je symbolický algoritmický jazyk, který se používá pro názorné zobrazení algoritmu zpracování informací a případnou stručnou publikaci programů. Tento jazyk je tvořen přesnými definované značkami s jejich jednoznačným významem (sémantiku - slovník) a pravidla jak tyto značky ve vzájemné souvislosti používat (syntax - gramatika). Jeho používání je vhodné zejména u začínajících programátorů, protože dovoluje názorným způsobem formulovat postup řešení daného úkolu s vyznačením všech jeho možných alternativ. Používá se však i při týmovém řešení problému, kdy slouží jako komunikační prostředek mezi analytiky a programátory. (Analytik na základě provedené analýzy sestaví postup řešení ve formě vývojového diagramu, programátor pak podle tohoto diagramu napíše v odpovídajícím programovacím jazyce program). Vývojové diagramy jsou též vhodným prostředkem pro publikaci programů z několika důvodů. Např. je neefektivní publikovat program v některém z programovacích jazyků, každý program je dnes dílem, na který se vztahuje autorský zákon [16] a je tímto zákonem chráněn, atd.

Pro kreslení vývojových diagramů platí od 1. ledna 1996 nová česká státní norma ČSN ISO 5807 "Zpracování informací. Dokumentační symboly a konvence pro vývojové diagramy toku dat, programu a systému, síťové diagramy programu a diagramy zdrojů systému" [3]. Nahrazuje dřívější československou státní normu ČSN 36 9030 "Značky vývojových diagramů pro systémy zpracování dat" [4], která platila od roku 1974 (s nepatrnými změnami přijatými v roce 1989¹⁰). Přijetím této normy se i naše republika připojila k mezinárodní normě ISO 5807:1985 "Information processing - Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts".

Norma specifikuje symboly používané v dokumentaci pro zpracování informací a poskytuje návod pro jejich použití:

- a) ve vývojových diagramech toku dat;
- b) ve vývojových diagramech programu;
- c) ve vývojových diagramech systému;
- d) v síťových diagramech programu;
- e) v diagramech zdrojů systému..

Vývojový diagram programu je pak definován jako zobrazení posloupnosti operací v tomto programu a skládá se z následujících prvků:

- a) symbolů pro vlastní operace zpracování, včetně symbolů definujících stanovený tok, který má být dodržen při zachování logických podmínek;
- b) spojnic indikujících tok informací (dat);
- c) zvláštních symbolů pro usnadnění čtení a zápis vývojového diagramu.

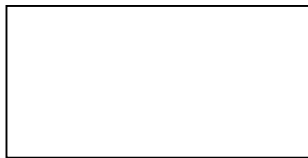
Vývojový diagram představuje tedy grafické znázornění logické struktury řešeného problému. Může být sestaven tak, že daný problém řeší v obecné rovině bez zřetele na speciální vlastnosti konkrétního počítače a programovacího jazyka. V praxi je však častěji používán druhý způsob, který má přímý vztah k požívanému počítači a zejména možnostem a vlastnostem zvoleného programovacího jazyka.

¹⁰ První norma tohoto druhu, ČSN 17 9801 "Značky blokových schémat pro programování samočinných počítačů", byla u nás vydána v roce 1964.

2.1 Symboly vývojových diagramů

Symboly vývojových diagramů představují grafické značky přesně definovaného významu. Pro upřesnění funkce symbolu se do nich vpisují slovní nebo symbolické operace, nebo i celé skupiny operací. Způsob psaní a symbolika tohoto textového zápisu nejsou normou určeny. Doporučuje se však používat jednoduchý text a výpočetní vztahy s použitím matematických značek podle normy ČSN ISO 31-11 "Veličiny a jednotky - Část 11: Matematické značky a značky používané ve fyzikálních vědách a v technice" (ne symboliky konkrétního programovacího jazyka jak se s tím můžeme často v řadě publikací i učebnicích setkat). Tím zajistíme čitelnost tohoto vývojového diagramu i čtenářům neznalým žádný programovací jazyk, případně programátorům, kteří budou daný vývojový diagram přepisovat do konkrétního programovacího jazyka.

Dále se seznámíme s některými základními vybranými symboly. S kompletním přehledem všech symbolů se čtenář může podrobně seznámit ve výše uvedené normě.

