

ЯЗЫКОВ программирования

С каждым днём наш мир становится более мобильным и информационным. Всё больше и больше компьютеры вступают в нашу повседневную жизнь, и чтобы облегчить наше общение с ними создаётся новое ПО с помощью различных языков программирования.

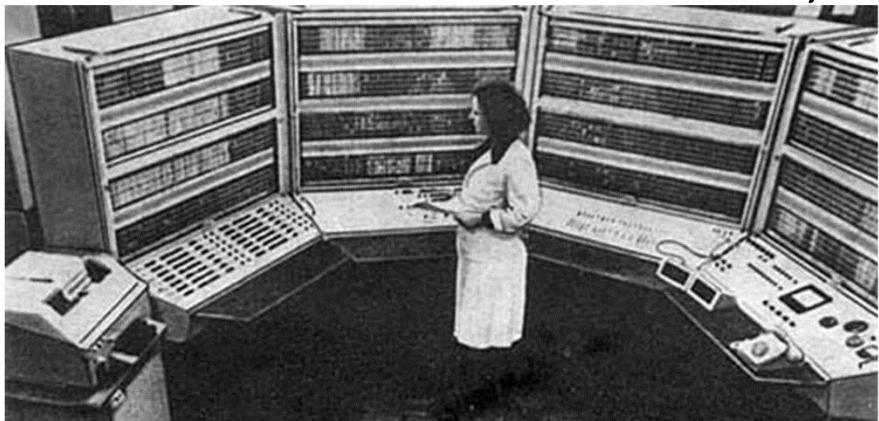
Как люди представляют программирование: Как оно выглядит на самом деле:



Что же такое программа и когда они

появились?

- Программа – набор инструкций для определенного исполнителя.
- Первый язык программирования состоял из единиц и нулей. Как показала практика он оказался громоздким и неудобным, там легко допустить ошибку, и нужно знать принцип работы каждого блока ЭВМ (Электронно-вычислительная машина)
- В 1950 году для записи программ начали применять мнемонический язык — Ассемблер (assembly). Язык ассемблера позволил представить машинный код в более удобной для человека форме: для обозначения команд и объектов, над которыми эти команды выполняются, вместо двоичных кодов использовались буквы или сокращенные слова, которые отражали суть команды. Например, на языке ассемблера команда сложения двух чисел обозначается словом add, тогда как ее машинный код м...



Один из первых ЭВМ



ЭВМ

```
00898159 8D78 04 LEA EDI, IERX+41
0089815C 3D85 77040000 LEA EAX, IEBP+4771
00898162 57 PUSH EDI
00898163 50 PUSH EAX
00898164 E8 46000000 CALL Decompress
00898169 57 PUSH EDI
0089816A E8 EB000000 CALL PrepareImage
0089816F 89C6 MOV ESI, EAX
00898171 31C0 XOR EAX, EAX
00898173 83EF 04 SUB EDI, 4
00898176 8B0F MOV ECX, DWORD PTR DS:[EDI]
00898178 57 PUSH EDI
00898179 F3AA REP STOS BYTE PTR ES:[EDI]
0089817B 5F POP EDI
0089817C 57 PUSH EDI
0089817D FF95 5B040000 CALL DWORD PTR SS:[EBP+45B]
00898183 8BBD 67040000 MOV EDI, DWORD PTR SS:[EBP+4671]
00898189 31C0 XOR EAX, EAX
0089818B 83EF 04 SUB EDI, 4
0089818E 8B0F MOV ECX, DWORD PTR DS:[EDI]
00898190 57 PUSH EDI
00898191 F3AA REP STOS BYTE PTR ES:[EDI]
00898193 5F POP EDI
00898194 57 PUSH EDI
00898195 FF95 5B040000 CALL DWORD PTR SS:[EBP+45B]
0089819B 85F6 TEST ESI, ESI
0089819D 74 07 JE SHORT 008981A6
0089819F 897424 1C MOV DWORD PTR SS:[IESP+1C], ESI
008981A3 61 POPAD
008981A4 FFE0 JMP EAX
008981A6 61 POPAD
008981A7 B8 FFFFFFFF MOV EAX, -1
008981AC C2 0800 RETN 8
```

Ассемблер

Развитие языков программ. Высокого уровня

С середины 50-х г. XX в. начали создавать первые языки программирования высокого уровня (high-level programming languages). Эти языки не были привязаны к определенному типу ЭВМ (машинонезависимы). Для каждого из них были разработаны собственные компиляторы. Компиляция — трансляция программы, составленной на исходном языке высокого уровня, в эквивалентную программу на низкоуровневом языке, близком машинному коду

