

Изявяване на алгоритми чрез блок-схеми

Posted by dbalinov on Ср., 01/28/2009 - 21:03

В ежедневието си човек използва много указания и препоръки, в които е описано как да постъпим за да решим даден проблем. Когато готвим, например, ние използваме рецепта, в която е описано подробно кои продукти да добавим, както и кога да ги добавим. Когато пресичаме улицата, ние също спазваме някаква последователност от правила: спираме, оглеждаме се наляво, после - надясно и накрая пресичаме.

Рецептите, правилата, с които си служим, когато програмираме се наричат алгоритми. Алгоритъмът е съвкупност от указания, които задават последователността от елементарни действия, които се изпълняват, с цел да се получи даден резултат. Елементарни действия са такива действия, които се изпълняват без помощта на други действия.

Алгоритмите могат да се изразяват по различни начини. Като текст, чрез последователност от изображения и други. В информатиката, се използват най-често блок-схеми. Те предават най-точно същността на даден алгоритъм. Изясняват всяка стъпка ясно и недвусмислено. Алгоритъмът може да се предаде и чрез написване на команди на даден програмен език, но за тази цел трябва хората, които ще използват алгоритъма да разбират точно този програмен език. Чрез блок-схемите информацията се предава по универсален начин и лесно можем да напишем кода на езика, който знаем.

Основни блокове

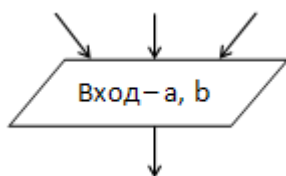
Когато създаваме блок-схеми използваме различни по форма блокове, които характеризират еднотипни по същността си елементарни действия. Имаме блок за начало, за елементарно действие, за логическо условие, за вход, за край и други:

1. Блок за начало



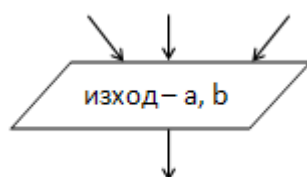
Този блок има форма на елипса, от която излиза само една стрелка, насочена надолу. Всяка блок-схема започва с такъв блок.

2. Блок за вход



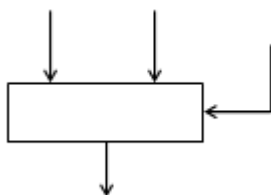
Този блок има форма на успоредник. В него се записват имената на променливите, които се въвеждат по време на изпълнение на програмата. Към блока могат да сочат повече от една стрелка, но излиза само една.

3. Блок за изход



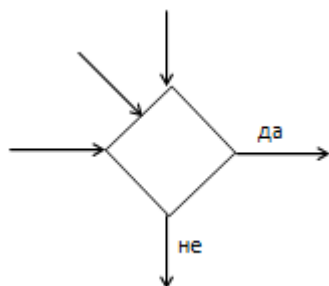
Този блок също има форма на успоредник, подобно на блока за вход. Той съдържа имената на променливите, чийто стойности ще бъдат изведени на екрана. Към него могат да сочат повече от една стрелка, но излиза само една.

4. Блок за елементарно действие



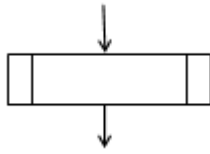
Блоковете от този тип се чертаят като обикновени правоъгълничета, в които се изписва елементарното действие. Към този блок сочи поне една стрелка, а от него излиза винаги само една стрелка.

5. Блок с логическо условие



Блокът за логическо условие представлява правилен ромб, в който влиза поне една стрелка и винаги излизат две стрелки. Над едната от излизащите стрелки пишем „да“ и тя сочи блока, към който ще продължи изпълнението на алгоритъма, ако логическото условие е вярно. Над другата стрелка, която излиза от блока пишем „не“ и тя сочи към блока, който да се изпълни, ако логическото условие е невярно.

6. Блок за обръщане към подалгоритъм



Тези блокове съдържат име на подалгоритъм.

7. Блок за край



В цялата блок-схема има само един блок за край, който има форма на елипса, подобно на блокът за начало. Към този блок сочат една или повече стрелки, а от него не излиза нито една.

