



ХИМИКОТЕХНОЛОГИЧЕН И МЕТАЛУРГИЧЕН УНИВЕРСИТЕТ - СОФИЯ

ИНФОРМАТИКА

част първа

лектор: ас. д-р Фани Томова
лекции: доц. д-р Атанас Атанасов
Катедра “Информатика”

Лекция 6

АЛГОРИТМИ И ПРЕДСТАВЯНЕТО ИМ С БЛОКОВИ СХЕМИ

АЛГОРИТМИ И БЛОКОВИ СХЕМИ

□ Въведение

□ Същност и видове алгоритми

- Числени
- Нечислени
- Диагностични
- Управляващи
- Математически
- За превод от един език на друг

□ Основни свойства на алгоритмите

- Детерминираност
- Резултатност
- Масовост
- Ефективност

АЛГОРИТМИ И БЛОКОВИ СХЕМИ

□ Източници за създаване на алгоритми

- Научни теории
- Експерименти и наблюдения
 - Имитационни
 - Емперични
- Други алгоритми

□ Форми на представяне на алгоритмите

- Словестна
- Текстова
- Графична
 - Блокови схеми, UML диаграми и др.
- Програми на някой от езиците за програмиране

АЛГОРИТМИ И БЛОКОВИ СХЕМИ

□ Блокoви схеми

- Основни блокове
 - За начало, край и пауза
 - За действие
 - За разклонение
 - За вход и изход
 - За процедура
 - Други символи

□ Основни топологични структури

- Линейна
- Разклоненна
- Циклична
- Сложна(комбинация от предходните)

Алгоритми - Въведение

В човешката практика постоянно се използват най-различни **алгоритми**.

Алгоритъмът в най-общ смисъл представлява съвкупност от действия, необходими за решаване на дадена задача.

Алгоритми са:

- Инструкциите за работа с един уред
- Правилата за оказване на медицинска помощ
- Инструкциите за действие при различни бедствия
- Редът за провеждане на химически или физически опити
- Правилата, по които се решават различни математически задачи и др.

Алгоритми - Въведение

Понятието *алгоритъм* се появило най-напред в математиката и се е възприемало, като правило за решаване на множество сходни задачи.

Единна и точна формулировка на понятието не съществува и днес. Неговото масово използване започва с все по-широкото навлизане на компютрите в ежедневната човешка дейност.

По-свободно определение за това понятие може да се формулира по следия начин:

Алгоритъм - това е последователност от краен брой правила или действия, записани в точно определен ред, с помощта на които се решава всяка една задача, принадлежаща на даден клас или тип.

Алгоритми - Въведение

Изпълнител на отделните действия, съдържащи се в алгоритъма, може да бъде:

- човекът в ролята на оператор на сложна апаратура,
- контролорът на качеството на дадено изделие,
- мениджърът в една фирма,
- автоматът, проверяващ пътниците, влизащи в метрото,
- автопилотът в самолета
- стругът с цифрово управление и т. н.

В съвременния свят най-често като изпълнител на алгоритмите се подразбира, че това са компютърните системи.

Същност и видове алгоритми

▪ Числени

Базират се на последователното изпълнение на основните аритметични действия (събиране, изваждане, умножение и деление).

▪ Нечислени

Това са алгоритми в които преобладават логическите действия(сравнения, конюнкции, дизюнкции и отрицания).

Например играта на шах, изход от лабиринт, избор на подходящ вариант и др. се реализират с помощта на такъв вид алгоритми.

Същност и видове алгоритми

■ **Диагностични**

Алгоритми, които са свързани с откриване и отстраняване на неизправности в даден обект (биологичен обект-човек или какъвто и да е друг жив организъм, машина, технологична линия, фирма, организация и т.н.). Наименованието на групата идва от термина “медицинска диагностика”.

■ **Управляващи (регулиращи)**

Такива, които се използват за осигуряване на нормалното функциониране на даден обект. Тук отново понятието обект се разглежда в много широк аспект. Обект може да бъде машина, технологична линия, завод, фирма, организация и т.н .

Същност и видове алгоритми

- **За решаване на математически проблеми**

Това са алгоритмите, реализиращи различните числени методи, които са предмет на изчислителната математиката.

Чрез тях всеки математически метод се свежда до последователност от краен брой математически операции.

Числени методи са разработени за всички математически задачи от инженерно-техническата, икономическата и научна дейности.

Същност и видове алгоритми

- **За превод от един език на друг**

Алгоритми за превод на една програма от един език за програмиране на машинния език на компютъра. С помощта на такива алгоритми се реализират транслаторите.

Алгоритми за превод от един говорим език на друг, които се реализират на мощни КС, защото за реализацията им е необходимо да се съхраняват и обработват големи бази от данни и бази от знания.