FORTRAN

```
0056      do 0057 l=1,mmax      go to 0061
0057      dept1(l)=depth(l)
0058      dept1(0)=0
C-----attention - water: ignore for Love waves!!!!
0059      if (b(0).le.0.0) dept1(0)=d(0)
0060      amp(0)=1.
0061      stress(0)=0.
0062      dcd0(0)=0.
0063      dcd1(0)=0.
0064      g(0)=0.
C-----
C-----Calculation of eigenfunction and its derivatives
C-----
0065      if (ledit.ne.0)      go to 0066
0066      do 799 l=1,n
0067      do 799 j=1,ndep1
0068      continue
0069      do 801 l=1,ndpth
0070      if (dpth(l).lt.dept1(l))      go to 801
C-----If dpth(l) is above the free surface, the output will be zero
C-----
0071      do 802 ll=1,mmax
0072      rab1=dept1(ll)-dpth(l)
0073      if (abs(rab1).lt.1.0e-8) rab1=0.
0074      if (rab1.le.0.)      go to 802
0075      t=ll-1
0076      j=ll
0077      if (b(0).le.0.0) t=ll
C-----attention - water: ignore for Love waves!!!!
0078      rab2=d(l)/z.
0079      rab1=rab1-rab2
0080      if (abs(rab1).lt.1.0e-8) rab1=0.
0081      if (rab1.le.0.)      go to 801
0082      j=j-1
0083      t=t-1
0084      continue      go to 801
0085      j=mmax
0086      t=mmax-1
0087      if (b(0).le.0.0) t=j
C-----attention - water!!!!
0088      dz=dpth(l)-dept1(j)
0089      ampl=amp(j)
0090      stress=stress(j)
0091      covb=c(b(l))
0092      rb=svno*sqrt(abs(covb**2-1.))
0093      if (c-b(l)) 800,811,812
0094      nderv=nderiv+1
0095      do 800 ll=1,nderv
0096      k=ll-1
0097      rab1=rb**k
```

ALGOL 69

```
begin
integer i, n;
real s;
real array x[1:n];
s:=0;
for i:=1 step 1 to n
do
s:=s+x[i];
s:=s/n
end
```

LISP

```
55 (push cdfh :internalFileAttributes (read-u16 f))
56 (push cdfh :externalFileAttributes (read-u32 f))
57 (push cdfh :localFileHeaderOffset (read-u32 f))
58 (push cdfh :filename (read-b-string f (key cdfh :filenameLength)))
59 (stream-position f (+
60 (stream-position f)
61 (key cdfh :extraFieldLength)
62 (key cdfh :fileCommentLength)))
63 cdfh)
64
65
66 (procedure read-zip (arch :optional fn)
67 (var z (file arch))
68 (var pos)
69 (var filename)
70 (var sign (read-u32 z))
71 (while (<= sign $06054B50)
72 (stream-encoding z :utf8)
73 (case sign
74 ($04034B50
75 (var ldfh (read-LDFH z))
76 (set filename (key ldfh :filename))
77 (set pos (stream-position z))
78
79 (cond
80 ((NIL? fn)
```

COBOL

```
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO-WORLD.

*
ENVIRONMENT DIVISION.
*
DATA DIVISION.
*
PROCEDURE DIVISION.
  PARA-1.
    DISPLAY "Hello, world.".
  *
  EXIT PROGRAM.
END PROGRAM HELLO-WORLD.
```

Выводы развития языков

программирования

- Подводя итог данному периоду развития языков программирования, можно сделать вывод, что языки программирования высокого уровня (FORTRAN, ALGOL, LISP, COBOL и т. д.) не похожи на язык ассемблера. Языки высокого уровня разработаны специально для того, чтобы можно было иметь дело непосредственно с задачей, решаемой программой. В этом качестве они иногда называются процедурными языками, поскольку описывают процедуру, используемую для решения задачи. Языки высокого уровня машиннезависимы. Программы же на языке ассемблера непосредственно относятся к той машине, на которой они должны выполняться.
- Позднее появились языки программирования, которые делятся на классы.
- **Класс** — это структура данных, содержащая в себе не только переменные, но и функции, которые работают с этими переменными.
- Коротко, это достижение в области программирования было очень велико. Теперь программирование можно было разбить на классы и тестировать не всю программу, состоящую из 10 000 строк кода, а разбить программу на 100 классов, и тестировать каждый класс. Это существенно облегчило написание программного продукта.
- В итоге в конце 1970-х и начале 1980-х были разработаны принципы **объективно-ориентированного программирования**. ООП сочетает лучшие принципы структурного программирования с новыми концепциями **инкапсуляции** (размещение в одном компоненте данных и методов, которые с ними работают), **полиморфизма подтипов** (способность функции обрабатывать данные разных типов) и **наследования**.
- Первым объектно-ориентированным языком программирования является Симула-67, в котором впервые появились классы. Концепции ООП получили дальнейшее развитие в языке Smalltalk, в котором также были заложены основы систем с **оконным управлением**. Более поздними и современными примерами объектно-ориентированных языков являются **Object Pascal**, **C++**, **Java**, **C#** и др.



Спасибо за внимание

Какое?

Текст