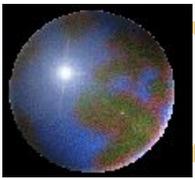


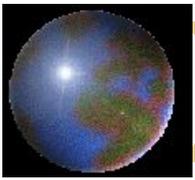
# *Программирование в AutoCAD*

## Лекция 1



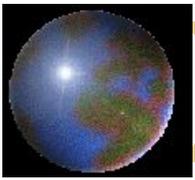
# *Предметные области*

- Электрические схемы
- Машиностроительные детали и узлы
- Схемы трубопроводов и сантехнических коммуникаций
- Архитектурные чертежи
- Карты и планы
- Лекала выкроек



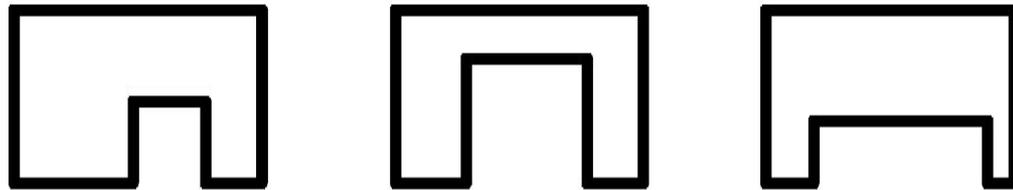
# *Преимущества библиотек блоков*

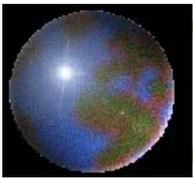
- Возможность масштабирования и поворота при вставке
- Возможность редактирования блоков
- Возможность дополнения блоков текстовыми пояснениями
- Экономия времени
- Уменьшение размеров файлов чертежей



## *Недостатки библиотек блоков*

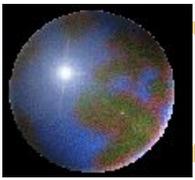
- Громоздкость при больших элементных базах (крепежные изделия)
- Невозможность параметрического изменения геометрии блоков





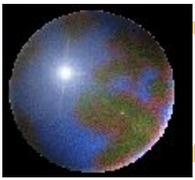
## *AutoLisp позволяет:*

- Использовать переменные и выражения при ответах на запросы AutoCAD
- Создавать функции и новые команды AutoCAD
- Осуществлять программный доступ к информации о чертеже
- Программно управлять графическим экраном AutoCAD



## *Особенности функциональных алгоритмических языков*

- Все вычисления, преобразования и управление в программе осуществляются с помощью функций (встроенных или пользовательских)
- Программа является суперпозицией функций и может быть использована как функция в других программах и в себе самой (рекурсия)

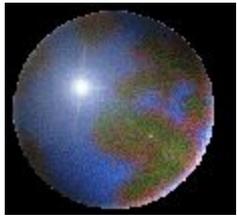


## *Примеры выражений LISP*

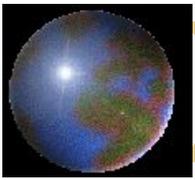
`(+ 4 7.8)`

`(setq a b)`

`(command "rectang" pause "@15,25")`

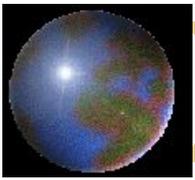


*Морфология  
алгоритмического языка  
LISP*



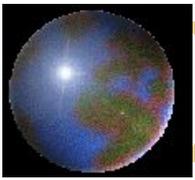
## *Переменные*

Объекты, имеющие  
уникальное имя и  
способные в себе хранить  
некоторые значения



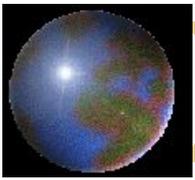
## *Имя переменной*

- Набор любых алфавитно-цифровых символов кроме:  
. , « ; ( ) пробел - = + \* / < > ? ` ! \ ^
- Нельзя использовать в качестве имени переменной зарезервированное имя
- Большие и малые буквы не различаются
- Не следует начинать имя с цифры



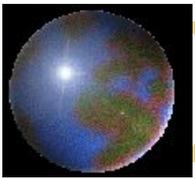
## *Тип переменной*

- В программе на LISP нет блока описания переменных
- Операции, производимые с переменной, определяются её типом
- Тип переменной зависит от данного, которое в ней хранится
- Переменной отводится место в момент присвоения первого значения
- (type имя) – функция для определения типа