Основни свойства на алгоритмите

■ Детерминираност (определеност)

Това означава, че всяко правило, включено в алгоритъма трябва да бъде записано ясно и точно и винаги да предизвиква еднозначно действие.

Изпълнението на даден алгоритъм е точно определен (детерминиран) процес, който не зависи от това кой и кога ще изпълнява алгоритъма.

Важното в случая е, че при едни и същи условия (данни) алгоритъмът винаги трябва да дава един и същи резултат.

■ Резултатност

Алгоритъмът, след краен брой действия, трябва да води до решението на задачата или с други думи след краен брой стъпки алгоритъмът трябва да завършва, като дава един или много резултати.

Основни свойства на алгоритмите

■ Масовост

Всеки един алгоритъм може да се използва за решаването на цял клас от задачи, различаващи се само по входните данни, които ще се използват. Много често входните данни трябва да принадлежат на дадена област, която се нарича област на приложимост на алгоритъма.

■ Ефективност

Алгоритъмът трябва да бъде оптимален (най-добър) по смисъла на предварително избран критерий за оптималност, като минимално време за решаването на задачата, достигане на определена точност, получаване на резултати в определени граници, необходимост от минимална памет и др.

Източници за създаване на алгоритми

■ Научни теории

Коректността на алгоритмите, създадени на базата на научната теория се гарантира от наличието на строги доказателства. Тези алгоритми реализират известни методи от областта на физиката, математиката, техниката, икономиката и др.

■ Експерименти и наблюдения

В тази група се включват алгоритмите, създадени на базата на предварително проведени експерименти и наблюдения. В този случай източник на алгоритмите се явява практиката и получените алгоритми се разделят две групи:

- Имитационни
- Емперични

Източници за създаване на алгоритми

- **Имитационни**

Алгоритъм от имитационен тип се получава, ако при многократно наблюдение на даден субект се опише последователността на неговите действия за решаването на определен проблем. Това означава, че ако друг се опита да изпълни този алгоритъм, то той ще имитира дейността на субекта източник.

- **Емперични**

Ако се наблюдава някакъв процес или явление и подробно се изследва влиянието на отделните фактори, то на тази база могат да се разработят алгоритми за моделиране, изследване и оптимизиране на тези процеси и явления.

Източници за създаване на алгоритми

■ **Базиран на други алгоритми**

Това са алгоритмите, създадени на базата на вече разработени (съществуващи) алгоритми. В този случай, като източник на алгоритми се явяват самите алгоритми. По характерни са следните два случая:

- **Модификация на съществуващи алгоритми**, чрез вмъкване или премахване на отделни части с цел коригиране на техните възможности (разширяване, намаляване или ограничаване).
- **Комбинация от съществуващи алгоритми** за решаване на по-сложни и комплексни и проблеми. За целта се използват готови вече алгоритми и се разработват допълнителни части за тяхното взаимно обвързване и съгласуване на данните.

Форми на представяне на алгоритмите

■ **Словестна**

Алгоритъмът се представя, като последователност от указания в устен вид. Обикновено се използва за обмен на не много сложни и обемисти алгоритми между хората. Словестната форма е неефективна, защото е възможно изпълнителя да не запомни всички указания или да не запомни реда на тяхното следване.

■ **Текстова**

Алгоритъмът се представя, като последователност от указания за действие в писмена форма. Тук се изключват недостатъците на устната форма, но изпълнителят не получава ясна представа за сложността на алгоритъма.

■ **Графична**

Алгоритъмът се представя с блокови схеми, които са обект на следващата част на лекцията.

БЛОКОВИ СХЕМИ

□ Блокoви схеми

- Основни блокове
 - За начало, край и пауза
 - За действие
 - За разклонение
 - За вход и изход
 - За процедура
 - Други символи

□ Основни топологични структури

- Линейна
- Разклоненна
- Циклична
- Сложна (комбинация от предходните)

Блокови схеми

Блоковата схема е графично изображение на структурата на алгоритъма, като за означаването на отделните типове действия се използват точно определени геометрични фигури, които се наричат **блокове**.

