

ЯЗЫКИ ПРОГРАММИРОВАНИЯ

ПОНЯТИЕ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ.

Прежде чем разбираться с понятием языков программирования, необходимо усвоить, как работает компьютер. Когда мы пользуемся компьютером, будь то вводим какой-либо текст, смотрим фотографии, слушаем музыку или устанавливаем программу, компьютер работает только с двоичным кодом. Любая ЭВМ воспринимает команды только из единиц и нулей, где 0 - отсутствие заряда, а 1 - его присутствие. То есть компьютер работает с последовательностью перепада напряжения. Чтобы компьютер выполнял какие-либо действия, люди создавали машинный код, состоящий только из единиц и нулей, и с ним работал компьютер. Нетрудно догадаться, что такой способ написания программ имел очень много недостатков - он был громоздк, запутан, непонятен, а допустить ошибку в огромных последовательностях нулей и единиц было очень просто. Тогда люди задались вопросом - как объяснить компьютеру на понятном языке то, что мы хотим от него?



Вспомним понятие языка. **Язык - сложная знаковая система, естественно или искусственно созданная и соотносящая понятийное содержание и типовое звучание (написание).** Здесь нас интересуют слова "естественно созданная" и "искусственно созданная". Человеческие языки являются **естественными**, они не подчинены строгим правилам, они бурно и стихийно развиваются и зависят от многих факторов. **Искусственными**, или **формальными** языками называются языки, которые подчинены строгим правилам написания и понимания. К таким языкам и относятся языки программирования. Если мы допустим ошибку в слове при разговоре или при письме, это не повлияет на то, что мы хотим донести. А вот в формальных языках любая неточность недопустима.

Естественные языки



Формальные языки



Пример

```
TITLE Сложение и вычитание (AddSub.asm)
; числа 32-разрядные
.386
.MODEL flat, sdtcall
.STACK 4096
ExitProcess PROTO, dwExitCode:DWORD
DumpRegs PROTO

.code
main PROC
    mov  eax,1000h
    add  eax,4000h
    sub  eax,2000h
    call DumpRegs
    INVOKE ExitProcess,0
main ENDP
END main
```

