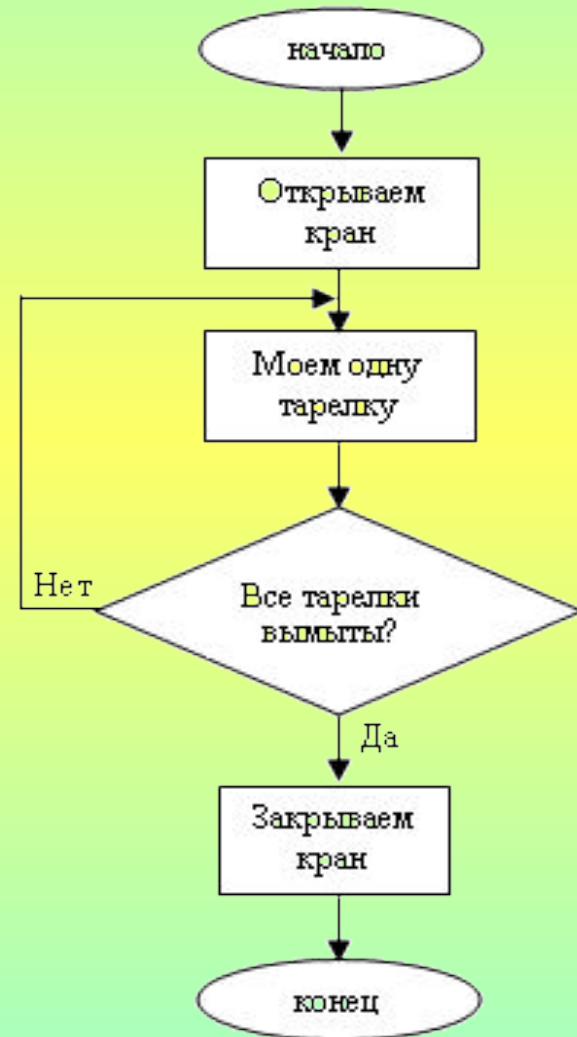


Алгоритм и его формальное исполнение

Алгоритм

- **Алгоритм** – это предназначенное для конкретного исполнителя точное описание последовательности действий, направленных на решение поставленной задачи.



Свойства алгоритма

✓ **Дискретность** - разделение алгоритма на последовательность шагов.

Пример: Алгоритмы кулинарных рецептов состоят из отдельных действий, которые обычно нумеруются.

✓ **Результативность** - получение из исходных данных результата за конечное число шагов.

Пример: Алгоритм всегда приводит к результату, алгоритм покраски забора.

Свойства алгоритма

- ✓ **Массовость** - возможность применения алгоритма к большому количеству различных исходных данных.

Пример: Алгоритмы сложения, вычитания, умножения и деления.

- ✓ **Детерминированность (определённость)** - исполнитель должен выполнять команды алгоритма в строго определенной последовательности.

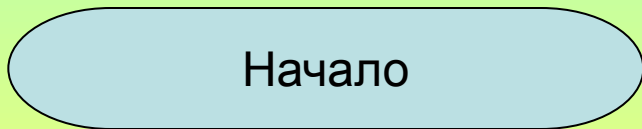
Пример: алгоритм управления самолётом.

Свойства алгоритма

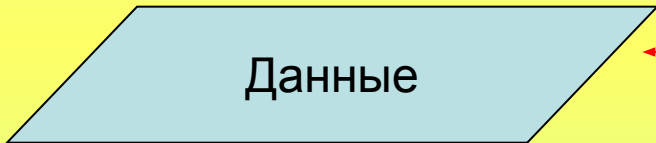
- ✓ **Выполнимость и понятность** - алгоритм должен содержать команды, входящие в систему команд исполнителя и записанные на понятном исполнителю языке.

Пример: алгоритм включения компьютера.

