АЛГОРИТМИ

1. Понятия и основни характеристики

А) ОПРЕДЕЛЕНИЕ

Алгоритъм – това е последователност от **елементарни действия** над дадени входни данни, които след изпълнението си дават резултат и приключват за крайно време.

Елементарните действия не изискват допълнително пояснение, наричат се още стъпки.

Алгоритмите се съставят с цел решаване на конкретна задача

Абу Абдаллах (или Абу Джафар) Мухаммад ибн Муса ал-Хорезми (780–847)

Според някои автори думата **„алгоритъм“** произлиза от името на персийския учен Ал Хорезми, роден на територията на днешен Узбекистан. През по-голямата част от живота си е творил в гр. Багдад (столица на Ирак). Има голям принос в развитието на математиката, географията, астрономията и картографията.

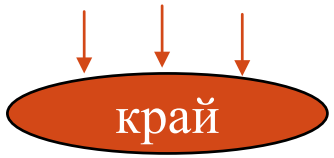
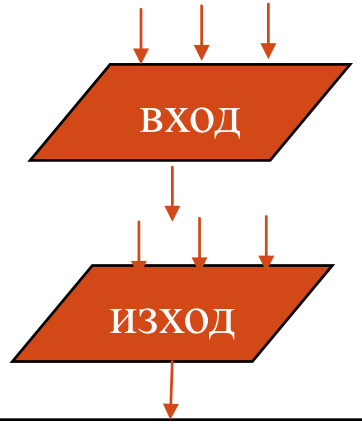
Б) СВОЙСТВА

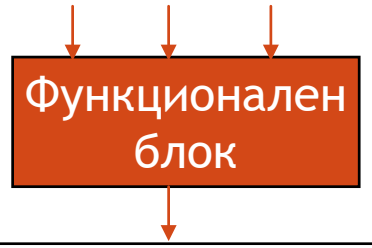
- **определеност (детерминираност)** - алгоритъмът може да се изпълни многократно от различни хора, но за едни и същи начални данни се получава един и същ резултат;
- **масовост** - алгоритъмът трябва да е приложим не за една конкретна задача, а за цял клас еднотипни задачи;
- **крайност** - алгоритъмът се изпълнява за краен брой стъпки и дава краен резултат;
- **резултатност** - след изпълнението на алгоритъма се получава еднозначно определен резултат;
- **дискретност** - всеки алгоритъм се състои от краен брой стъпки, които се изпълняват за крайно време. Всяка стъпка е ясно определена и се изпълнява след приключване на предходната;
- **формалност** - алгоритъмът трябва да се изпълнява стъпка след стъпка. Изпълняващият може да няма представа за решаваната задача и получаваните резултати. Това позволява изпълняващият да е автоматизирано устройство;
- **сложност** - за решаването на някои задачи може да бъдат създадени няколко алгоритъма. Разликата между тях е времето, което им е необходимо за решаване на задачата (обемът на входните данни и необходимата памет също са от значение за компютърните алгоритми). Сложността обикновено се оценява с приблизителния брой стъпки за изпълнение на даден алгоритъм

2. ПРЕДСТАВЯНЕ НА АЛГОРИТМИТЕ

А) СЛОВЕСНО – използва се естествен човешки език. Подобно описание може да има като недостатък многословност, характерна за човешката реч, и двусмисленост при някои стъпки. От друга страна, това е бърз и лесен метод, намиращ широко приложение;

Б) СХЕМАТИЧНО (БЛОК-СХЕМА) – структуриран и точен начин за представяне, като се избягва двусмислието на словесното описание. Ползват се свързани блокове с различна форма и предназначение:

Форма	Име на блока, Пояснение
	Блок за начало – представя се с овал (или правоъгълник със заоблени ъгли). Всеки алгоритъм започва изпълнението си от този блок. От него излиза точно една стрелка
	Блок за край – представя се с овал (или правоъгълник със заоблени ъгли). Определя мястото, където приключва изпълнението на алгоритъма. Всяка блок схема трябва да съдържа поне един такъв блок. От него не излиза стрелка, а към него могат да сочат повече от една
	Входен/Изходен блок – представя се с успоредник. Чрез него се задават входни данни, нужни за изпълнението на алгоритъма, както и се извеждат крайни или междинни резултати. Към блока могат да сочат повече от една стрелки, но излиза точно една

Форма	Име на блока, Пояснение
	Блок за обработка (функционален блок) – представя се с правоъгълник. С негова помощ се извършват аритметични действия. Към него могат да сочат повече от една стрелки, но излиза точно една
	Условен блок (логически блок) – представя се с ромб. В него се записва логическо условие, което ще бъде проверено. Резултатът от проверката може да е: „вярно“ (да, 1) или „невярно“ (не, 0). В зависимост от този резултат изпълнението на алгоритъма ще продължи в направлението на „да“ или „не“. Към него могат да сочат повече от една стрелки, но излизат точно две. За удобство при създаването на блок схема двете изходящи стрелки могат да се поставят в различни комбинации на долните три върха

В) ПСЕВДОКОД – начин за описание на алгоритмите с някакъв език за програмиране, без да се използва конкретен. Може да бъде съчетан с математически формули, както и да се използва част от истински език за програмиране. Не може да бъде изпълнен директно от компютъра. В примера: алгоритъм, който проверява дали дадено число а, може да бъде страна на квадрат или не;

