Годишен преговор

Определение за бройна система

Начинът на записване на число чрез краен брой символи, наречени цифри.

- Пример:
  - римската бройна система: XIV --> 14
  - десетична бройна система: 14 --> 14
  - двоична бройна система: 1110 --> 14

Видове бройни системи

- непозиционни – цифрите имат една и съща стойност независимо от позицията им в числото.
  - Пример - римската бройна система:

    XIV --> X=10 I=1 V=5

    XVI --> X=10 I=1 V=5

Видове бройни системи

- позиционна – при нея стойността на цифрата зависи от позицията, на която се намира в числото.
  - Пример - десетичната бройна система:

41 --> 4=40 1=1

14 --> 1=10 4=4

Основа на бройната система

- Броят на различните цифри в записа на числото
  - Пример:
    - десетична бройна система:
      основа: 10, цифри: 0..9, общо 10
    - двоична бройна система:
      основа: 2, цифри: 0 1, общо 2
    - шестнайсетична бройна система:
      основа: 16, цифри: 0..9 A..F, общо 16

Правило за представяне на числа, записани в Q-ична бройна система

- Пример:
  - $365_{(10)} = 300 + 60 + 5 = 3.10^2 + 6.10^1 + 5.10^0$
- Правило:
  - $N_{(q)} = a_n a_{n-1} \ldots a_{0\,(q)} = a_n.q^n + a_{n-1}.q^{n-1} + \ldots + a_0.q^0$
    - $N_{(q)}$ числото в q-ична бройна система
    - $a_n a_{n-1} \ldots a_0$ цифрите на числото
    - q основата на бройната система
- Още примери:
  - $145_{(8)} = 1.8^2 + 4.8^1 + 5.8^0 = 64 + 32 + 5 = 101_{(10)}$
  - $1011_{(2)} = 1.2^3 + 0.2^2 + 1.2^1 + 1.2^0 = 8 + 0 + 2 + 1 = 11_{(10)}$

Единици за измерване на информация

- 1 bit: 0 или 1
- 1 Byte: 8 bit-ово число = $0..255_{(10)}$

  $00000000_{(2)} .. 11111111_{(2)}$
- 1 KB: 1024 Byte
- 1 MB: 1024 KB
- 1 GB: 1024 MB
- 1 TB: 1024 GB
- 1 PB: 1024 TB

Двоична бройна система

- Има две цифри – 0 и 1

- Предимство – лесно е да бъде записана информацията, ако има само две възможни стойности:
  - наличие / отсъствие на перфорация
  - наличие / отсъствие на електрическо напрежение
  - наличие / отсъствие на заряд
  - положително / отрицателно намагнитяване
  - наличие / отсъствие на прогаряне

Шестнайсетична бройна система

- Има 16 цифри – 0..9, A..F
  - A – 10
  - B - 11
  - C - 12
  - D - 13
  - E - 14
  - F - 15

- Предимство – по-кратък запис от двоичната бройна система и лесно преобразуване от и към нея

- Пример – $10100000_{(2)} = A0_{(16)} = 160_{(10)}$

Съждение

Изречение, което може да бъде проверено като вярно или невярно.

- Пример за съждения
  - Сега е есен.
  - Ние сме на Луната.
- Изречения, които не са съждения
  - Колко е часът?
  - Петре, излез навън!

Видове съждения

- прости – съждения, които не са съставени от други съждения.
  - Пример – Днес е вторник.

- сложни – съждения, които са образувани от прости съждения и съждителни операции.
  - Пример - Това е презентация и вие я гледате.

Логическо отрицание (инверсия)

Такова съждение, което е вярно тогава и само тогава, когато съждението, към което е приложено, е невярно.

- Записване:

  NOT X, ¬X

- Пример:

  Ние се намираме на морето.

  Ние **НЕ** се намираме на морето.

Логическо умножение (конюнкция)

Такова съждение, което е вярно тогава и само тогава, когато са верни и двете съждения, към които е приложена операцията.

- Записване:

  X AND Y, X ^ Y

- Пример:

  Вие сте в 9-ти клас **И** сега сме в училище.

  Вие сте в 9-ти клас **И** сега сме в парка.

  Вие сте в 12-ти клас **И** сега сме в парка.

