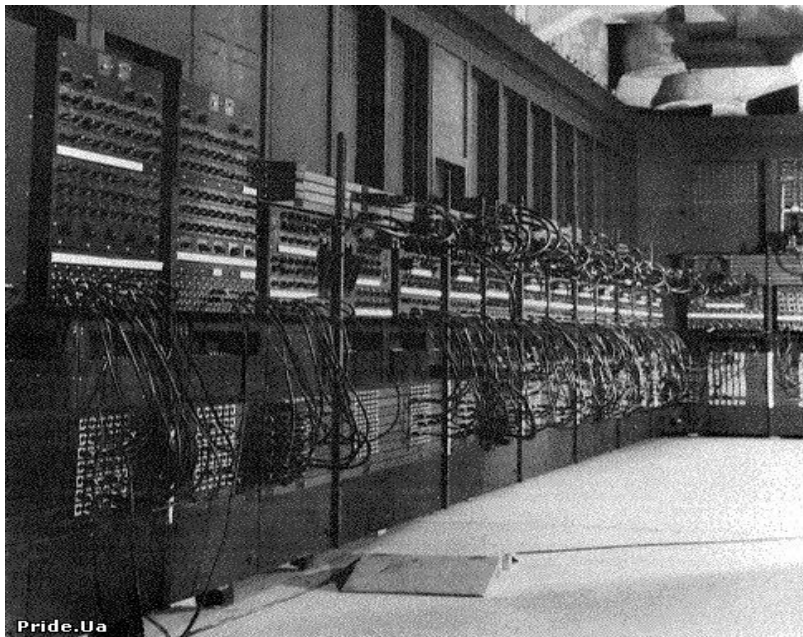




Языки программирования высокого уровня

Компьютерная программа
представляет собой логически
упорядоченную
последовательность команд,
предназначенных для управления
компьютером.

В компьютерах **первого** поколения



использовались программы, написанные в
машинных кодах.

Программа вывода «Привет, друг!»:

```
BB 11 01 B9 0D 00 B4 0E 8A 07 43 CD 10 E2 F9 CD  
20 48 65 6C 6C 6F 2C 20 57 6F 72 6C 64 21
```

Ассемблер

- первый язык программирования (1950 –е гг.).

Близок к машинному языку.

Появилось понятие переменной.

Команды обозначаются словами, вместо конкретных адресов можно использовать имена.

LD (nn),A - (машинный код: 50) «загрузить в ячейку памяти, адрес которой задан двухбайтным числом **nn**, содержимое регистра **A**».

ADD A,B (ее машинный код - 128) - «прибавить содержимое регистра **B** процессора к содержимому регистра **A** и результат оставить в регистре **A**».

Например, хотим сложить содержимое ячеек 30013 и 30014, а результат поместить, а ячейку 30015:

Ассемблер	Адрес	Машинный код	Комментарий
LD A,(30013)	30000	58	Загрузить в регистр A число, содержащееся в адресе 30013.
	30001	61	
	30002	117	
LD B,A	30003	71	Загрузить в регистр B содержимое регистра A
LD A,(30014)	30004	58	Загрузить в регистр A число, содержащееся в адресе 30014.
	30005	62	
	30006	117	
ADD A,B	30007	128	Прибавить к содержимому A содержимое регистра B .
LD (30015),A	30008	50	Выгрузить содержимое регистра A в адрес 30015.
	30009	63	
	30010	117	
NOP	30011	0	Нет операции. Пауза.
RET	30012	201	Возврат туда, откуда эта программа вызывалась.

Хотим сложить 2 числа: 40 и 60.

```
10 CLEAR 29999 'выделение памяти для этой
                    программы в машинных кодах
20 FOR i=30000 TO 30012: READ q: POKE i,q: NEXT i
    'ввод программы в память, начиная с адреса
    30000
30 DATA 58, 61, 117, 71, 58, 62, 117, 128, 50, 63, 117,
    0, 201
40 POKE 30013, 40: POKE 30014, 60 'запись в ячейки
    30013 и 30014 двух чисел, которые хотим сложить
50 RANDOMIZE USR 30000 'ввод команды на
                    исполнение нашей программы
```

Языки программирования, ориентированные на команды процессора и учитывающие его особенности, называются **языками низкого уровня.**

Языком программирования **высокого уровня** называется фиксированная система обозначений и правил для описания алгоритмов и структур данных.