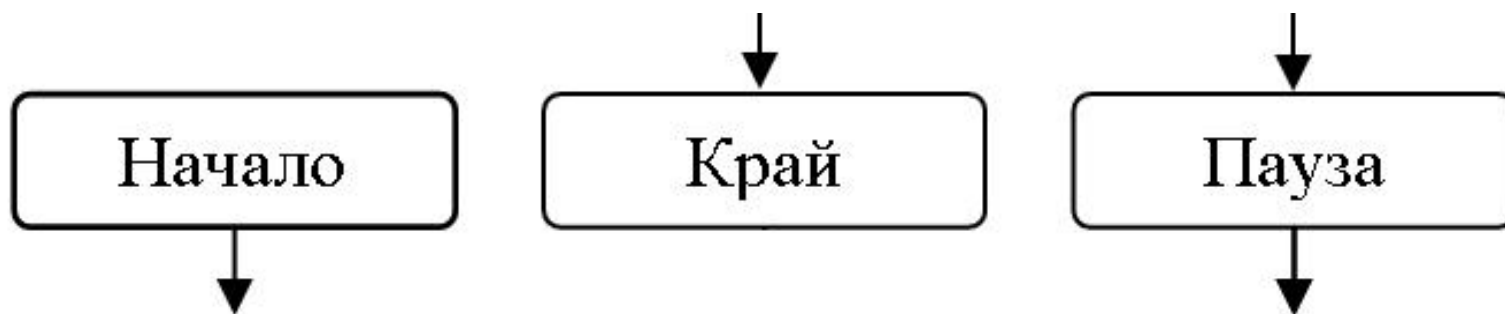
Връзката между отделните **блокове** се осъществява с помощта на **информационни линии**, които показват **посоката на предаване и преработване на данните или информацията**.

Блоковете, които се използват за означаване на отделните типове действия са стандартизирани и са разделени в няколко групи.

Блокови схеми

Блокове за начало, край и пауза

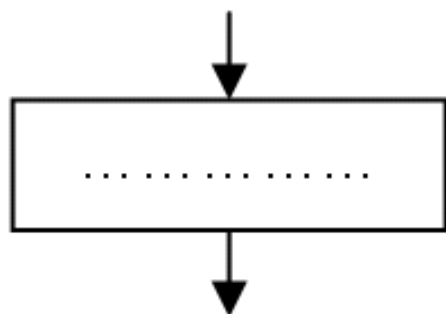
Всяка блокова схема задължително започва с блок за начало и завършва с блок за край. Те се използват еднократно в една блокова схема. С блока за пауза се означава моментно прекъсване в работата на алгоритъма. Този блок може да се използва многократно на различни места в блоковата схема.



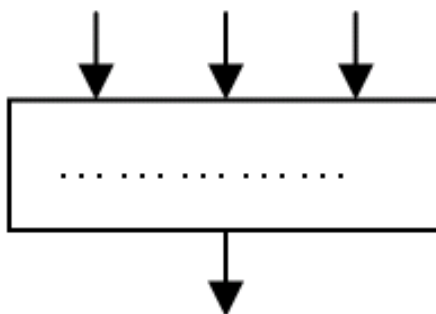
Функционални блокове

Функционален блок за действие

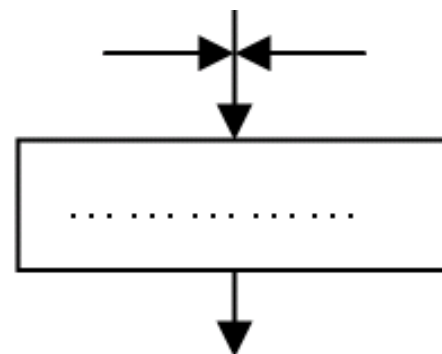
Действието, което се реализира в този блок може да се запише като текст или като математическа формула в зависимост от конкретния случай. За целта се ползва някой от следните блокове, дадени по-долу. Случаят показан на фиг. б може да се сведе до стандартния случай, показан на фиг. а, като предварително се обединят входните информационни линии, като на фиг с.



а.



б.

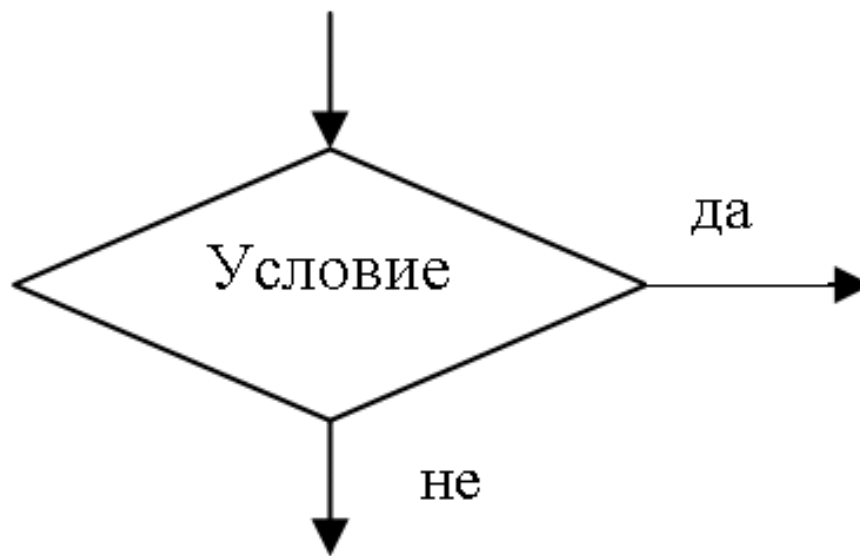


с.