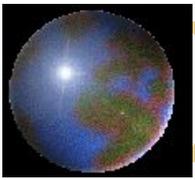
## *Строковый - STR*

- Ряд алфавитно-цифровых символов, заключенный в двойные кавычки  
"From point" "Radius <12.7>" "105"
- Строковое данное может содержать управляющие символы, например:
  - \n – перевод строки
  - \t – табуляция
  - \\ - \



## *Тип переменной*

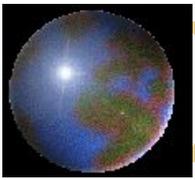
- **Целый (INT)** – положительные и отрицательные целые числа
- **Действительный (REAL)** – положительные и отрицательные числа с точкой
- **Дескрипторы файлов (FILE)** – имена открытых файлов, например:  
"acad.lsp"                      "drawing.dwg"



# *Тип переменной*

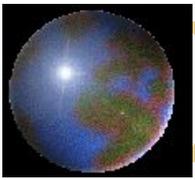
- **Встроенные функции (SUBR)** – внутренние адреса встроенных функций
- **Символы (SYM)** – адреса, указатели, ссылки. Символами часто называют переменные





## *Тип переменной*

- **Имя примитива (ENAME)** – восьмизначное шестнадцатеричное число, используемое для получения информации о примитиве
- **Наборы выбора AutoCAD (PICKSET)** – группа из одного или нескольких примитивов



# *Константы*

- Числовые

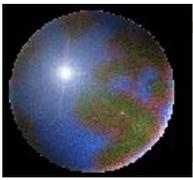
1.2    356    28.34E-14

- Строковые

"From point"

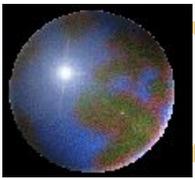
- Логические

NIL    T



# Атомы

- Константы и переменные простых типов (перечисленные выше) являются неделимыми элементами языка LISP и называются атомами
- (atom имя)  
(setq a 2) (atom nil)  
(atom a) T  
T



## *Тип переменной*

**Список (LIST)** – набор разделенных пробелами атомов или/и списков, заключённый в круглые скобки

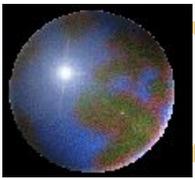
**()** или **nil** – пустой список

(12.6 45.7 77.8)

("cat" "mouse")

(1 (12 6 18) "cat" ("mouse"))

(\* 2 5)    nil    (nil)    (( ))    ((0))    ()



## Точка

(X Y Z)      (12 34)      (10 12.5 36)

## Отрезок

(P1 P2)      ((X1 Y1 Z1) (X2 Y2 Z2))

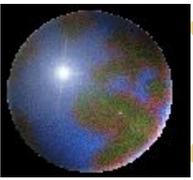
Сколько элементов в списке?

((1 2 3))

