

Основы программирования в Maple

1. Условный оператор

- Полная форма: `if ... then ... elif ... else ... end if` или `fi`.

```
if x ≠ 2 then print(x=2);  
elif x=3 then print(x=3);  
else print(x);  
end if;
```

2. Операторы цикла

- **Цикл** `for`:
 - С шагом: `for i from 1 by 2 to 9 do ... od`
 - Сокращённый (шаг=1): `for i from 1 to 3 do ... od`
 - Для элементов списка: `for x in [10,20,30] do ... od`

- **Цикл** `while`:

```
while x=2 do x:=x-1; od;
```

- **Комбинированный цикл**:

```
for i from 1 by step while x<3 do ... od
```

3. Итеративные команды

- `seq` – создание последовательностей.
- `add, mul` – сумма и произведение.
- `select, remove, selectremove` – фильтрация элементов.
- `map` – применение функции ко всем элементам.
- `zip` – попарное применение функции.

```
select(isprime, [10..20]); # [11,13,17,19]  
map(x→x^2, [10..20]); # квадраты чисел
```

4. Процедуры

```
имя := proc(параметры::типы)
  local ...;
  global ...;
  options ...;
  description "...";
  тело процедуры;
end proc;

sum2 := proc(a::integer, b::float) a^2+b^2; end;
```

5. Локальные и глобальные переменные

- `local` – переменные, видимые только внутри процедуры.
- `global` – переменные, доступные вне процедуры.

6. Специальные переменные процедуры

- `_passed` – список всех переданных аргументов.
- `_npassed` – количество переданных аргументов.
- `_rest` – «лишние» аргументы.
- `_nrest` – количество «лишних» аргументов.

7. Обработка ошибок

- Использование `error` для генерации сообщений об ошибках.

```
if not type(x, numeric) then error "CHECK INPUT: %1", x; end if;
```

8. Возврат значений

- `return` – завершает выполнение процедуры и возвращает значение.
- Можно возвращать несколько значений через запятую.

9. Дополнительные возможности

- Использование `uses` для подключения модулей внутри процедуры.
- Вывод кода процедуры с помощью `print(имя_процедуры)`.