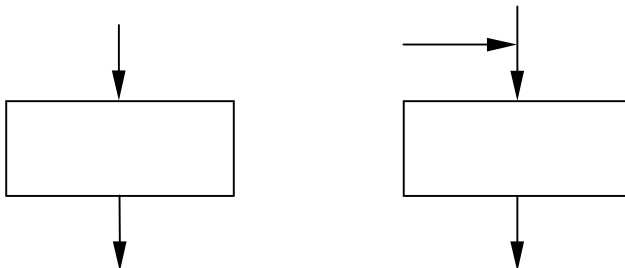


2.1.1 Zpracování

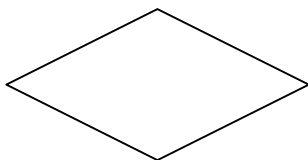
Symbol představující jakýkoliv druh zpracování nebo provedení definované operace nebo skupiny operací, jejichž výsledkem je transformace informace, např. změna hodnoty, umístění a pod. Možnost vstupu do tohoto symbolu je z libovolné strany a těchto vstupů může být i několik. Přitom to

mohou být vstupy samostatné, nebo se může jednat o jednu spojnicí výslednou, která vznikla spojením několika spojnic dílčích. Výstup je však v zásadě vždy jenom jeden.

Příklady:



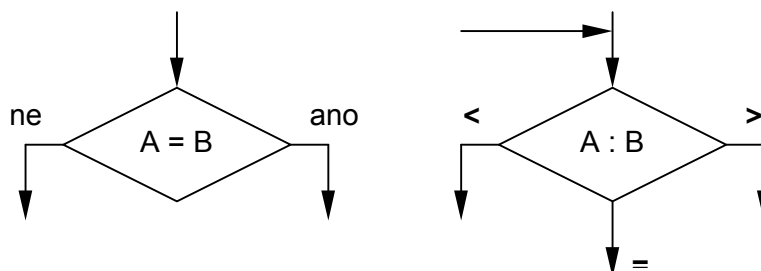
2.1.2 Rozhodování



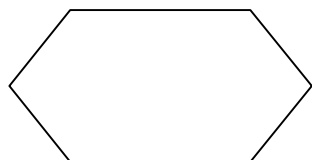
Symbol představuje rozhodovací nebo přepínací funkci. Symbol má jeden vstup (ten může být realizován jako výsledná spojnice) a alternativní výstupy. Daný výstup je aktivován po vyhodnocení podmínek uvnitř symbolu. Příslušné výsledky vyhodnocení podmínky mohou být zapsány přidruženě u spojnic reprezentujících dané cesty dalšího řešení. Tato značka má obvykle dva výstupy (je-li vývojový

diagram podkladem pro sestavení programu vzhledem k bistabilnímu charakteru počítače), obecně může mít i tři výstupy a může mít i jeden větvený výstup (viz spojnice).

Příklady:

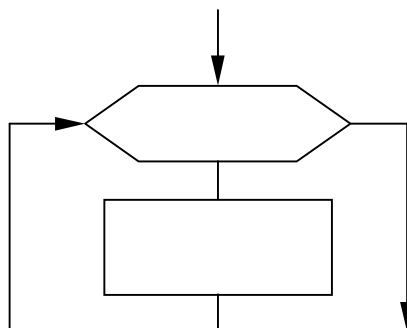


2.1.3 Příprava

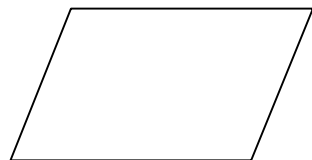


Tento symbol představuje úpravu nebo modifikaci (proto se mu někdy také říká modifikační symbol) činnosti, která mění vlastní postup následné činnosti, např. nastavení přepínače, vyjmenování hodnot, kterých nabývá proměnná cyklu, úprava indexového registru a jiné. Symbol má dva vstupy, jeden sekvenční, druhý pro návrat po provedení příslušného bloku operací a dva výstupy, jeden vstupující do daného bloku operací, druhý sekvenční, který pokračuje do další části programu.

Příklad:



2.1.4 Data - vstup a výstup dat



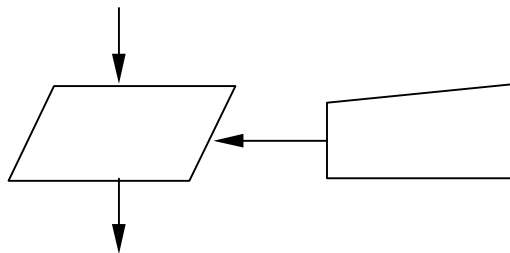
Symbol reprezentuje vstupně - výstupní operace s daty, tj. dodání dat pro zpracování v programu nebo zpracování dat do požadované formy výstupu. Druh vstupního nebo výstupního zařízení nebo nosiče dat je buď jednoznačně určen charakterem zpracovávané úlohy, nebo může být popsán slovně, nebo k tomuto symbolu je možné připojit symbol charakterizující vstupně-výstupné zařízení nebo nosič dat. V případě použití vstupně - výstupního zařízení je tento připojen k uvedenému symbolu paralelně Symbol má jeden vstup (realizovaný jako výsledná spojnice) a jeden výstup.