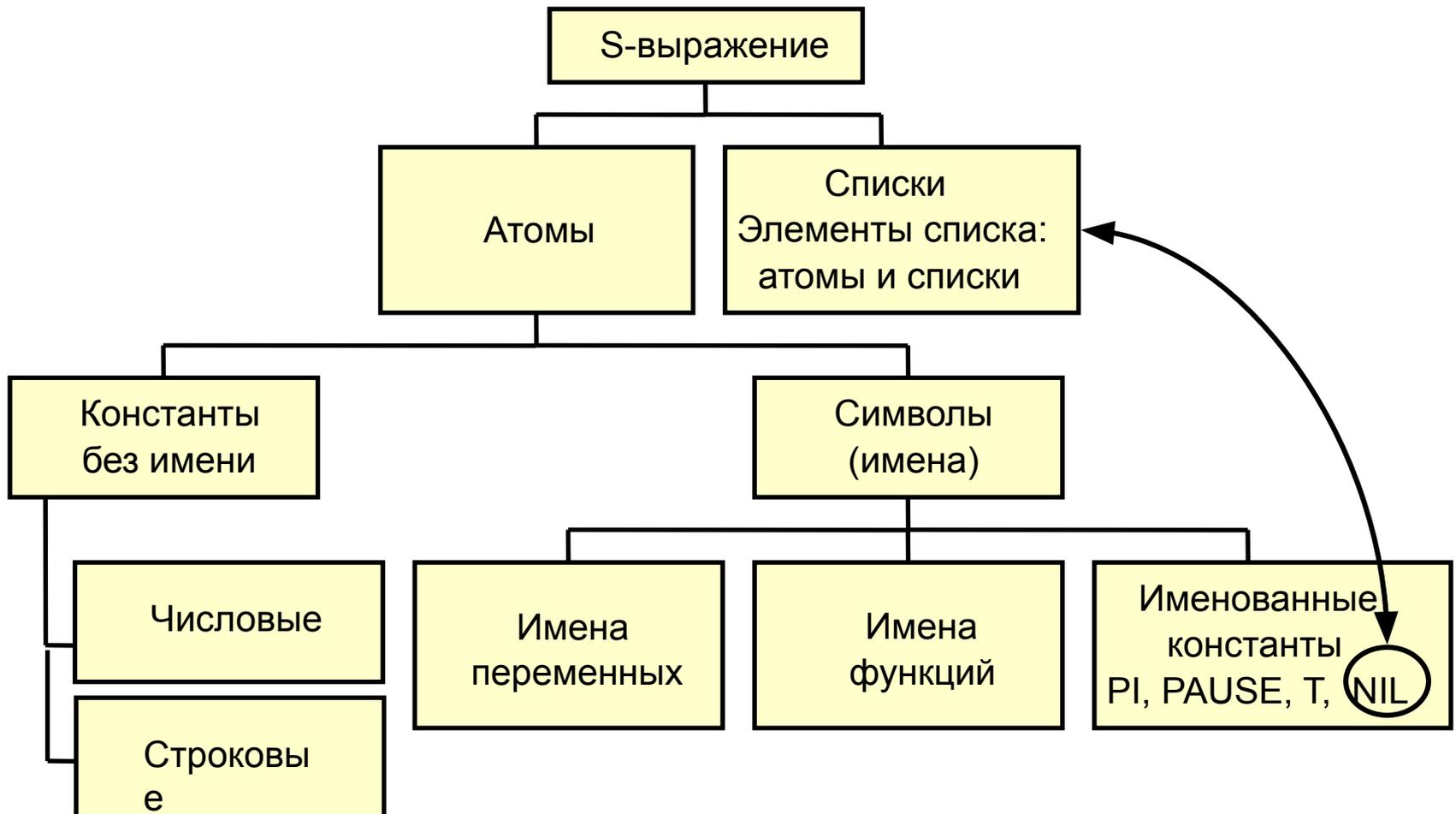
((a b) c (d (e)))

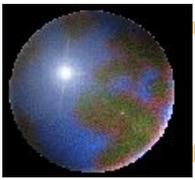
(a ((( ))) nil nil)

(((((a (b (c d) e) f) h ((i (j) k) l) m) n)))



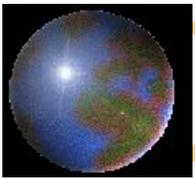
# Структура языка LISP





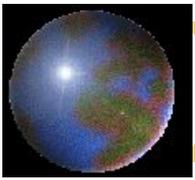
## *Выражения LISP (S – выражения)*

- Любая конструкция является **СИМВОЛЬНЫМ выражением (S – выражением)** и записывается в виде списка
- Первый элемент списка – имя функции. Исключение – список-данное, например, координата точки: (x y z)
- Следующие элементы списка – вычисляемые аргументы функции



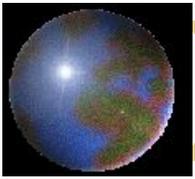
## *Результат вычисления S-выражения*

- Результат вычисления константы – она сама
- Результат вычисления переменной – её текущее значение
- Результат вычисления списка определяется функцией
- Тип S-выражения определяется типом результата его вычисления



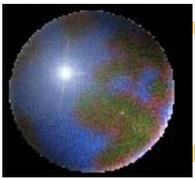
# *Для правильного использования функции надо знать:*

- Назначение функции
- Имя функции
- Количество аргументов
- Тип каждого аргумента
- Тип получаемого результата



## *Вложенность*

$( * 4 ( - 3 ( * 2 5 ) ) )$   
 $( + ( - 4 1 ) ( * 5 2 ) )$



## *Ввод S-выражения в AutoCAD:*

Command: (+ 5 3) <Enter>

8

Command: (setq a 9 b 9.0) <Enter>

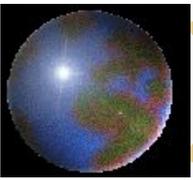
9.0

Command: (type a) <Enter>

INT

Command: (type b) <Enter>

REAL



## *Универсальная функция QUOTE*

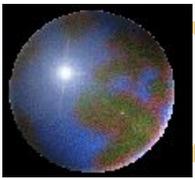
- Не вычисляя, возвращает аргумент в качестве результата
- Используется, когда аргументом является список-данное

