

Лекция 2

ПРОГРАММИРОВАНИЕ

I.1 Переменные. Константы

Переменные могут быть буквами греческого алфавита

α – alpha А - Alpha

γ – gamma Г - Gamma

λ – lambda Л - Lambda

θ – theta

ω – omega

И т.д.

маленькие буквы пишутся с маленькой. Больше с большой

Математические константы:

i – мнимая единица

π – число π

e – число e (экспонента)

II.1 Скобки и специальные СИМВОЛЫ

В Maple используются 3 типа скобок для различных операций и выражений

- Круглые $()$ – задают порядок действий при построении математических выражений и для указания параметров функций
- Квадратные $[]$ – служат для работы с индексами. Например списками или матрицами
- Фигурные $\{ \}$ – этот тип скобок служит для создания множеств

II.2 Диапазоны

Диапазоны предназначены для указания набора последовательных значений. Для этого используется оператор ..

Например для указания диапазона значений ряда.

```
>sum(x^2,x=1..5);
```

Вычисляется сумма квадратов от 1 до 10.
Эквивалентно $1^2+2^2+3^2+4^2+5^2$

```
> sum(x^2,x=1..5);
```

II.3 Результат предыдущей операции

% - результат выполнения предыдущей операции

%% - результат пред-предыдущей операции

```
> x:=2;
x := 2
> x^2;
4
> %^2;
16
> %-6;
10
> %%^2;
256
> solve(2*y=256, y);
128
```

III. Выражения

Выражения в Maple это обычные математические выражения, состоящие из констант, имен переменных, чисел, скобок и знаков операций

```
> x+(x*y)^2+y^x;  
x+x^2*y^2+y^x  
> 2*x^2+4*x+2;  
2x^2+4x+2  
> x+y < 1;  
x+y < 1  
> cos(x)^2+sin(x)^2=1;  
>  
cos(x)^2+sin(x)^2=1
```

III.1 Уравнения

Особый тип выражений это уравнения.
Для задания уравнений используется

СИМВОЛ

=

Вместо

:=

```
> eq:=x=12; # в переменной eq находится уравнение(!) x=12
                                eq := 12 = 12
> x:=12; # а в переменной(!) x находится число (выражение) 12
                                x := 12
> eq;
                                12 = 12
```


IV. Функции работы с выражениями

Алгебраические преобразования.

`simplify(<выражение>);` # упростить

`expand (<выражение>);` # раскрыть скобки

`factor (<выражение>);` # разложить на множители

`normal (<выражение>);` # привести к общему знаменателю

`combine (<выражение>);` # преобразовать степени и тригонометрические выражения

`collect (<выражение>);` # привести подробные

После ключевого слова в скобках вводится аналитическое выражение или

его имя — идентификатор, а также параметры, часть которых или все могут отсутствовать — быть необязательными.

IV.1 Примеры

```
[> restart;
[>
[> simplify((a^3-b^3)/(a-b));
[> expand((a-b)*(a^2+a*b+b^2));
[> factor(a^3-b^3);
[> normal(y/x+1/(x^2));
[> collect(x^2+3*x^2+4*x+y,x);
[> simplify(2*a/sqrt(a^2), assume(a<0));
[>
```

$$a^2 + b a + b^2$$
$$a^3 - b^3$$
$$(a - b)(a^2 + b a + b^2)$$
$$\frac{y x + 1}{x^2}$$
$$4 x^2 + 4 x + y$$
$$-2$$

VI. Последовательности. Списки. Множества.

VI.1 Последовательности. В Maple это набор элементов (чисел, переменных, строк, выражений, и т.д.) разделенных запятыми без скобок.

```
> s:=1,2,"a",a,x+1,a^2;
```

```
s:=1,2,"a",a,x+1,a2
```

Последовательности можно объединять указывая их через запятую

```
> a:=1,2,3,4;
```

```
a:=1,2,3,4
```

```
> b:="a","b","c";
```

```
b:="a","b","c"
```

```
> c:=x,x^2,x^3;
```

```
c:=x,x2,x3
```

```
> d:=a,b,c;
```

```
d:=1,2,3,4,"a","b","c",x,x2,x3
```

VI.2 Генерация последовательностей

Генерация последовательностей осуществляется с помощью функции **seq**

Форматы:

- ◎ **seq**(<выражение для элемента последовательности>, <переменная последовательности>=<диапазон изменения>);
- ◎ **seq**(<выражение для элемента последовательности>, <выражение для переменной>)

VI.3 Примеры генерации последовательностей

```
> L:=seq(i,i=0..10);
```

$L = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$

```
> seq(sin(Pi*i/6),i=0..6);
```

$0, \frac{1}{2}, \frac{\sqrt{3}}{2}, 1, \frac{\sqrt{3}}{2}, \frac{1}{2}, 0$

```
> seq(i^2 mod 7,i=L);
```

$0, 1, 4, 2, 2, 4, 1, 0, 1, 4, 2$

VI.4 Множества

Множества обозначаются с помощью фигурных скобок и по свойствам соответствуют математическим множествам.

```
> set1:={sin,cos,csk,tan};
```

```
set1 := {cos, csk, sin, tan}
```

VI.5 Операции над множествами

- ◎ **op**(<номер элемента или диапазон>, <множество>) – извлечение элементов

```
> op(1..2, set1);  
cos, csh  
> op(4, set1);  
tan
```

- ◎ **nops**(<множество>); - количество элементов в множестве

```
> nops(set1);  
4
```