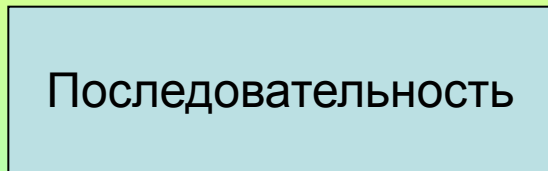
Блок-схемы



→ Прямоугольник с закругленными углами, применяется для обозначения начала или конца алгоритма

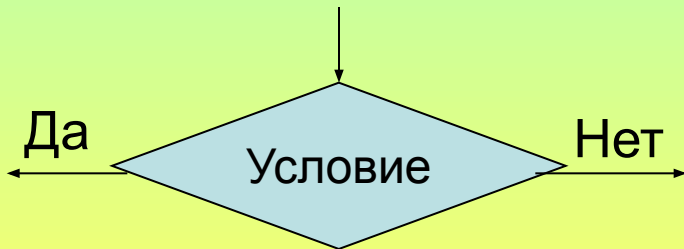


→ Параллелограмм, предназначен для описания ввода или вывода данных, имеет один вход сверху и один выход внизу

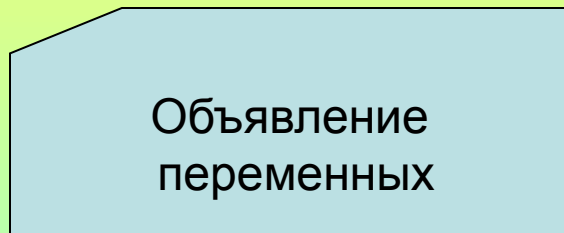


→ Прямоугольник, применяется для описания линейной последовательности команд, имеет один вход сверху и один выход внизу

Блок-схемы



Ромб, служит для обозначения условий в алгоритмических структурах «ветвление» и «выбор», имеет один вход сверху и два выхода (налево, если условие выполняется, и направо, если условие не выполняется)



Прямоугольник со срезанным углом, применяется для объявления переменных или ввода комментариев