2.1.5 Ruční vstup

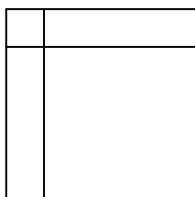


Tento symbol představuje všechny nosiče a zařízení pro ruční vstup informací jako jsou např. spřažená klávesnice počítače, světelné pero, snímač čárového kódu, různé přepínače apod. Symbol má jenom jeden výstup.

Příklad



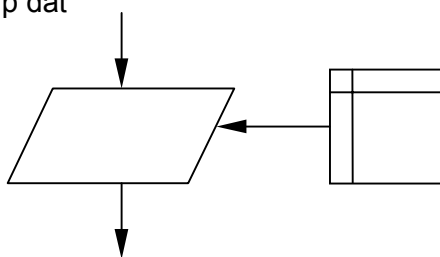
2.1.6 Interní paměť



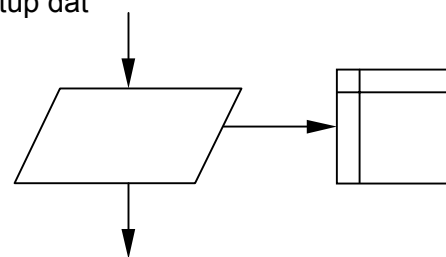
Symbol představuje nosič dat v případě, že nosičem je vnitřní paměť počítače. Data v tomto případě mohou do programu z tohoto nosiče vstupovat, nebo se mohou z programu do této paměti ukládat. Symbol může tak mít jeden výstup nebo jeden vstup.

Příklady:

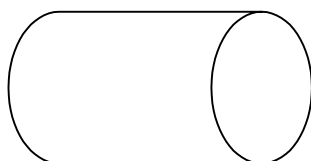
a) vstup dat



b) výstup dat

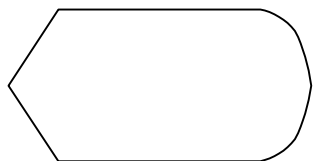


2.1.7 Paměť s přímým přístupem



Symbol představuje nosič s přímým přístupem dat. Příkladem je např. magnetický disk, magnetický buben nebo pružný disk (disketa). Podobně jako u interní paměti mohou data do programu z tohoto nosiče vstupovat, nebo se mohou z programu do této paměti ukládat. Symbol může mít tak jenom

jeden vstup nebo jeden výstup.



2.1.8 Zobrazení

Tento symbol představuje zařízení pro vizuální zobrazení dat, jako je např. monitor počítače, spřažené indikátory, signální zařízení a pod. Symbol má jenom jeden vstup.



2.1.9 Dokument

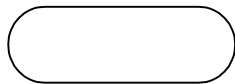
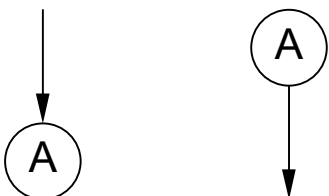
Symbol představuje zařízení pro tištěný výstup, kterým může být klasická tiskárna, ale i zařízení pro tvorbu mikrofilmu, pokladní tiskárna a jiné. Symbol má jenom jeden vstup.



2.1.10 Spojka

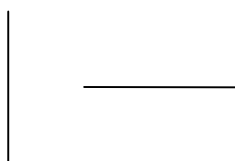
Tento symbol představuje přechod z jedné části vývojového diagramu na jinou část. Používá se k přerušení spojnice a k jejímu pokračování na jiném místě. Odpovídající symboly musí obsahovat stejné označení. Spojka má buď pouze jeden vstup nebo jeden výstup. Spojku lze použít i pro zobrazení větvení programu pomocí přepínače.

Příklad:



2.1.11 Mezní značka

Symbol představuje vstup z vnějšího prostředí do programu nebo výstup z programu do vnějšího prostředí, např. začátek nebo konec programu, zastavení programu nebo začátek a konec samostatně zpracované části programu (např. podprogram) a pod. Podobně jako spojka může mít buď jenom jeden výstup nebo jeden vstup.