Их **отличительные особенности:**

- аппаратная независимость;
- близость к естественному языку.

Преимущества языков программирования высокого уровня перед машинно- ориентированными:

- 👍 лучшая читаемость программы;
- 👍 меньшее время разработки программы.

Языки программирования

Процедурные

Объектно-
ориентированные

Декларативные

Функциональные

Логические

Процедурные языки требуют описания алгоритма по шагам.

Фортран

(**FOR**mula **TRAN**slator – переводчик формул).

Создан в период с 1954 по 1957 г. группой программистов фирмы IBM.

Основное назначение – программирование научно-технических задач.

```
real,dimension(:,:) :: V
```

...

```
allocate(V(-2:2,0:10)) ! Выделить память под массив, индексы которого  
                        могут меняться в пределах от -2 до 2 (первый индекс)  
                        и от 0 до 10 – второй
```

...

```
V(2,2:3)=V(-1:0,1)    ! Повернуть кусочек массива
```

```
write(*,*)V(1,:)     ! Напечатать все элементы массива V,  
                      первый индекс которых равен 1.
```

```
deallocate(V)
```

Программа «Привет, друг!» :

```
program privet
```

```
print *, "Privet, drug!«
```

```
end
```

Кобол

(**CO**mmon **B**usiness **O**riented **L**anguage –
общепринятый деловой язык).

Разработан в 1959 г. Руководитель – Грейс Хоппер.

Был ориентирован на решение учетно-экономических и управленческих задач.

IDENTIFICATION DIVISION.
PROGRAM-ID. PRIVET-DRUG.

*

ENVIRONMENT DIVISION.

*

DATA DIVISION.

*

PROCEDURE DIVISION.

PARA-1.

DISPLAY "Privet, drug.".

*

EXIT PROGRAM.

END PROGRAM PRIVET-DRUG.

Алгол

(**ALGO**rithmic **L**anguage – алгоритмический язык). Позволяет легко переводить алгебраические формулы в программные команды.

Разработан в 1958 -1960 гг. группой зарубежных специалистов.

Применение – составление программ для решения научно-технических задач на ЭВМ.

В нем впервые были введены понятия «блочная структура программы», «динамическое распределение памяти».

procedure Absmax(a) Size:(n, m) Result:(y) Subscripts:(i, k);

value n, m; **array** a; **integer** n, m, i, k; **real** y;

comment Максимальный элемент матрицы a, размера n на m передаётся в виде результата в y, а его индексы — в параметры i и k;

begin integer p, q;

y := 0; i := k := 1;

for p:=1 **step** 1 **until** n **do**

for q:=1 **step** 1 **until** m **do**

if abs(a[p, q]) > y **then**

begin y := abs(a[p, q]);

i := p; k := q

end

end Absmax

program HiFriends;

begin

print "Privet, drug";

end;

В 1963 г. сотрудники Дартмутского колледжа **Томас Курц** и **Джон Кемени**



создали новый язык, который называли
универсальным символическим кодом для
начинающих.

Beginners All-purpose Symbolic Instruction Code.

BASIC

Получил самое широкое распространение при работе на персональных компьютерах.

Популярность Бейсика объясняется как простотой его освоения, так и наличием достаточно мощных универсальных средств, пригодных для решения научных, технических, экономических задач, а также задач бытового характера, игровых и т.д.

10 CLS 'Очистка экрана

20 PRINT "Добро пожаловать!" 'Заголовок в первой строке

30 'Цикл, выводящий линию под заголовком, на всю ширину
экрана

40 FOR I=1 TO 80

10 PRINT "Privet, drug!"

50 PRINT "=";

60 NEXT I

65 'Ввод символьных данных от пользователя (комментарий
добавлен после ввода нижних строк)

70 INPUT "Имя: ",N\$

80 INPUT "Фамилия: ",S\$

90 INPUT "Отчество: ",T\$

95 'Вырезаем копию первых символов из имени и отчества

100 N2\$=LEFT\$(N\$,1)

110 T2\$=LEFT\$(T\$,1)

120 'Выводим результат

130 PRINT "Ваше имя кратко: ";S\$;" ";N2\$;" ";T2\$;" "

140 INPUT "Повторить программу? (Y/N) ",U\$

150 IF U\$="Y" THEN GOTO 10

160 END

Паскаль

Pascal

Был создан в 1968-1969 гг. Никлаусом Виртом



в Цюрихе (Швейцария).

