|  |  |
| --- | --- |
| **Značky vývojových diagramov** | |
| **Vymedzovacie značky**  Vymedzovacia značka predstavuje vstup z vnútorného prostredia do programu alebo výstup z programu do vnútorného prostredia, napr. začiatok alebo koniec programu. Používa sa tiež pre začiatok a koniec samostatne spracovanej časti programu, má vždy len jednu orientovanú hranu. | |
|  | **Začiatok**  Ak je vymedzovacia značka použitá na začiatku, potom je v nej napísané začiatok. Do značky nesmie vstupovať žiadna hrana a musí z nej vystupovať práve jedna hrana. Pred touto značkou už nesmie byť žiadna ďalšia. |
|  | **Koniec**  Ak je vymedzovacia značka použitá na konci, potom je v nej napísané koniec. Do značky musí vstupovať práve jedna hrana a nesmie z nej vystupovať žiadna hrana. Za touto značkou už nesmie byť žiadna ďalšia. |

|  |  |
| --- | --- |
| **Sekvenčné bloky**  Vyskytujú sa vnútri vývojového diagramu (nemôžu byť použité ako prvé alebo posledné). Označujú sekvenčný postup algoritmom (je možné sa do neho dostať iba z predchádzajúceho prvku a po jeho vykonaní možno postúpiť iba na jeden prvok a to nasledujúci). V ich priebehu nesmie dôjsť k rozvetveniu algoritmu. | |
| A, B | **Vstup** alebo **Výstup**  Pri behu programu je nutné komunikovať s počítačom. Je potrebné:   * Aby sa „dovnútra“ dostali dáta, ktoré program potrebuje k svojej činnosti (buď ich zadá užívateľ z klávesnice, alebo môžu byť načítané zo súboru) * Aby sa nakoniec užívateľ dozvedel výsledky spracovania (môže si ich nechať vypísať na obrazovku počítača, vytlačiť alebo si ich uložiť do súboru)   **Vstup**  Znázorňuje načítanie dát potrebných pre činnosť programu.  **Výstup**  Znázorňuje vypísanie výstupu programu na zobrazovacom zariadení. |
| A A+B | **Spracovanie**  Znázorňuje nejakú činnosť programu, behom ktorej dochádza k transformácií dát (napr. súčet dvoch čísel, alebo iná matematická či logická operácia).  V bloku môže byť zapísaná jedna alebo viac inštrukcií. Každá z týchto inštrukcií však musí byť natoľko podrobná, že ju možno vykonať naraz (nesmie v sebe skrývať niekoľko ďalších operácií).  Pretože spracovanie je tiež sekvenčný blok, musí mať práve jeden vstup a práve jeden výstup. |

|  |  |
| --- | --- |
| **Vetvenie**  V niektorých prípadoch sa nedá postupovať len sekvenčne, ale je potrebné algoritmus rozvetviť. K vetveniu nemôže dochádzať náhodne, ale vždy iba na základe nejakej podmienky. Ak je podmienka splnená, potom program pokračuje jednou cestou, ak nie je splnená, pokračuje druhou cestou.  (napr. ak idete po ceste a uvidíte z nej odbočku do Dolnej Lehoty. Určite neodbočíte náhodne, ale len za podmienky, že potrebujete ísť do Dolnej Lehoty. V opačnom prípade pokračujete ďalej.) | |
| podmienka  + - | **Rozhodovací blok**  Slúži k rozvetveniu programu na základe podmienky, ktorá je uvedená vo vnútri. Ak je podmienka splnená, potom program pokračuje vetvou označenou znamienkom +, ak nie je splnená, potom pokračuje vetvou označenou znamienkom -.  Na obrázku je uvedená jedna z možností zobrazenia vetvenia. Nie je rozhodujúce, či vetva označená znamienkom + bude vľavo a vetva označená – vpravo, alebo to bude naopak. Záleží na prehľadnosti vývojového diagramu. Pokiaľ v niektorej vetve bude napríklad viac príkazov ako v druhej, potom je vhodné kresliť ju na tú stranu, kde je viac miesta. |
| -  podmienka  + | Ďalšia možnosť zobrazenia rozhodovacieho bloku.  Podstatné je, že vstupná hrana musí prichádzať do horného vrcholu značky a značka musí mať práve dva výstupy (označené + a -). |

|  |  |
| --- | --- |
| **Ďalšie značky** | |
| I 1,N | **Príprava**  Označuje prípravnú fázu programu, používa sa napríklad pre zahájenie cyklu so známym počtom opakovaní. |
|  | **Spojka**  Umožňuje spojiť dve časti vývojového diagramu, ktoré nebolo možné nakresliť súvisle (napr. prerušenie na konci stránky alebo z dôvodu krížiacich sa čiar). Spojky na konci prerušenia a na začiatku pokračovania musia byť označené rovnakými číslami. |
| Načítaj maticu A | **Podprogram**  Znázorňuje samostatnú časť programu, ktorá môže obsahovať väčší počet krokov:   * Ak sa vyskytuje v algoritme(programe) nejaká jeho časť na viacerých miestach, potom je vhodné túto časť spracovať samostatne. Označí sa ako jeden blok a ten sa potom vloží do výsledného algoritmu na všetky potrebné miesta * Podobne – ak sa vyskytuje často v rôznych algoritmoch rovnaký postup, je vhodné ho spracovať samostatne, označiť ho jedným blokom a potom ho vložiť do všetkých algoritmov, kde sa vyskytuje.   Ušetrí sa tým práca a algoritmus sa sprehľadní. Následne pri písaní programu sa zjednoduší ich ladenie (chyby sa odstraňujú len jedenkrát). |