Функционални блокове

Блок за разклонение

Блокът за разклонение има един вход и два изхода. Вътре в блока се записва условието за разклонение във вид на логически израз. Когато условието е изпълнено (да / true), изпълнението на алгоритъма продължава по единия клон, ако не е изпълнено (не / false) по другия клон.

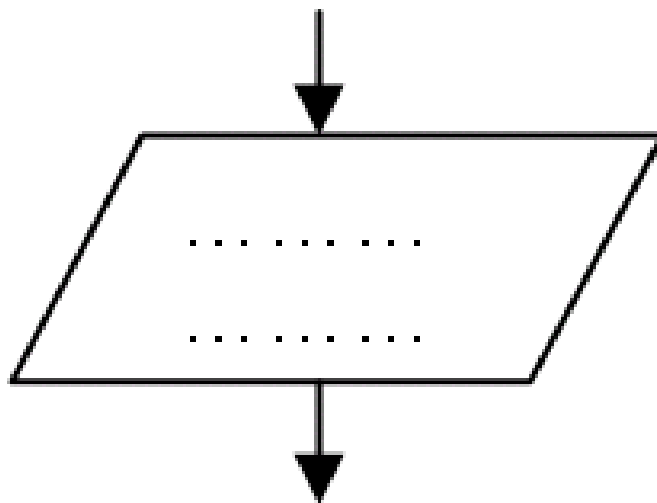


Функционални блокове

Блок за вход и изход

Стандартният блок за означаване на входно-изходни операции е показан на фигурата по-долу.

Конкретното значение на блока, дали е за въвеждане или е за извеждане и каква точно е информацията, която ще се въвежда или извежда се записва вътре в блока.

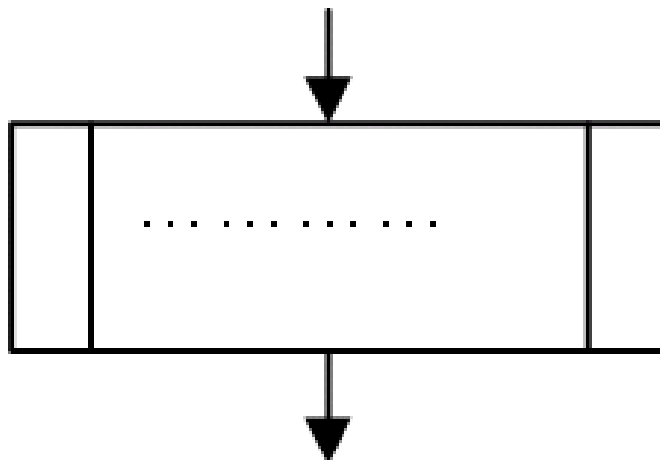


Функционални блокове

Блок за процедура

С този блок се означава относително самостоятелна част от даден алгоритъм, за която вече има разработена подробна блокова схема.

Използването на такива блокове прави блоковата схема по-компактна и сложните и обемисти алгоритми стават по ясни и лесно разбираеми.



Функционални блокове

Информационни линии

Информационните линии се използват за свързване на отделните блокове в блокова схема.

Те показват посоката на предаване и обработка на информацията и са насочени прави линии.

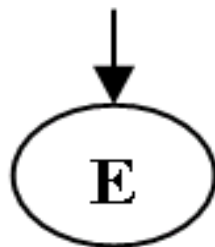
Стандартът препоръчва вертикалните да са насочени от горе надолу, а хоризонталните от ляво надясно.



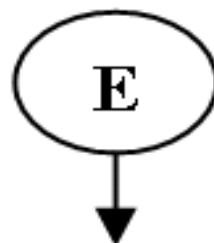
Функционални блокове

Съединители

Съединителите, използвани в блоковите схеми са два вида: - предаващ съединител (фиг. а), и приемащ (фиг. б). Символът **Е** е етикет, с който се означава определена точка от блоковата схема. За етикети най-често се използват букви или цифри. Използването на съединителите се налага, когато алгоритъмът е обемист и неговата блокова схема се разполага на повече от една страница.



а.



б.