Был назван в честь выдающегося французского математика, физика, литератора и философа **Блеза Паскаля**



(1623 – 1662 г.),
который создал
первую в мире
механическую
машину,
складывающую
числа.

Создавался, чтобы способствовать хорошему стилю программирования.

Широко применялся в промышленном программировании, до сих пор используется для обучения программированию студентов, является базой для других языков.

Особенность языка – строгая типизация.

```
C:\WINDOWS\system32\cmd.exe - turbo
File Edit Run Compile Options Debug Break/watch
Line 4 Col 6 Insert Indent Unindent C:NONAME.PAS
uses Crt;
var
  name: string[100];
begin
  clrscr;
  write ('NAME: ');
  readln(name);
  writeln ('Hello ', name);
  while not keypressed do;
end.
Watch
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F10-Menu NUM
```

```
Program HiFr;
begin
  write('Privet, drug!')
end.
```

Ада

- язык программирования, созданный в 1979 г.
в результате проекта, предпринятого
Министерством обороны США. Назван в честь
Ады Лавлейс.



Язык Ада предназначен для создания и длительного сопровождения больших программных систем.

Четко выражена модульность.

Важное достоинство – возможность параллельного программирования ветвей программы.

```
with Ada.Text_IO;  
  procedure Privet is  
  use Ada.Text_IO;  
Begin  
  Put_Line("Privet, drug!");  
end Privet;
```


Язык С

стандартизированный процедурный язык программирования, разработанный в начале 1970-х годов сотрудниками Bell Labs **Кеном Томпсоном** и **Деннисом Ритчи**.



Си был создан для использования в операционной системе **Unix**. Является самым популярным языком для создания

системного программного обеспечения.

```
#include <stdio.h>
int main(void)
{
    printf("Privet, drug!\n");
    return 0;
}
```

Декларативные языки

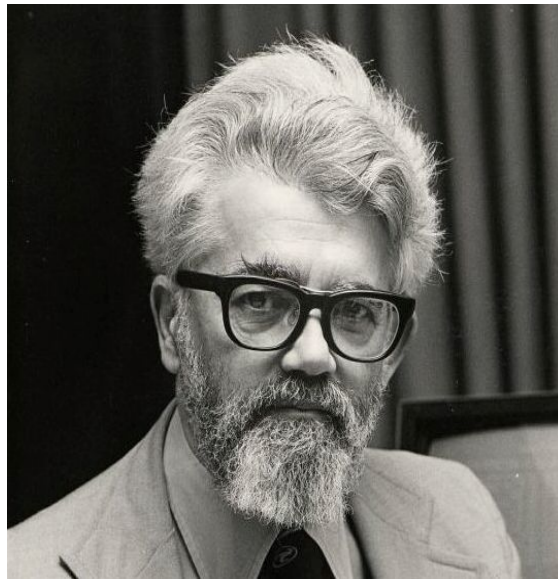
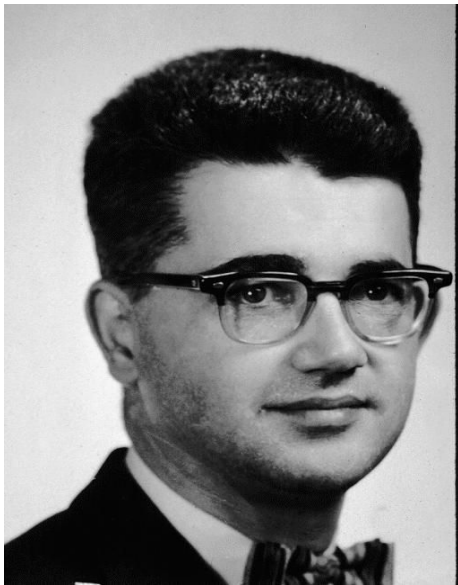
Не описывают процесс решения, а содержат набор правил – деклараций (законов, аксиом), применение которых (не обязательно последовательное) приводит к решению задачи.