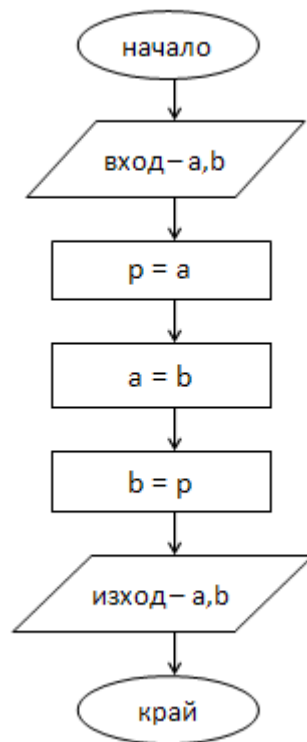
Видове блок-схеми

1. Линейни алгоритми

При линейните алгоритми елементарните действия се изпълняват едно след друго като се достига до крайния резултат. За да демонстрираме линеен алгоритъм ще опишем алгоритъм, който ще разменя стойностите на две променливи.

Какво ще ни трябва:

- Блок за начало
- Блок за въвеждане на стойностите на двете променливи
- Блок, в който ще се разменят стойностите им
- Извеждане на резултат
- Изход



2. Разклонени алгоритми

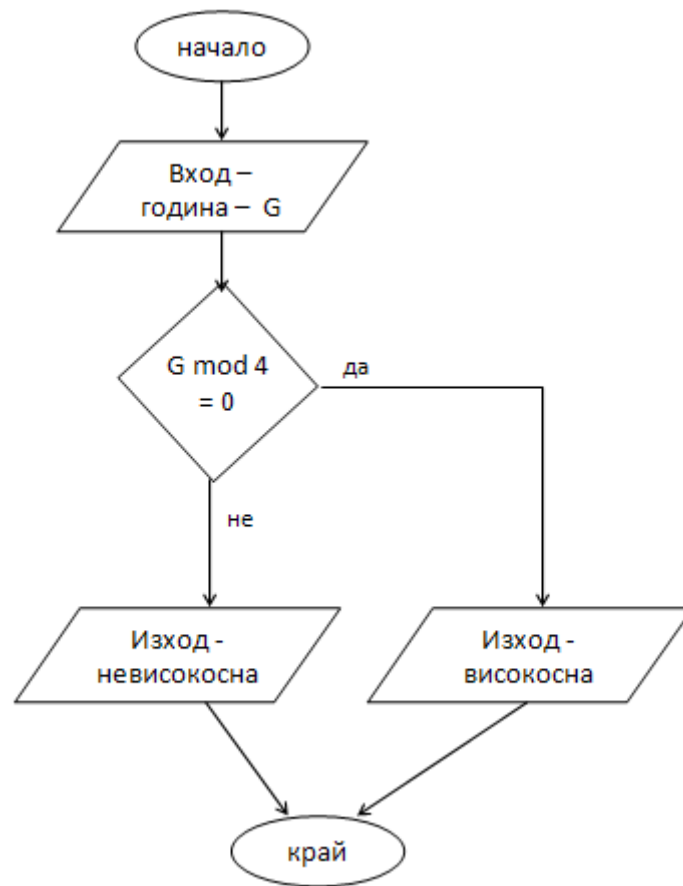
При разклонените алгоритми е характерно, че при определено условие се изпълняват някои елементарни действия, а в противен случай – други.

Нека съставим една блок-схема, която ще описва алгоритъма на програма, проверяваща дали въведена от нас година е високосна. За да бъде една година високосна, тя трябва да се дели на 4 без остатък. В някои програмни езици е прието деленето с остатък да се отбелязва чрез командата MOD.

Какво ще ни трябва:

- Блок за начало
- Блок за въвеждане на годината, която ще проверяваме
- Логически блок в който проверяваме дали годината се дели без остатък
- Извеждане на резултат
- Изход

Сега е време да опишем алгоритъма с блок-схема:



3. Циклични алгоритми

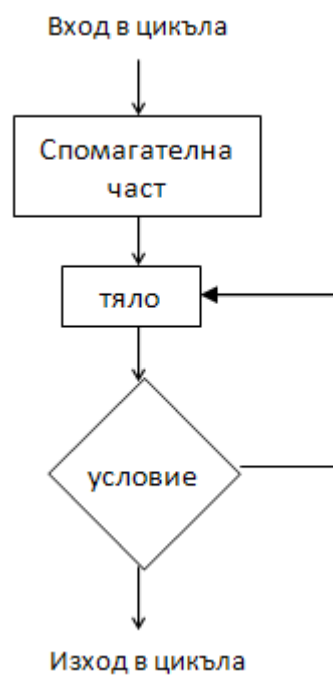
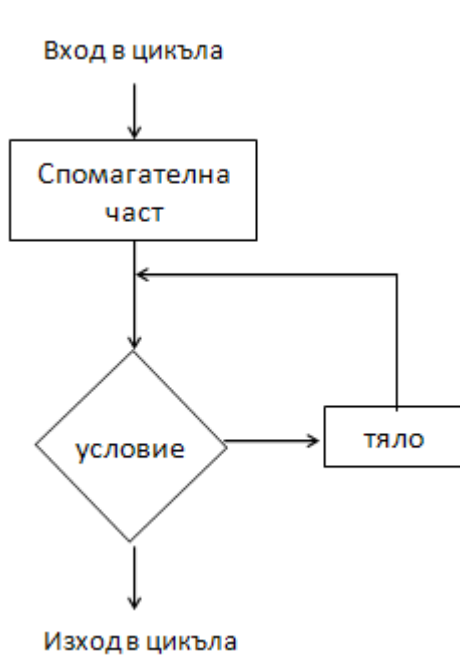
Цикълът е алгоритмична конструкция, чрез която многократно се изпълняват група от елементарни действия. Цикличните конструкции се състоят най-често от три части:

- Спомагателна – съдържа началните стойности на променливите, необходими на цикъла
- Тяло – съдържа групата от конструкции, която ще се повтаря
- Условие за прекратяване или продължаване на цикъла – проверява се необходимостта от следващо изпълнение на цикъла.

Циклите биват два вида:

С предусловие

С постусловие



Тялото може да не се изпълни нито веднъж Тялото ще се изпълни поне веднъж

Сега ще направим блок-схема на алгоритъм, който изчислява сумата на числата от 1 до n , като използваме цикли с пред- и постусловие.

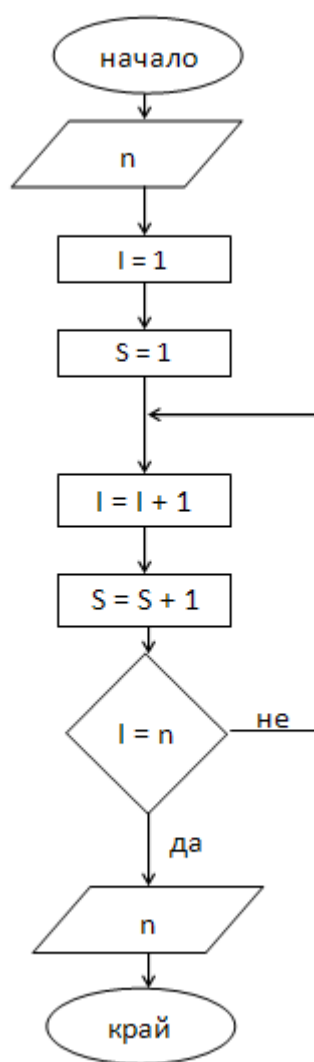
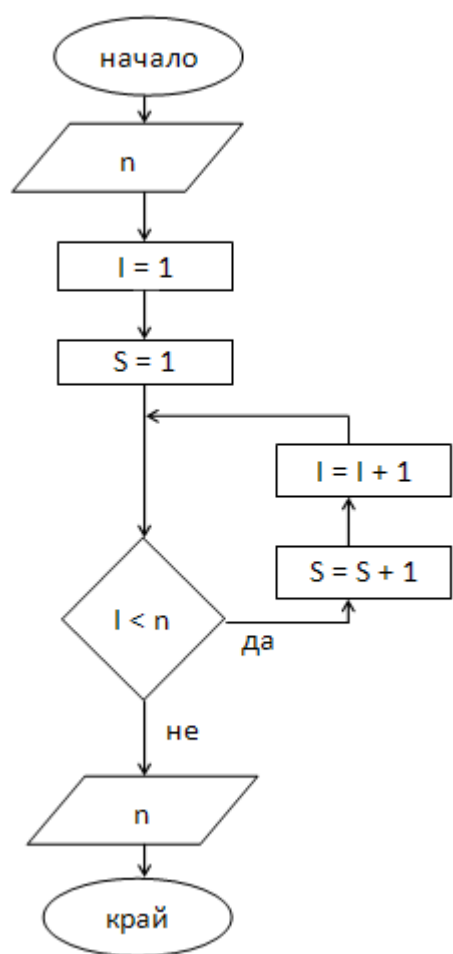
Какво ще ни трябва:

- Блок за начало
- Блок за въвеждане на числото n
- Логически блок в който проверяваме дали годината се дели без остатък
- Извеждане на резултат
- Изход

Ще направим две блок-схеми на този алгоритъм, като едната ще съдържа цикъл с предусловие, а другата - с постусловие.

С предусловие

С постусловие



Автор: Драгомир Балинов