

**28. Подпрограммы (процедуры и функции) в языках программирования. Способы передачи параметров в процедурах. Программирование рекурсивных алгоритмов.**

Разбиение алгоритма (программы) на вспомогательные алгоритмы (подпрограммы)

Вызов подпрограмм происходит из **ОСНОВНОЙ** программы

У алгоритма есть **аргументы и результаты алгоритма** (данные или действие).

**Параметры** делятся на **формальные** параметры (идентификаторы) \ **фактические** (значения). Их порядок и тип должны соответствовать

**Рекурсия** — это такой способ организации вспомогательного алгоритма (подпрограммы), при котором эта подпрограмма (процедура или функция) в ходе выполнения ее операторов обращается сама к себе. Для ее завершения необходимо условие выхода.

**Глубина рекурсии** - макс. кол-во копий подпрограммы, которое может быть вызвано одновременно

**Рекурсивный подъем** - завершение работы подпрограмм с последней до первой.

**Рекурсивный спуск**

**СПОСОБЫ РЕАЛИЗАЦИИ (подъем)**

```
procedure P; // подъем
```

```
  P
```

```
  <операторы>
```

```
end;
```

```
procedure P;
```

```
  < операторы > //спуск
```

```
  P
```

```
end;
```

```
procedure P; //подъем спуск
```

```
  < операторы >
```

```
  P
```

```
  < операторы >
```

```
end;
```

Пример избавление от рекурсии с помощью стека - префиксный обход.

## МЕХАНИЗМЫ ПЕРЕДАЧИ ПАРАМЕТРОВ

- ссылка
- значение

## СПОСОБЫ ПЕРЕДАЧИ (Pascal)

- параметры-значения;
- параметры-переменные, перед которыми должно стоять служебное слово `var` и за которыми указывается их тип;
- константы - нельзя изменить внутри. нельзя передать др. подпрограмме по значению
- параметры-процедуры \ параметры-функции

```
type
  TMyFuncType = function: integer;
var
  a: TMyFuncType;

function foo: integer;
begin
  result := 1;
end;
function f( p: TMyFuncType ): Integer;

begin
  a := foo;
  f(longint(MyFunc));
```

- нетипизированные параметры, перед которыми должно стоять служебное слово `var`, а указание типа должно отсутствовать.

```
Function Norma (var N) : Real;
begin
  integer(N);
...
```