В логических языках эти законы формулируются в виде формул логики предикатов.

В функциональных языках программа записывается в виде композиции функций.

Lisp

- функциональный язык. «Язык обработки списков». Создан в 1959 г. Джоном Маккарти.



Программы и данные представляются системами линейных списков символов.

(format t "Privet, drug!~%")

Логические языки

Логический **вывод** может быть **прямым и обратным**.

Примером логического языка является **Prolog** (**PRO**gramming in **LOG**ic – программирование в терминах логики).

Был создан в 1973 г. Аланом Кольмероэ.



В Прологе реализуется **обратный логический вывод**.

Языки программирования для компьютерных сетей

Это скрипт-языки (скрипты) – языки, которые можно использовать для программирования реакции на действия пользователя интернет-страниц.

VBS (Visual Basic Script),

JAVA-скрипт,

Perl.

Perl

Practical Extraction and Report Language —
«практический язык для извлечения
данных и составления отчётов»

Был создан в 1980-х годах Ларри Уоллом



Основная особенность —
многочисленные функции
работы со строками.

```
#!/usr/bin/perl  
print "Privet, drug!\n";
```



HTML

Hyper Text Markup Language

– язык разметки гипертекста.

Используется
для подготовки
Web-документов
для сети
Интернет.

Был разработан
британским ученым
Тимом Бернесом-Ли
в 1989-1991 годах в
стенах Европейского
совета по ядерным
исследованиям в
Женева.

 Этот текст - жирный,
<i> а этот - ещё и курсивный</i>

Этот текст - жирный, а этот — ещё и курсивный

Структуры и типы данных языка программирования

Тип данных характеризуется множеством допустимых значений и набором допустимых над этими значениями операций.

Тип переменной определяет:

- диапазон возможных значений;
- объем выделяемой памяти;
- допустимые операции.

Integer – целые числа.