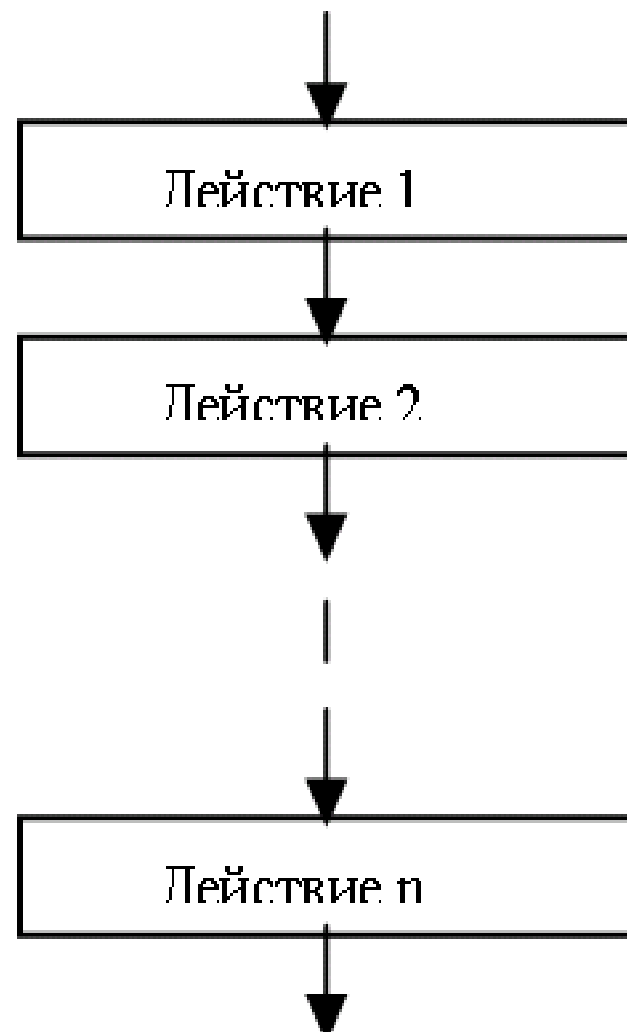
Основни топологични структури

Всеки конкретен алгоритъм се съставя от отделни прости структури, наречени базови топологични структури, като линейна, разклонена и циклична.

Линейна структура

Това е последователност от действия, които се изпълняват еднократно и едно след друго, по реда на тяхното записване, както е показано на фигурата.

Отделните действия се представят с помощта на стандартните функционални блокове за действие или за вход и изход.



Линейна блокова схема

Пример: Да се състави алгоритъм и блок-схема за намиране обема на триъгълна пирамида по зададени страни на основата **a**, **b**, **c** и височина **h**.

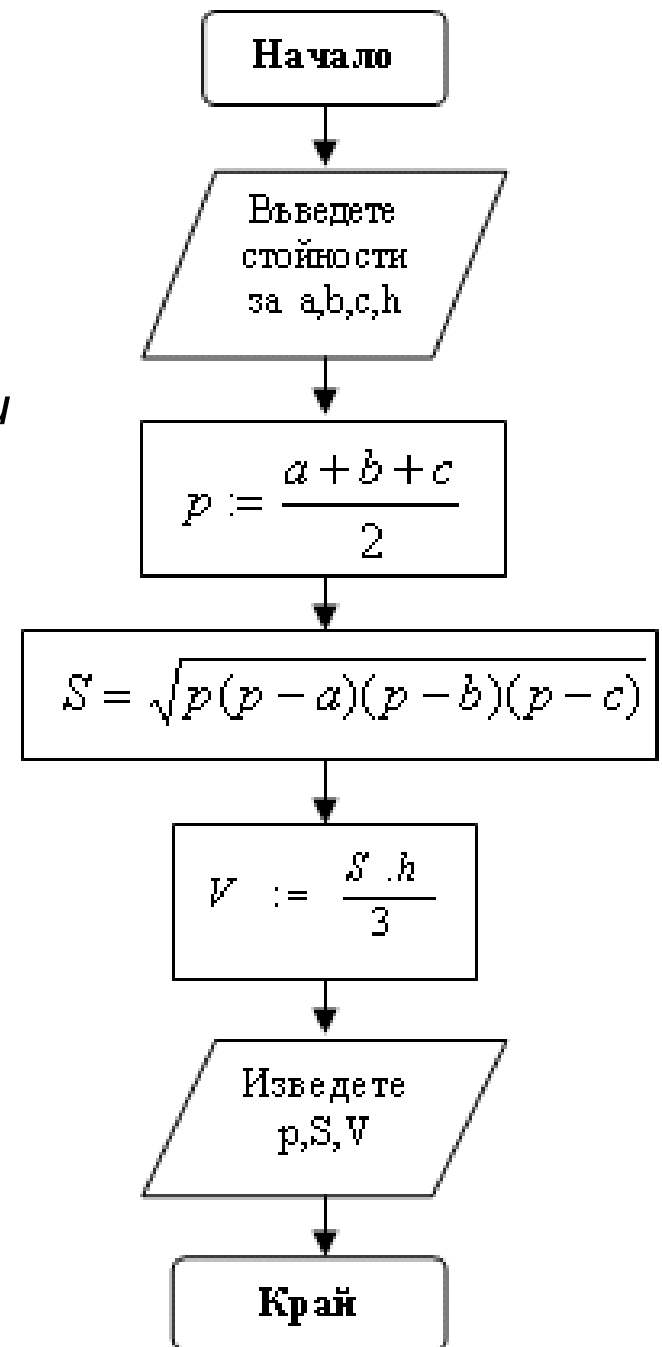
От математиката са известни следните зависимости за решаване на проблема:

$$V = \frac{S \cdot h}{3},$$

където **S** е лицето на основата на пирамидата. За намирането на лице на триъгълник по зададени три страни ще използваме хероновата формула:

$$S = \sqrt{p \cdot (p - a)(p - b)(p - c)}$$

където с **p** е означен полупериметъра на триъгълника. $p = \frac{a+b+c}{2}$,



Разклонени блокови схеми

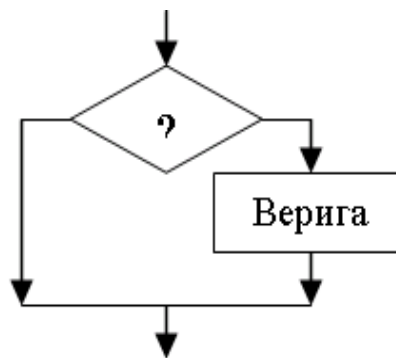
Разклонена структура.

Разклонението е част от алгоритъма, при което в зависимост от изпълнението или неизпълнението на дадено условие, обработката на информацията се разклонява в две посоки (към два клона).

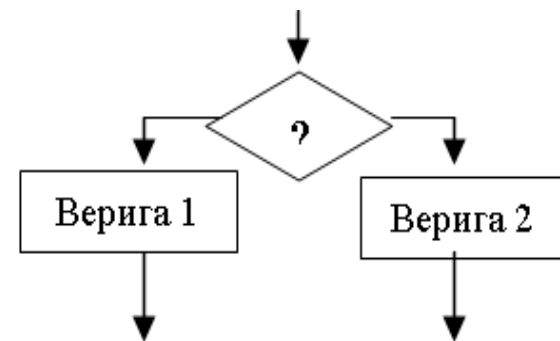
Условието за разклонение се записва вътре в блока и може да бъде под формата на въпрос записан на естествен език или може да се запише като логически израз (отношение, логическа променлива, логически едночлен или многочлен).

Разклонени блокови схеми

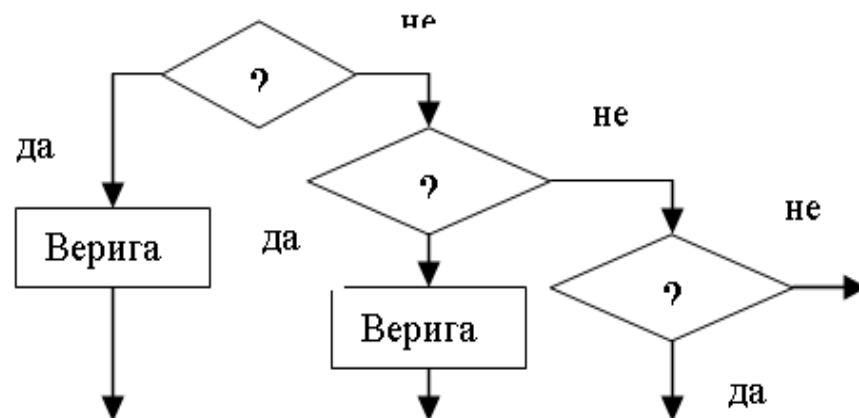
На фигурата са показани три варианта на реализиране на разклонение. На фиг. а е показано разклонение, когато единия клон е празен т.е. липсва верига. На фиг. б е показан случай, когато и в двата клона трябва да се изпълнят определени действия(верига 1 и верига 2). В случая, показан на фиг. с се вижда, че в единия клон вместо верига може отново да има разклонение.



Фиг. а.



Фиг. б.