Логическо събиране (дизюнкция)

Такова съждение, което е вярно тогава и само тогава, когато е вярно поне едно от съжденията, към които е приложена операцията.

- Записване:

  X OR Y, X v Y

- Пример:

  Вие сте в 9-ти клас **ИЛИ** вие не сте ученици.

  Вие сте в 9-ти клас **ИЛИ** сега сме в училище.

  Вие сте в 12-ти клас **ИЛИ** сега сме в парка.

Логически изрази

Представляват комбинация от съждителни променливи, константи и операции между тях

- Пример:

  (1 AND X) OR Y

  (X OR Y) OR (1 OR 0)

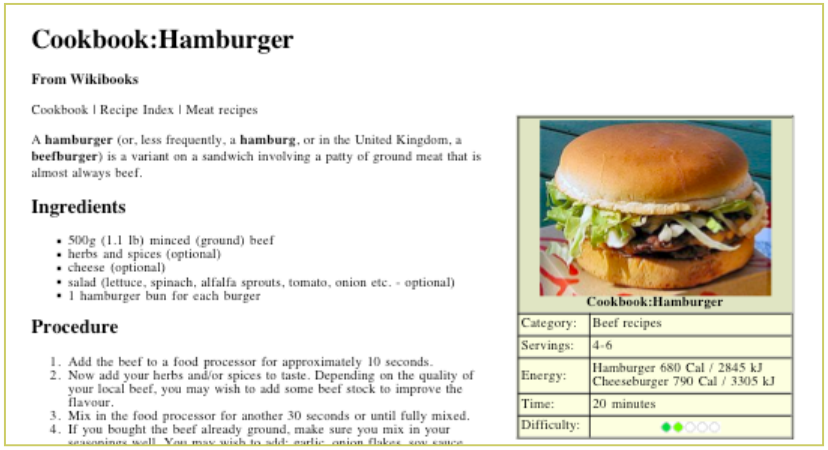
  (NOT X) AND X

  (NOT Y) OR Y

Определение за алгоритъм

Поредица от указания, определящи елементарни действия, изпълнението на които води до решаването на проблема, поставен пред алгоритъма.

Понятието алгоритъм е сходно по значение с понятията метод, упътване, рецепта...



Cookbook:Hamburger

From Wikibooks

Cookbook | Recipe Index | Meat recipes

A **hamburger** (or, less frequently, a **hamburg**, or in the United Kingdom, a **beefburger**) is a variant on a sandwich involving a patty of ground meat that is almost always beef.

Ingredients

- 500g (1.1 lb) minced (ground) beef
- herbs and spices (optional)
- cheese (optional)
- salad (lettuce, spinach, alfalfa sprouts, tomato, onion etc. - optional)
- 1 hamburger bun for each burger

Procedure

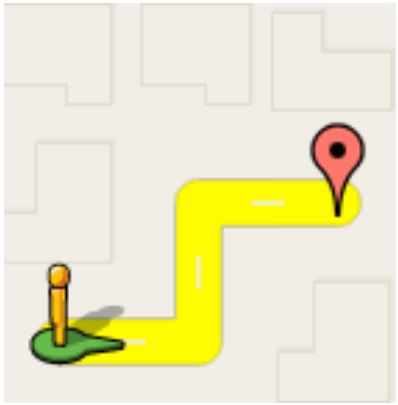
1. Add the beef to a food processor for approximately 10 seconds.
2. Now add your herbs and/or spices to taste. Depending on the quality of your local beef, you may wish to add some beef stock to improve the flavour.
3. Mix in the food processor for another 30 seconds or until fully mixed.
4. If you bought the beef already ground, make sure you mix in your

Cookbook:Hamburger

Category:	Beef recipes
Servings:	4-6
Energy:	Hamburger 680 Cal / 2845 kJ Cheeseburger 790 Cal / 3305 kJ
Time:	20 minutes
Difficulty:	