Dim A As Integer ‘описание типа переменной A

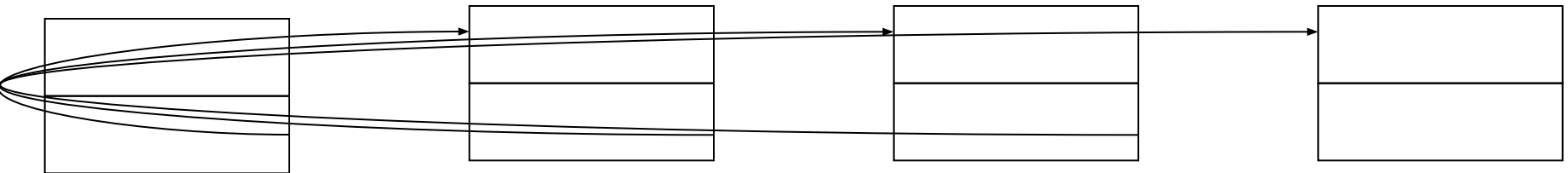
Integer A ‘описание типа переменной A

Dim A(10) As Integer ‘описание массива A из 10
целых чисел

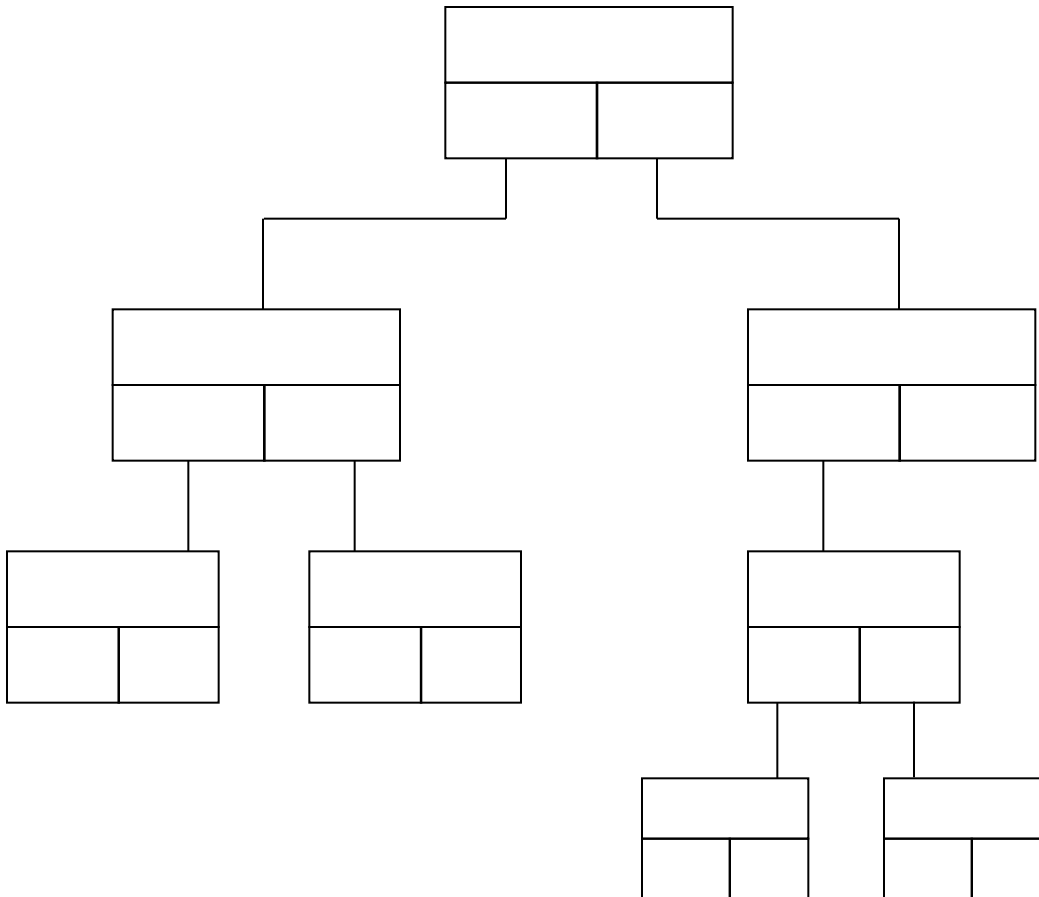
Int A[10] ‘описание массива A из 10
целых чисел

Список – это структура данных, состоящая из нескольких связанных друг с другом элементов.

Значение каждого элемента содержит собственное значение (некоторую информацию) и ссылку (адрес) на следующий элемент.



Бинарные деревья – от каждого элемента идет 2 адреса – указатели на левую ветку и на правую ветку.



Запись – это последовательность элементов **разного** типа.

ИМЯ	возраст	ПОЛ

Трансляция, компиляция и интерпретация

Транслятор – это программа, принимающая на вход программу на одном языке (**исходный код**) и преобразующая ее в программу, написанную на другом языке (**объектный код**).

Транслятор, который осуществляет перевод всей исходной программы в эквивалентную ей результирующую программу на языке машинных команд, называется **компилятором**.

Пошаговый перевод программы, написанной на языке программирования, в машинные команды, которые **сразу исполняются**, называется **интерпретацией**.

Программа для непосредственного исполнения программ называется **интерпретатором**.

Стадии обработки программы:

1) Лексический анализ.

Лексический анализатор занимается преобразованием исходного текста программы в последовательность специальных символов, кодирующих слова языка.

24#3 - эта ошибка обнаруживается на этапе лексического анализа

Посимвольный анализ слов **for** и **to** – на этом этапе.

2) Синтаксический анализ.

Синтаксический анализатор занимается разбором структуры программы в соответствии с правилами **грамматики** языка.

Результатом синтаксического анализа программы является **дерево разбора** со ссылками на таблицы объектов.

if for i=1 To 2 - эта ошибка распознается на этапе синтаксического анализа.

3) Семантический анализ.

Включает в себя проверку соответствия типов, наличия описаний.

Идентификатор abc не описан

- это сообщение выдается семантическим анализатором.

4) Генерация кода.

Каждой конструкции ставится в соответствие набор машинных команд.

После генерации кода работает
редактор связей.

Результатом работы редактора
связей является **загрузочный
модуль.**