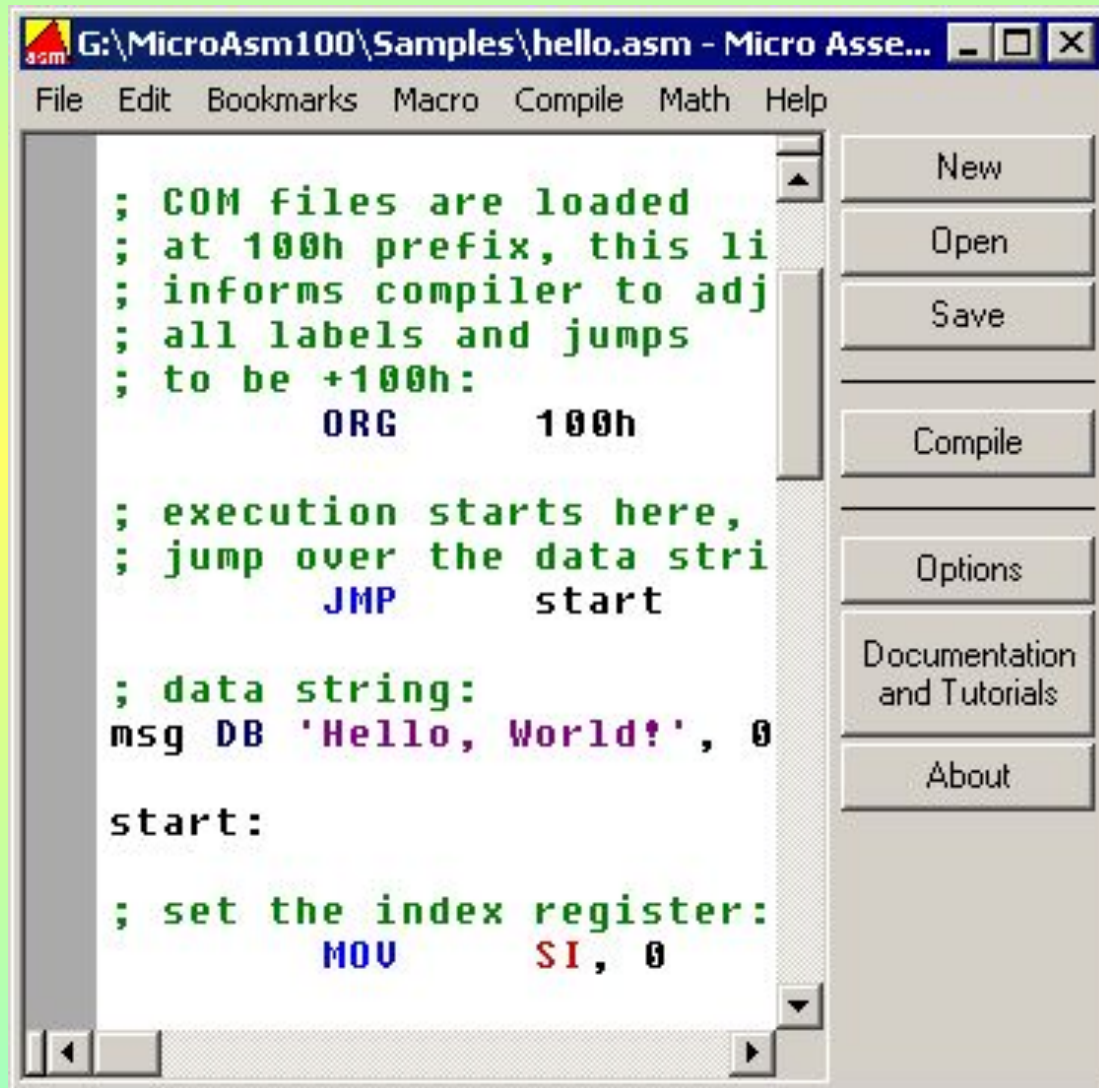
Программа

Программа - алгоритм, записанный на
«понятном» компьютеру языке
программирования

Машинный язык

```
01008010: 78 00 00 00-01 00 00 00-4E 00 6F 00-74 00 65 00 x 0 Note
01008020: 70 00 61 00-64 00 00 00-FF FF FF FF-01 00 00 00 p a d
01008030: 03 00 00 00-05 00 00 00-00 00 00 00-08 00 00 00 v 2 0
01008040: 10 00 00 00-11 00 00 00-0C 00 00 00-12 00 00 00 > + 0
01008050: 13 00 00 00-18 00 00 00-19 00 00 00-1A 00 00 00 || T I +
01008060: 1E 00 00 00-1F 00 00 00-20 00 00 00-22 00 00 00 ^ v "
01008070: 23 00 00 00-28 00 00 00-2C 00 00 00-2D 00 00 00 # + -
01008080: 2E 00 00 00-2F 00 00 00-30 00 00 00-32 00 00 00 . / 0 2
01008090: 34 0 mov     ecx,0C0000000 ;"
010080A0: 17 0 mov     edx,000401000 ;"
010080B0: 51 0 xor     eax,edx
010080C0: 2C 8 push   eax
010080D0: 3C 8 push   000000000 ;"
010080E0: 4C 8 push   003
010080F0: 58 8 push   eax
01008100: 68 8 push   eax
01008110: 78 8 push   ecx
01008120: 88 8 push   edx
01008130: 9C 8 call   001008547 ;----- (1)
01008140: AC 8 cmp    eax,-001 ;"0"
01008150: 98 1 je     001008478 ;----- (2)
01008160: 02 0 mov     [000401000],eax
01008170: 05 0 push   000
01008180: 04 1 push   d,[000401000]
01008190: 08 1 call   001008541 ;----- (3)
010081A0: 00 0
010081B0: B9 00 00 00-C0 BA 00 10-40 00 33 C0-50 68 80 00
010081C0: 00 00 6A 03-50 50 51 52-E8 7A 03 00-00 83 F8 FF
010081D0: 0F 84 A5 02-00 00 A3 0A-10 40 00 6A-00 FF 35 0A
010081E0: 10 40 00 E8-59 03 00 00-A3 0E 10 40-00 8B 00 0E
010081F0: 10 40 00 83-C1 52 E8 95-02 00 00 A3-7D 10 40 00
01008200: C7 05 12 10-40 00 00 00-01 00 81 3D-0E 10 40 00
01008210: 00 00 01 00-77 0A A1 0E-10 40 00 A3-12 10 40 00
```