Субекти при работата с алгоритми

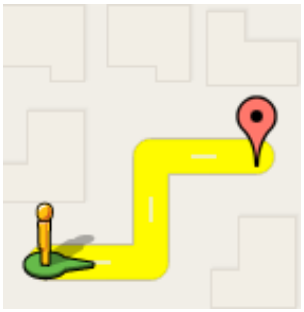
- Съставител – този, който описва указанията на алгоритъма така, че да решат проблема
  - обикновено това е човек
- Изпълнител – този, който извършва действия според указанията в алгоритъма
  - може да е човек или компютър

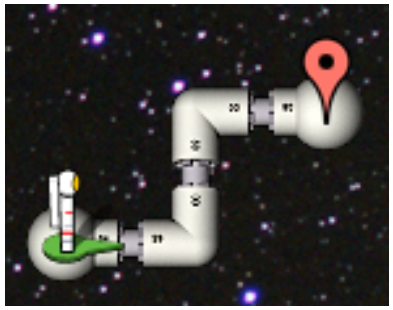


Етапи в живота на алгоритъм

- Описание – когато съставителят описва онова, което трябва да бъде извършено
- Изпълнение – когато изпълнителят извършва действия според указанията в алгоритъма

- Един алгоритъм, ако е проектиран добре, може да бъде изпълняван многократно от множество различни изпълнители, винаги с еднакъв резултат

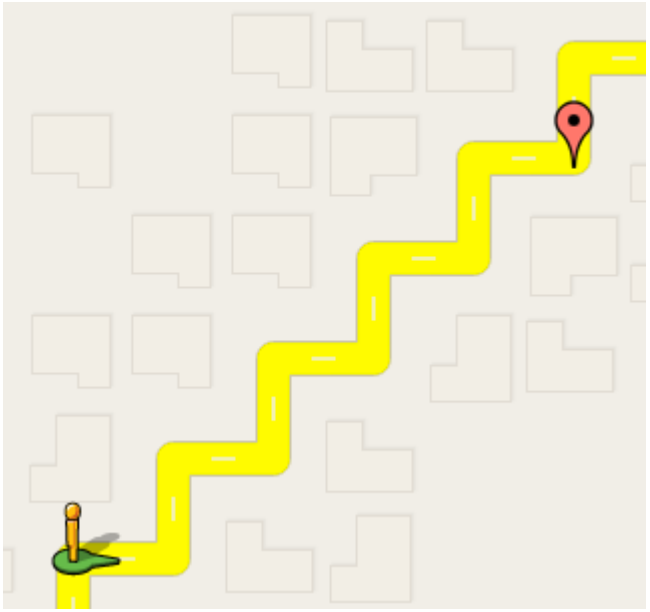




Елементарно действие

Елементарно действие – такова действие, което може да бъде извършено от изпълнителя без допълнителни указания

- Напред!
- Наляво!
- Надясно!



Видове алгоритми

- линейни – при тях броят на указанията е равен на броят на действията, които ще бъдат извършени;
- разклонени (условни) – броят на указанията е по-голям от броят на действията, които ще бъдат извършени, защото някои от действията ще бъдат пропуснати;
- циклични – броят на указанията е по-малък от броят на действията, които ще бъдат извършени, тъй като някои от действията ще се повтарят многократно.

Величини

Величините са стойности, които се използват в описанието на алгоритъма. Биват:

- константи: величини, които НЕ МОГАТ да променят стойността си по време на изпълнението на алгоритъма
- променливи: величини, които МОГАТ да променят стойността си по време на изпълнението на алгоритъма

Величините се характеризират с:

- име: обикновено се състои от латински букви и евентуално цифри
- тип на данните – т.е. каква информация съхраняват

Тип данни

Определя множеството от допустимите стойности, които може да приема дадена величина и операциите, в които може да участва тя.

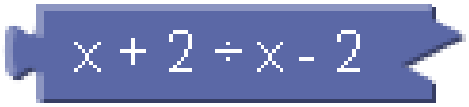
Например:

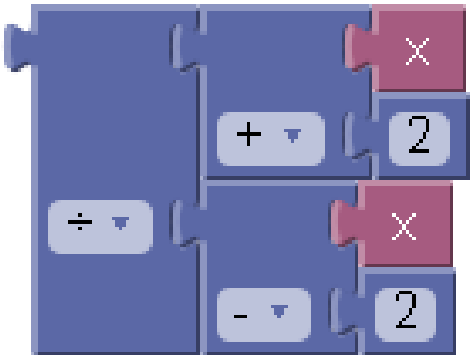
- целочислен тип данни: състои се от **цели числа**; валидни операции – **събиране**, **изваждане**, **умножение**, **деление**
- булев тип данни: има само две възможни стойности – **вярно** и **невярно**. Операциите са **И**, **ИЛИ**, **НЕ**
- текстов тип данни: съдържа текст. Операция - слепване

