

Понятие о языках программирования

Язык программирования – это формальный язык, предназначенный для записи компьютерных программ.

Язык программирования определяет набор лексических, синтаксических и семантических правил, определяющих внешний вид программы и действия, которые выполнит исполнитель (обычно — ЭВМ) под её управлением.

Спецификация языков программирования:

Стандартизация – приведение языков программирования к международным стандартам, производимое по результатам работы специальных организаций по обновлению, публикации спецификаций языков, их разработка и модернизация.

Алфавит – фиксированный для данного языка набор основных символов, допускаемых для составления текста программы на этом языке.

Лексика – фиксированный набор основных слов (команд), определяющих выполняемые компьютером действия.

Синтаксис – система правил, определяющих допустимые конструкции языка программирования из букв алфавита.

Семантика – система правил однозначного толкования отдельных языковых конструкций, позволяющих воспроизвести процесс обработки данных.

Классификация языков программирования

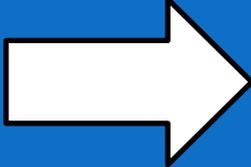


Языки программирования 1 поколения (1GL)

Машинный язык (платформенно-ориентированный код) — система команд (набор кодов операций) конкретной вычислительной машины, которая интерпретируется непосредственно процессором или микропрограммами машины. Представляет собой двоичный код. Иногда для упрощения программу записывают в шестнадцатеричном коде, который переводится в двоичный непосредственно микропроцессором. Машинные языки хороши для детального понимания функционирования конкретной машины, но сложны для изучения и решения прикладных задач.

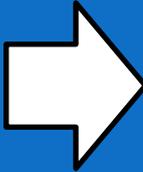
Позднее стал применяться **язык ассемблера** — язык, в котором двоичные и шестнадцатеричные коды стали заменяться буквенными обозначениями, которые называются **мнемоники**. Программа из языка ассемблера переводилась в машинный код при помощи программы-транслятора, которая называется ассемблер (данная программа дала название языку).

Примеры



1001 0001	91
1000 1000	88
1000 1100	8C
1000 0010	82
1000 1110	8E
1000 1011	8B

Данной программой закодировано слово «СИМВОЛ»



```
SECTION.text
org 0x100
mov ah, 0x9
mov dx, hello
int 0x21
mov ax, 0x4c00
int 0x21
SECTION.data
hello: db "Hello, world!",
0xD, 0xA.
```

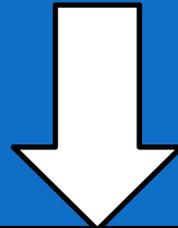
Языки программирования 2 поколения (2GL)

Появились в 1950-е годы для перехода в выражениях языка от низкоуровневых машинных понятий ближе к тому, как обычно мыслит программист. Основные отличия от языков 1 поколения:

- команды пишутся словами;
- для перевода программы в машинный код применяется программа – компилятор.

К языкам поколения 2 GL относятся:

- Fortran;
- Cobol и т.д.



Пример программы на языке «Fortran»:

```
PROGRAM PR_1
INTEGER:: J=2
REAL :: A=3.4, F=5.25, B=9.7
A=F ! значение переменной F присваивается переменной A
J=B ! значение переменной B присваивается переменной J
PRINT*, "A=",A," F=",F," J=",J ! вывод результатов на экран
END
```

В результате выполнения программы выводится: A=5.25 F=5.25 J=9

Языки программирования 3 поколения (3GL)

Языки 3 поколения унаследовали все достоинства языков 2 поколения и дополнили их своими достоинствами:

- 1) Простота и понятность использования.
- 2) Независимость от конкретного компьютера – это достигалось тем что теперь между пользователем и архитектурой ЭВМ была ОС.
- 3) Возможность использования специальных синтаксических приемов – программы стали более сложными в них появились блоки команд объединённые в процедуры и функции.
- 4) Модульность программ – написание отдельных процедур и функций для решения отдельных маленьких задач позволило их повторно использовать в других проектах.

Пример программы языке «Pascal».

Данная программа вводит с клавиатуры n чисел и определяет четные.

```
var n, i, k, a: integer;
begin
writeln ('введите количество чисел');
Readln (n);
a:=0;
for i:=1 to n do begin
    writeln ('введите ',i:1,'-е число');
    readln (a);
    if a mod 2=0 then k:=k+1;
    end;
writeln ('кол-во четных чисел ',k);
readln;
end.
```

Примеры языков программирования 3GL: Lisp, BASIC, Pascal, C и т.д.

Языки программирования 4 поколения (4GL)

Они относятся к временному периоду с 1980-х настоящее время.

Языки этого поколения предназначены для реализации крупных проектов, повышают их надежность и скорость создания, ориентированы на специализированные области применения.

В них встроены операторы, позволяющие одной строкой описать такую функциональность, для реализации которой на языках младших поколений потребовались бы тысячи строк исходного кода.

Данные языки наряду с языками 3GL оперируют метаданными (данные о данных, раскрывающие сведения о признаках и свойствах, характеризующих какие-либо сущности, позволяющие автоматически искать и управлять ими в больших информационных потоках).



Пример программы нахождения 10 наиболее частых слов на web-странице (язык «Python»):

```
from urllib2 import urlopen
u = urlopen("http://python.org")
words = {}
for line in u:
    line = line.strip(" \n")
    for word in line.split(" "):
        try:
            words[word] += 1
        except KeyError:
            words[word] = 1
pairs = words.items()
pairs.sort(key=lambda x: x[1],
reverse=True)
for p in pairs[:10]:
    print(p[0], p[1])
```

Примеры языков программирования 4GL:

C#, 1C, JavaScript, SQL, Prolog, Python.

Языки программирования 5 поколения (5GL)

Планируются в будущем. В настоящее время данных языков программирования не существует.

Предполагается:

1. Будут оперировать мета-мета-данными (сведения о сведениях о данных, т.е. содержат конкретные количественные и качественные показатели признаков и свойств о данных);
2. Будут иметь производительность в 10-10000 раз более высокую по сравнению с языками программирования 3GL и 4GL.
3. Возможность автоматического формирования результирующего текста на универсальных языках программирования, ввод инструкций в максимально наглядном виде с помощью методов, наиболее удобных для человека, не знакомого с программированием.

Сейчас существует единственный язык, который работает с мета-мета-данными, - это язык команд менеджеров пакетов или менеджеров зависимостей, таких как apt, yum, smart, maven, cran и другие. Однако данный язык является языком командной строки, а не языком программирования).