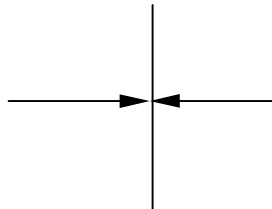
2.1.12 Spojnice

Symbol ve tvaru svislé nebo vodorovné čáry představuje tok dat nebo řízení a slouží ke spojení jednotlivých symbolů ve vývojovém diagramu. Standardní směr toku informací je shora dolů a zleva doprava. To znamená, že spojnice by měly do symbolu vstupovat shora nebo zleva a vystupovat dole nebo

vpravo. Jestliže tok není v těchto uvedených směrech dodržen, doporučuje se tuto spojnici pro označení směru toku informace opatřit plnou nebo otevřenou šipkou. Podobně je možné spojnice opatřit šipkou pro zvýšení názornosti a jasnosti vývojového diagramu (např. při spojování).

Spojnice se mohou podle potřeby spojovat. Pro zvýšení jasnosti by měl být u připojovaných spojnic uveden směr toku informací plnou nebo otevřenou šipkou.

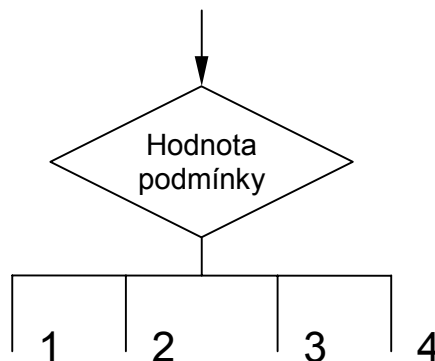
Příklad:



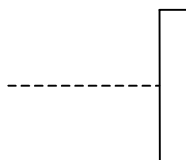
Křížení spojnic se nedoporučuje. Dojde-li ke křížení, neznamená to žádný vzájemný vztah a žádné změny ve směru informací.

Větvení spojnic se používá v případě výstupu z podmíněného symbolu. V tom případě by měla být u každé cesty uvedena hodnota této podmínky.

Příklad:



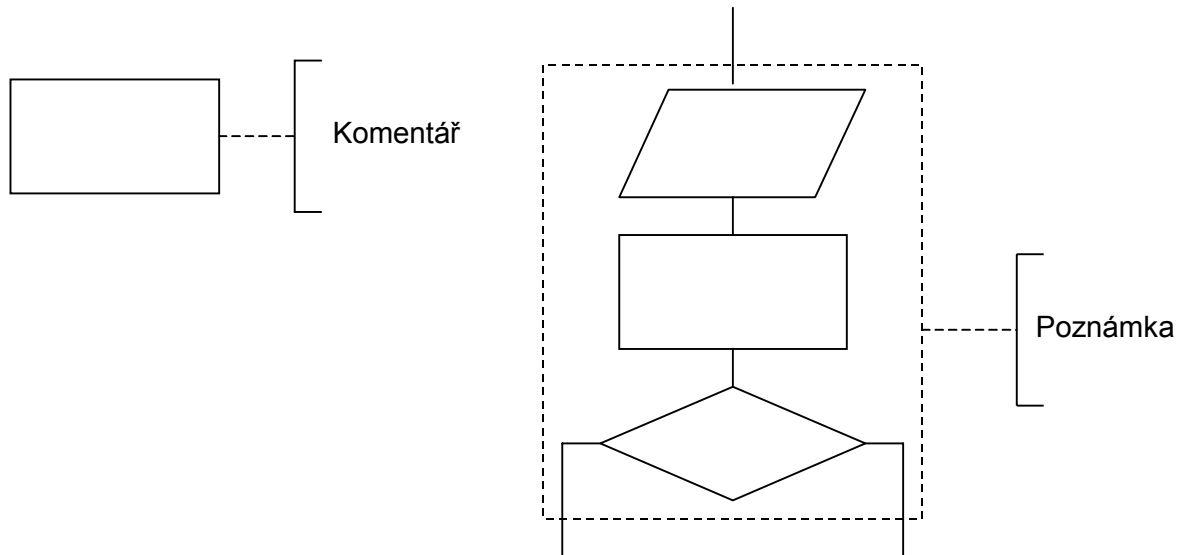
2.1.13 Anotace¹¹



Symbol se používá k připojení popisných komentářů nebo vysvětlujících textů. Přerušovaná čára symbolu anotace je připojena k příslušnému výkonnému symbolu, nebo může být připojena k ohraničené skupině symbolů.

¹¹ Jde o ne zcela vhodný překlad anglického názvu "annotation". Bylo by vhodnější použít české slovo "poznámka" nebo "vysvětlivka" či "komentář", které významově odpovídá obsahu dané činnosti.

Příklady:



Pozn.: Pro nakreslení výše uvedených symbolů byly použity předdefinované automatické tvary textového editoru WORD 97 [2]. Mohou se proto v některých detailech od normou definovaných značek tvarově lišit.

[⇒ Předcházející kapitola](#)

[⇒ Obsah](#)

[⇒ Následující kapitola](#)