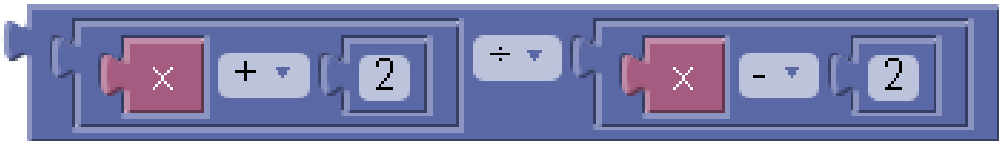
Изрази

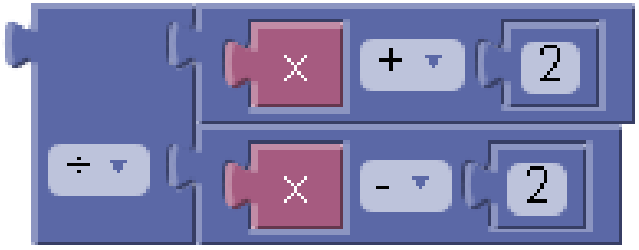
Комбинация от величини и валидни операции между тях. В изразите величините участват със своите **имена**, а операциите се извършват върху **стойностите им**.

- Например Y = (X + 2) / (X – 2)
  - за X=3 Y=?
  - за X=4 Y=?









Команда за присвояване

- предназначение: за указване на стойност на величината по време на *описанието* на алгоритъма.
- словесно представяне:
  променлива:= израз

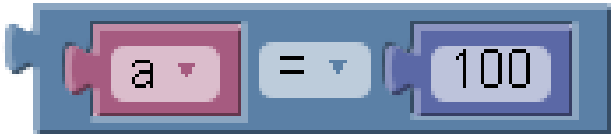
- действие: изчислява се *израз* и стойността му се присвоява на *променлива*

Разлика между присвояване и равенство

- с командата присвояване указваме стойност на дадена величина

a := 100

- с операцията равенство проверяваме дали две величини имат равни стойности

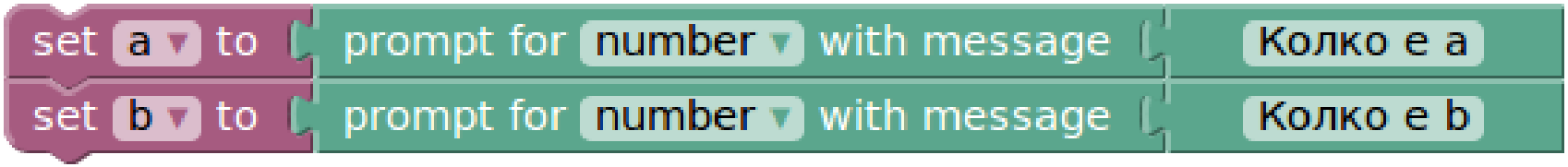


Команда за въвеждане

- предназначение: за указване на стойност на величина по време на изпълнението на алгоритъма.
- словесно представяне:

  въведи *променлива*

- действие: изпълнението на алгоритъмът спира, докато не бъде въведена стойност. Тя се присвоява на *променлива.*



Прилики и разлики между командите за въвеждане и за присвояване

И двете служат за указване на стойност на величина:

- командата за присвояване - по време на *описанието* на алгоритъма
- командата за въвеждане - по време на *изпълнението*.

Кога коя команда се използва:

- Когато стойността на променливата е винаги една и съща или се изчислява чрез израз, използваме команда за присвояване.
- Когато искаме всеки път при изпълнение да питаме каква да бъде стойността на променливата, използваме команда за въвеждане.

Команда за извеждане

- предназначение: за съобщаване на стойност по време на изпълнението на алгоритъма.
- словесно представяне:

  изведи израз

- действие: изчислява се стойността на израза и тя се извежда или съобщава.