Ассемблер



The screenshot shows the Micro Assembler software window. The title bar reads "G:\MicroAsm100\Samples\hello.asm - Micro Asse...". The menu bar includes "File", "Edit", "Bookmarks", "Macro", "Compile", "Math", and "Help". The main text area contains the following assembly code:

```
; COM files are loaded
; at 100h prefix, this li
; informs compiler to adj
; all labels and jumps
; to be +100h:
      ORG      100h

; execution starts here,
; jump over the data stri
      JMP      start

; data string:
msg DB 'Hello, World!', 0

start:

; set the index register:
      MOV      SI, 0
```

On the right side of the window, there is a vertical toolbar with buttons for "New", "Open", "Save", "Compile", "Options", "Documentation and Tutorials", and "About".

Языки программирования высокого уровня

QBasic



Pascal

```
Turbo Pascal
File Edit Search Run Compile Debug Tools Options Window Help
PAS\EXAMPL00.PAS 2-[F1]
< 11 областная олимпиада школьников. Киров. 1999 год.
Резус UOLVO + FIAT = MOTOR >
Program Rebus;
Uses Crt;
Const S:array[1..5] of longint = <10,100,1000,10000,100000>;
Var A1, A2, A3:longint;
    U,O,L,F,I,A,T:byte;
    Ss:set of 0..9;
Procedure Prov; < A1 + A2 = A3 ? >
var B:longint;
    Pr:byte;
begin
  b:=a3; pr:=b div s[4]; if pr in Ss then exit;
  b:=b mod s[4]; pr:=b div s[3]; if pr <> 0 then exit;
  b:=b mod s[3]; pr:=b div s[2]; if pr <> I then exit;
  b:=b mod s[2]; pr:=b div s[1]; if pr <> 0 then exit;
  pr:=b mod s[1]; if pr in Ss then exit;
  Writeln(A1,' + ',A2,' = ',A3);
end;
18:26
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local
```

Программа Pascal, названная в честь
Блеза Паскаля



Французский физик-математик
Блез Паскаль

Delphi

```
Unit1.pas  
Unit1  
unit Unit1;  
  
interface  
  
uses  
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls  
  Dialogs;  
  
type  
  TForm1 = class(TForm)
```