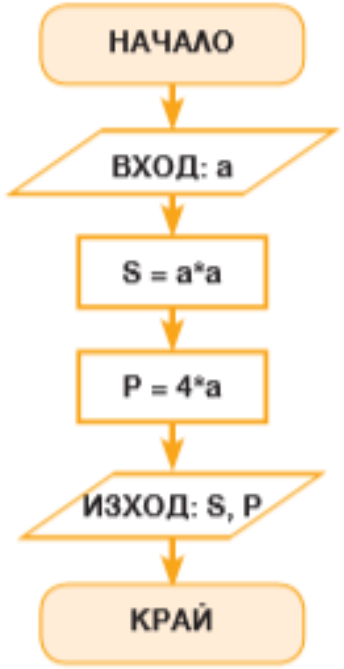
```
Въведи а;
Ако а>0, тогава
    Изведи „Може“;
иначе
    Изведи „Не може“;
Край
```

Г) ЕЗИК ЗА ПРОГРАМИРАНЕ – описанието на алгоритъма става с един от съществуващите езици за програмиране (C/C++, C#, Visual Basic, Pascal, Java и др.), като се спазват неговите правила. Този метод е точен, ясен и не допуска двусмислие, но за прилагането му е необходимо познаването на конкретен език за програмиране. В примера: програма, написана на език C#, която проверява дали дадено число a, може да бъде страна на квадрат или не.

```
float a;
Console.Write("a=");
a = float.Parse(Console.ReadLine());
if(a>0)
    Console.WriteLine("Може");
else
    Console.WriteLine("Не може");
```

2. Видове алгоритми

А) ЛИНЕЙНИ – последователността от действия е винаги една и съща и изпълнението „върви“ по едно направление. Всяка стъпка има точно една предхождаща и точно една следваща я. При тях се използват блокове само от първите четири вида (вж. по-горе);



Линеен алгоритъм за пресмятане на лице и обиколка на квадрат със страна a

Б) РАЗКЛОНЕНИ – последователността в изпълнението на стъпките зависи от верностната стойност на едно или няколко логически условия. За този вид е характерно използването на един или няколко условни блока;

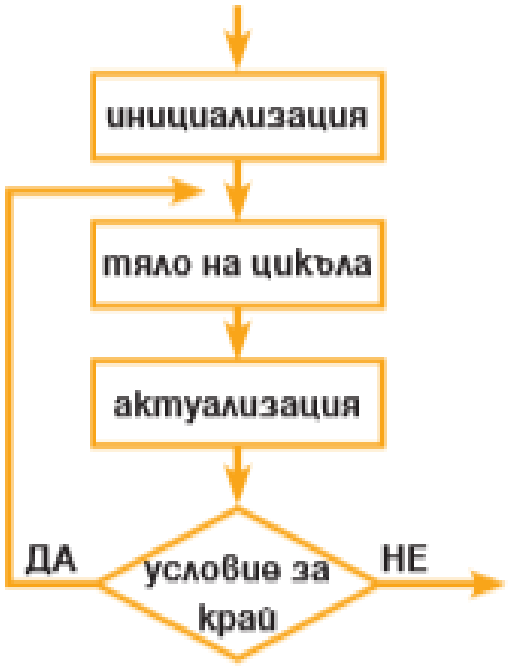
Разклонен алгоритъм, който проверява дали дадено число a може да бъде страна на квадрат или не (т.е. дали може да бъде дължина на отсечка или не)

В) ЦИКЛИЧНИ – едно или няколко действия се повтарят многократно в зависимост от верностната стойност на дадено условие.

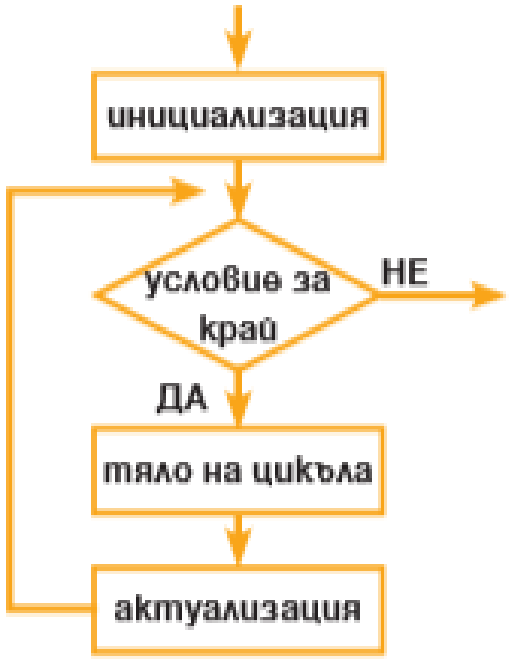
Важни термини, използвани при цикличните алгоритми, са:

- тяло на цикъла – действията, които се повтарят многократно;
- управляваща променлива – променлива, от чиято стойност зависи колко пъти ще се повтори тялото на цикъла. Променливите в езиците за програмиране са аналогични на неизвестните величини в математиката;
- инициализация – задаване на начални стойности на управляващата и други нужни променливи;
- актуализация – подготовка за следващата стъпка от изпълнението на цикъла. Най-често стойността на управляващата променлива се увеличава или намалява;
- условие за край – логическо условие, от което зависи кога цикълът ще приключи изпълнението си.

Съществена разлика при цикличните алгоритми е мястото на условието спрямо тялото на цикъла



Цикличен алгоритъм с постусловие



Цикличен алгоритъм с предусловие