Условна команда

Позволява разклоняване на алгоритъма в зависимост от стойността на някакво условие.

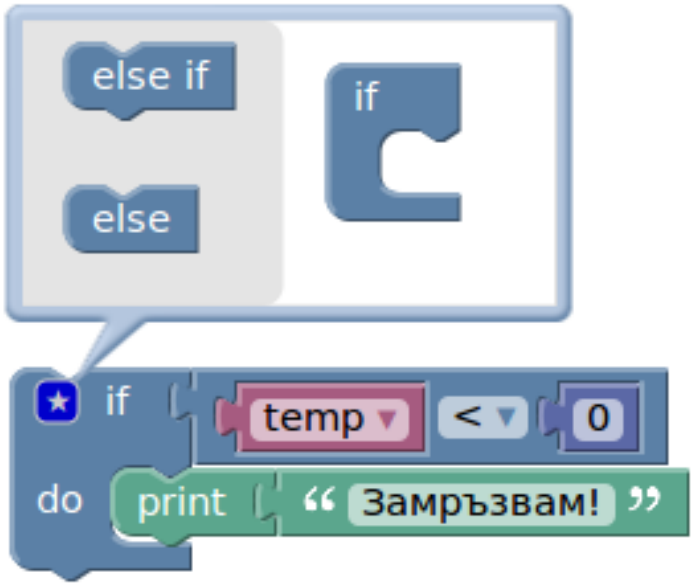
Само командите в избраното разклонение се изпълняват; командите от другите разклонения се игнорират.

Кратка форма на условната команда

- словесно представяне:

  ако *условие* тогава
  команда1

- действие: проверява се *условието* – ако е вярно, се изпълнява *команда1*, ако не е вярно – не се изпълнява нищо

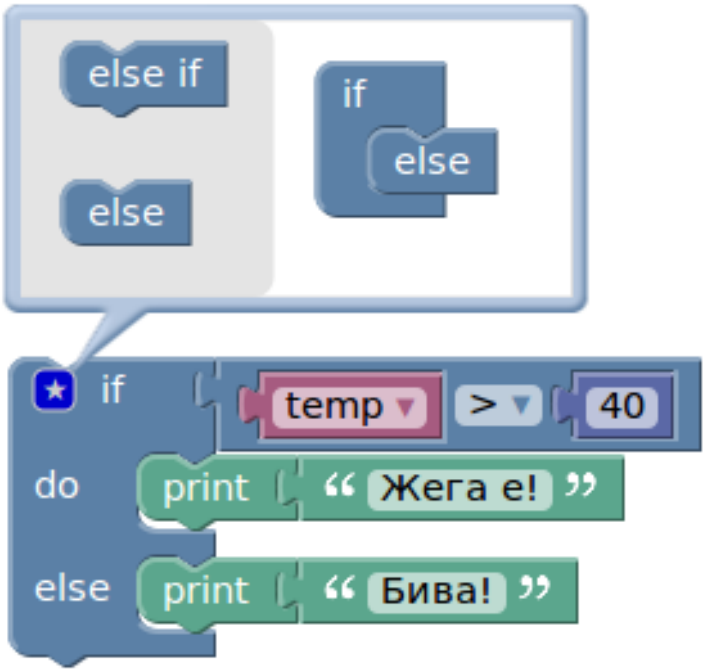


Пълна форма на условната команда

- словесно представяне:

  ако *условие* тогава
  команда1
  иначе *команда2*

- действие: проверява се *условието* – ако е вярно, се изпълнява *команда1*, ако не е вярно – *команда2*

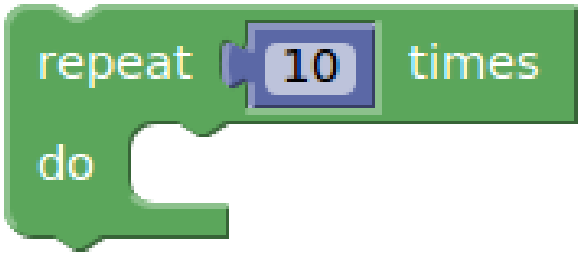


Определение за цикъл

Група от действия, които се повтарят многократно.
Едно завъртане на цикъла се нарича итерация.