Object TreeView

Form1

Object Inspector

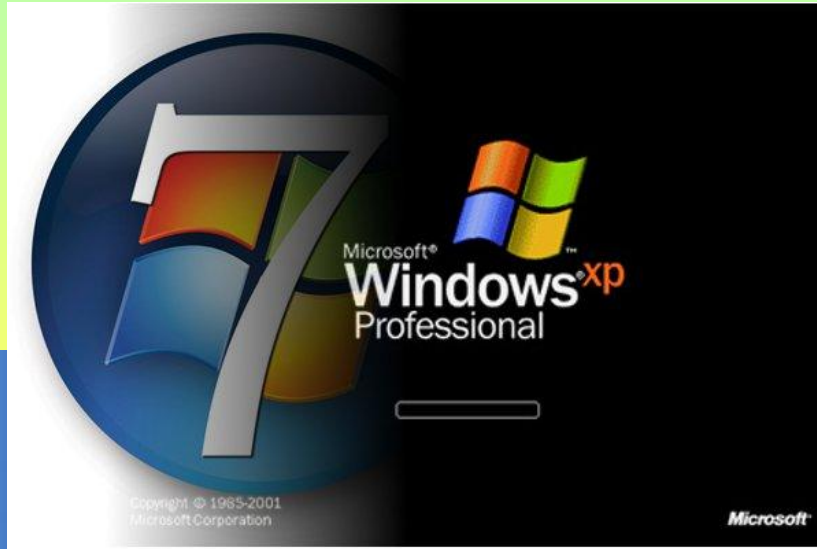
Form1 TForm1

Properties Events

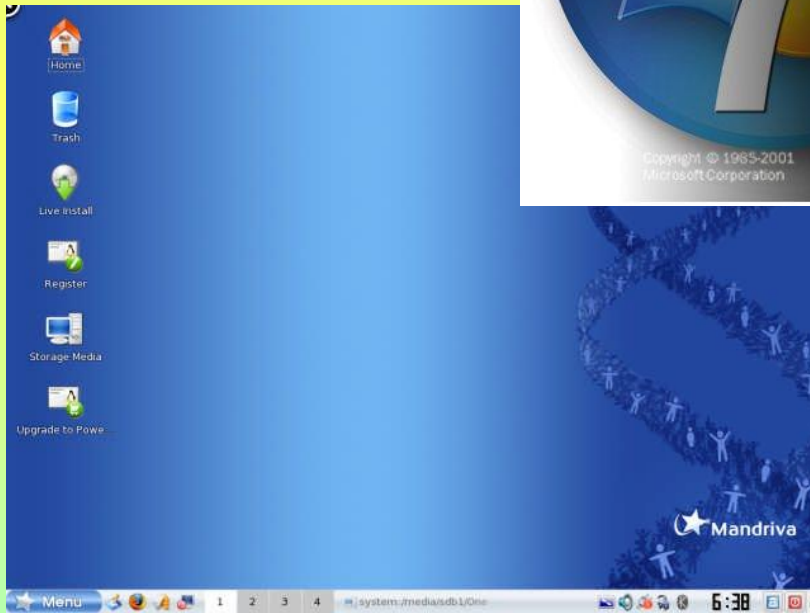
Action	
ActiveControl	
Align	alNone
AlphaBlend	False
AlphaBlendValue	255
Anchors	[akLeft,akTop]
AutoScroll	True
AutoSize	False
BiDiMode	bdLeftToRight
BorderIcons	[biSystemMenu]
BorderStyle	bsSizeable
BorderWidth	0
Caption	Form1

All shown

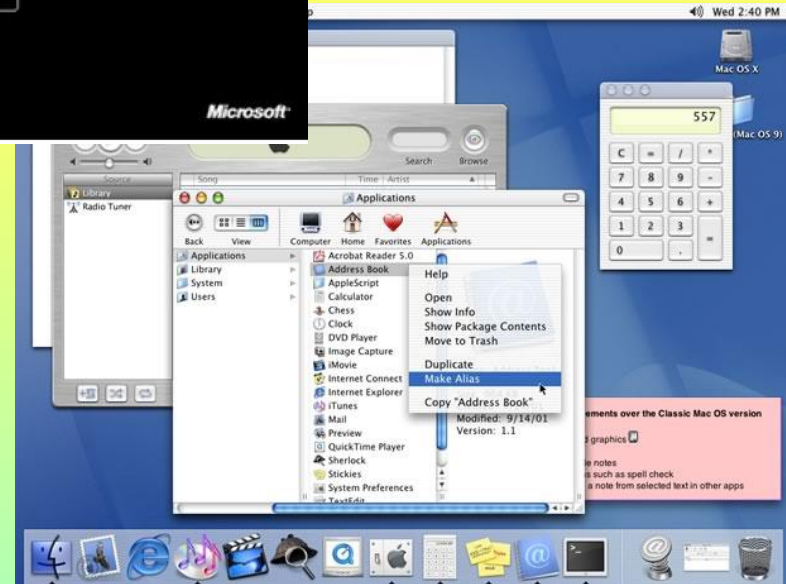
Операционные системы



XP



Linux



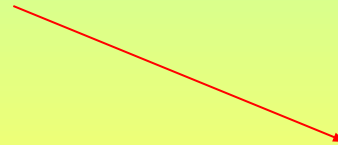
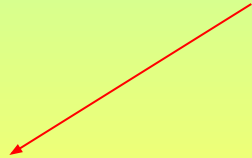
Mac OS

Программы-трансляторы

Программы-трансляторы

Интерпретаторы

Компиляторы



Домашнее задание

составить блок-схему решения
квадратного уравнения.