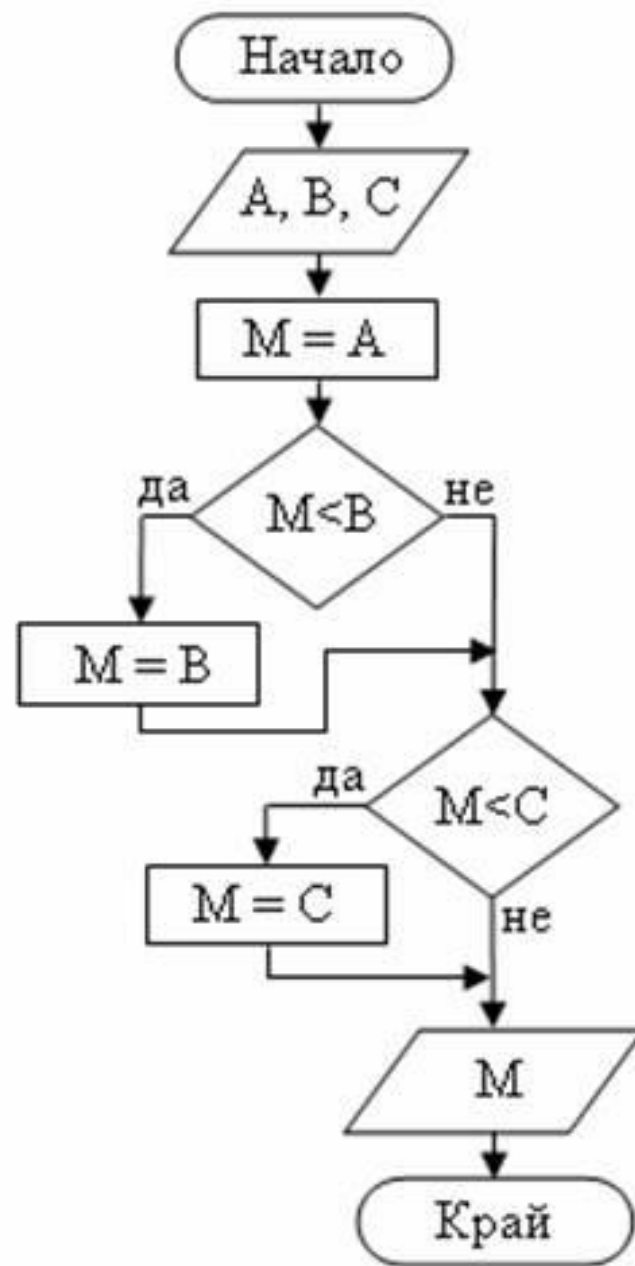
Фиг. с.

Разклонени блокови схеми

Пример: Да се състави алгоритъм и блокова схема за намирането на най-голямото от три числа A, B, C .

Въвеждат се стойности на A, B, C .

На променливата M се задава стойността на A и се проверява дали $M < B$. Ако M (A) е по-малко от B (да), то M приема ($M=B$) стойността на B и се прави преход към следващият блок за сравнение. Ако M е по-голямо от B (не) се прави директно преход към този блок за сравнение. Там M се сравнява с C ($M < C$). Ако M е по-малко от C , то M приема стойността на C . Накрая се разпечатва M , която съдържа най-голямото число.



Циклични блокови схеми

Циклична структура

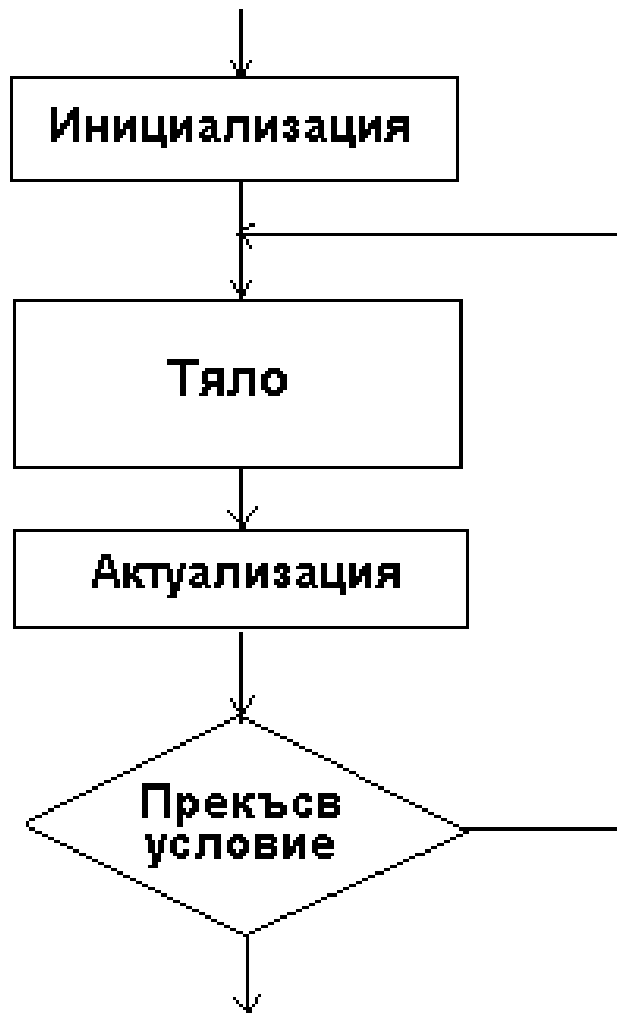
Цикълът е такава част от алгоритъма, която се повтаря многократно с различни входни данни, най-често за различни стойности на една променлива наречена *параметър* на цикъла. В обхвата на цикъла могат да се съдържат и трите основни структури: линейна част(верига), разклонения и цикли.