Команда за броячен цикъл

- предназначение: за повторение на група команди определен брой пъти



- словесно представяне:

  повтори N пъти *команди*

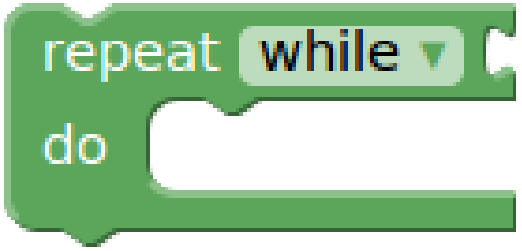
- действие: командите се повтарят указаният брой пъти.

Команда за цикъл с предусловие

- предназначение: за повторение на група команди докато **е** изпълнено условие. Условието се проверява **преди** всяко повторение.

- словесно представяне:

  докато *условие*
  повтаряй *команди*



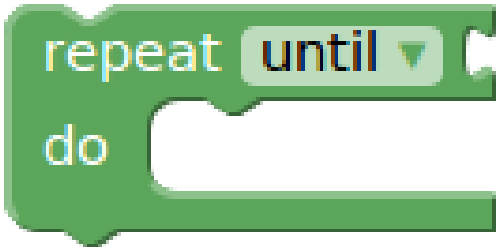
- действие: командите се повтарят, докато условието **е вярно**. Ако още отначало не е, командите няма да се изпълнят **нито веднъж**.

Команда за цикъл с постусловие

- предназначение: за повторение на група команди докато **се** изпълни условие. Условието се проверява **след** всяко повторение.

- словесно представяне:

  повтаряй *команди*
  докато *условие*



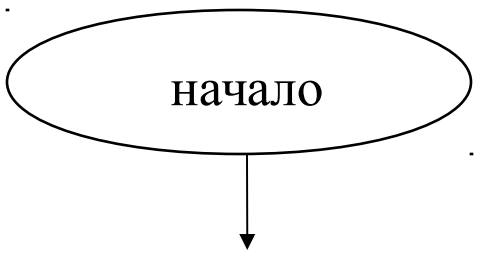
- действие: командите се повтарят, докато условието **стане вярно**. Дори още отначало да е вярно, командите ще се изпълнят задължително **поне веднъж**.

Видове описание на алгоритмите

- Словесно – чрез команди на близък до естествения език
  - предимство: достъпност
- Блок-схеми – чрез набор от геометрични фигури и свързващи стрелки между тях
  - предимство: прегледност
- Програмни езици – чрез изкуствен език, съставен от фиксиран набор стандартни думи и строги правила за употребата им
  - предимство: може да бъде изпълнявано от компютрите

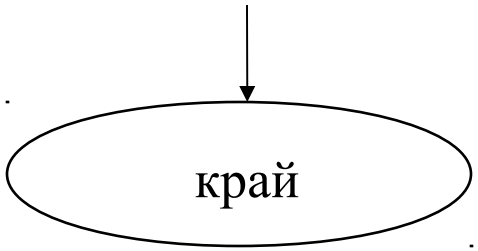
Основни блокове

- Блок за начало - указва откъде започва изпълнението на алгоритъма; в блок-схемата има само един такъв блок



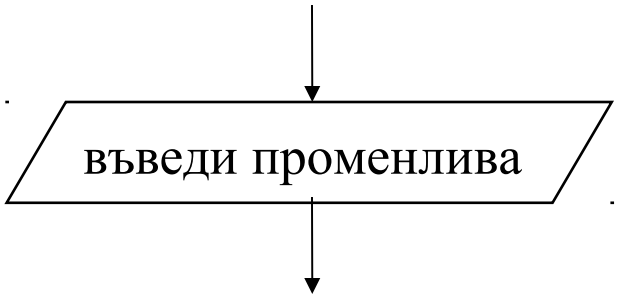
Основни блокове

- Блок за край - указва къде завършва изпълнението на алгоритъма; може да има повече от един блок за край