`(quote (10 15 31))` → `(10 15 31)`

`(quote 15)` → `15`

`(quote b)` → `b`

`(quote (+ 2 3))` → `(+ 2 3)`



## *Универсальная функция QUOTE*

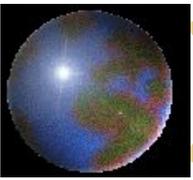
- ` – сокращенная запись функции

`(+ 2 3) → (+ 2 3)

`(a b `(c d)) → (a b `(c d))

(quote `b) → (quote b)

(quote quote) → quote



## *Универсальная функция EVAL*

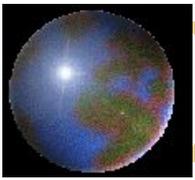
- Обращение к вычислителю

`(eval (quote (quote quote))) → quote`

`(quote (eval (quote (quote quote))))`



`(eval (quote (quote quote)))`

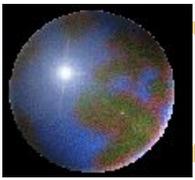


## *Псевдофункция SETQ*

- Служит для связывания переменной с её значением и типом

*(setq имя\_пер1 S-выр1  
[имя\_пер2 S-выр2...  
имя\_перN S-вырN])* → *S-вырN*

- Переменная остаётся связанной до следующей связи или до конца сеанса работы в AutoCAD



## *Псевдофункция SETQ*

(setq a 5.0) → 5.0

(setq b 123

c 4.7) → 4.7

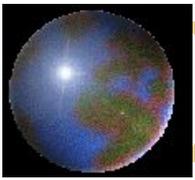
!a → 5.0   !b → 123   !c → 4.7   !d → nil

(setq a 5.0 b 12 c 2) → 2

(/ (+ a b) c) → 8.5

(setq a 5) → 5

(/ (+ a b) c) → 8

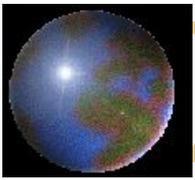


## *Псевдофункция SET*

- Служит для связывания переменной с её значением и типом

*(set S-выр\_имя1 S-выр1  
[S-выр\_имя2 S-выр2...  
S-выр\_имяN S-вырN])* → *S-вырN*

- Переменная остаётся связанной до следующей связи или до конца сеанса работы в AutoCAD



# *Переопределение базовых функций*

`(setq angle S-выр)`

`(setq t S-выр)`

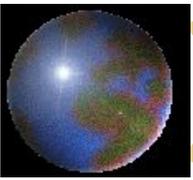
**Функция вывода списка  
зарезервированных имён**

`(atoms-family 0)` – вывод полного списка

`(atoms-family 0 (список имен))` – проверка

`(atoms-family 0 ("setq" "+" "xy"))` →

`(setq + nil)`



## *Косвенная связь переменных*

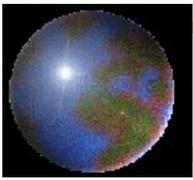
(setq b 123	INT
c 4.7	REAL
d (quote a))	SYM

Появляется переменная с именем **a**, но не связанная со значением

!d → a                      !a → nil

(set d (+ b c)) → 127.7

!d → a                      !a → 127.7



## *AutoLISP - калькулятор*

Command: `_circle` Specify center point for circle or [3P/2P/Ttr (tan tan radius)]:

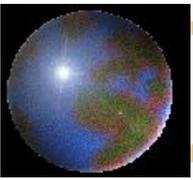
Specify radius of circle or [Diameter]:

`(/ 84.56 2.74)`

30.8613

`(setq rad (/ 25.6 3.4))`

Specify radius of circle or [Diameter]: `!rad`

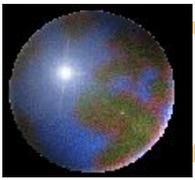


## *Функция связи с AutoCAD*

Функция выполняет команды AutoCAD  
(command [аргум1]... [аргумN]) → nil

### Аргументы

- Имена команд и опций – STR
- Координаты точек – LIST
- Длины, расстояния – REAL
- "" - <Enter>
- PAUSE



## *Функция связи с AutoCAD*

```
(setq y "0,0")
```

```
(command "_line" "10,10" y '(10 15) "")
```

```
(command "_break" pause "_f" pause "@")
```

```
(command "rectang" pause "@15,25")
```

```
(setq x 10.0 y 3)
```

```
(command "_circle" pause (* x y))
```