Независимо каква е конкретната реализация на даден цикъл, то той задължително включва 4 основни части: инициализация, тяло на цикъла, актуализация и прекъсващо условие.

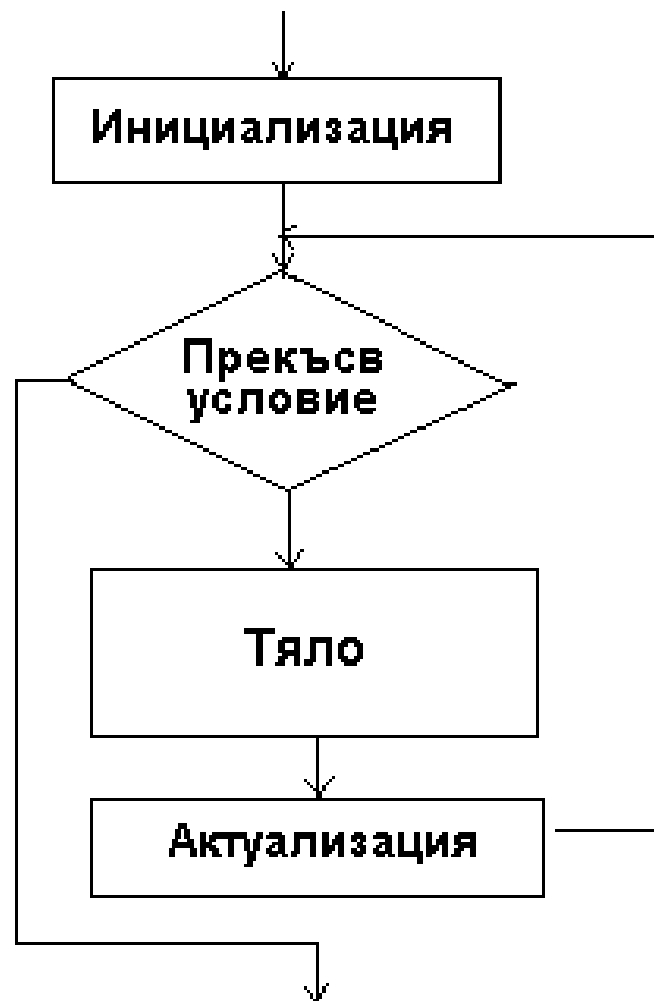
БЛОКОВИ СХЕМИ

Циклична структура

С пост-условие



С предусловие



Циклични блокови схеми

Инициализация. Това означава, че всички променливи, които ще участват в цикличния процес, трябва да са получили подходящи стойности преди неговото започване. Това е особено важно за параметъра на цикъла, защото той играе ролята на управляваща променлива на цикличния процес.

Тяло на цикъла. Съдържа всички действия, които са включени в обхвата на цикъла. Те се изпълняват многократно за всички възможни стойности на параметъра на цикъла.

Актуализация. Тази част подготвя цикъла за всяко следващо изпълнение на тялото. Тя формира новата (следващата) стойност на параметъра на цикъла.

Прекъсващо условие. Това е условието за излизане от цикъла (край на цикъла). Показва, до кога трябва да се изпълнява на тялото на цикъла.

Циклични блокови схеми

Пример: Да се състави алгоритъм и блокова схема за намирането на сумата на първите N числа.

Въвеждат се стойност на N .

Нулира се сумата $Suma=0$;

Инициализира се променливата на цикъла $I = 1$.

Сумата се обновява рекурентно $Suma=Suma + I$.

Актуализира се променливата на цикъла $I = I + 1$

Проверява се прекъсващото условие, дали I е достигнало стойността на N ($I \leq N$).

Ако стойността на I е по-малка от N (да) се повтарят инструкциите в тялото на цикъла.

Ако I е по-голямо от N се излиза от цикъла и се разпечатва сумата $Suma$