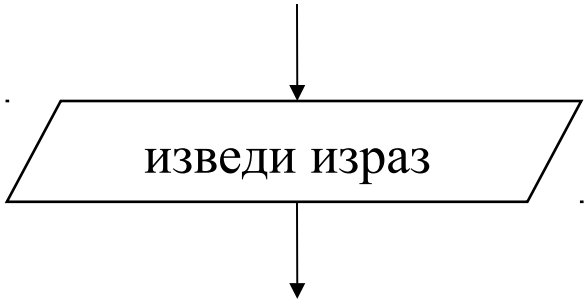
Основни блокове

- Блок за въвеждане - съдържа имена на величини, които получават стойност по време на изпълнението на алгоритъма



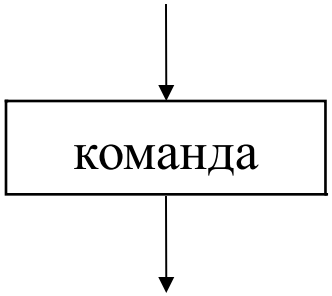
Основни блокове

- Блок за извеждане – служи за извеждане на резултат от работата на алгоритъма



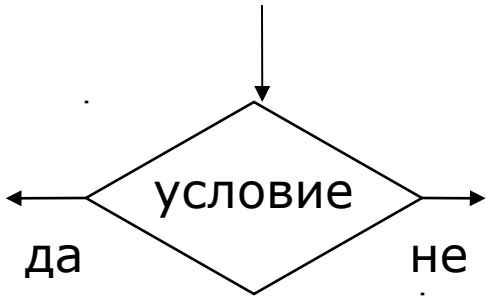
Основни блокове

- Блок за обработка - указва едно или повече елементарни действия, които трябва да бъдат извършени



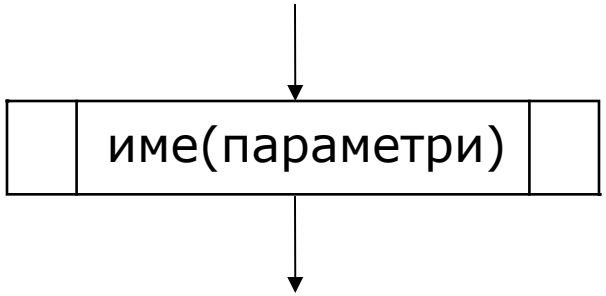
Основни блокове

- Условен блок – служи за разклоняване на алгоритъма в зависимост от стойността на някакво условие



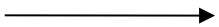
Основни блокове

- Блок за обръщение към подалгоритъм
  - съдържа име на друг алгоритъм, който ще бъде извикан и списък от параметри, които ще му бъдат подадени преди изпълнението



Основни блокове

- Свързващи стрелки - указват реда на изпълнение на блоковете



Правила при съставяне на блок-схеми

- Всяка блок-схема има точно едно начало и поне един край
- До всеки блок трябва да има път от началото и от всеки блок трябва да има път до края
- Във всеки блок (без блока за начало) влиза поне една свързваща стрелка и от всеки блок (без блока за край и условния) излиза точно една стрелка
- Свързващите стрелки трябва да са по възможност хоризонтални или вертикални

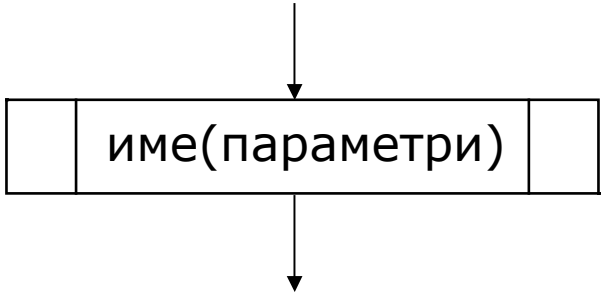
Подалгоритъм

- Определение – алгоритъм, който се използва в описанието на друг алгоритъм

Команда за извикване на подалгоритъм

- Предназначение - за извикване на един алгоритъм от друг алгоритъм.
- Общ вид:

  изпълни *подалгоритъм(параметри)*
  - Подалгоритъм – име на подалгоритъма, който ще се изпълнява
  - Списък с параметри - набор от стойности, които се предават при изпълнението му
- Действие - подалгоритъмът се изпълнява със стойностите, указани в параметрите



Край

... че ваканцията наближава...