

# PASCAL

30th September 2003

Ing. Vladimír Čech

# Contents of lecture

1. Syntax and semantic
2. Syntax – basic constructions in Pascal and examples

# Syntax and Semantic

**Syntax** – how write structures to be right from a view of programming language

`a := 1`

`a = 1`

`a == 1`

**Semantic** – what structures mean

`a := 1`

`a = 1`

# Syntax in Pascal

Pascal **isn't** case-sensitive!

`for` is the same like `FOR` or `For`

**Keyword** – special word, which can't be use freely

`begin`, `end`, `for`, `while`, `if`, `do`, `then`, ...

~~`if = 1`~~

# Program structure

**program** <name> ;

**var**

...

**const**

...

**begin**

...

**end.**

# Comments

**Two kinds of comments:**

{ } - brace

(\* \*) - parentheses and asterisk

# Output

```
writeln ( '<your text>' );
```

```
writeln (<variable>);
```

**Eg.:** Write program which writes text 'Hallo World'.

# Input

**program** <name> ;

**var**

...

**const**

...

**begin**

...

**end.**

We can read declared variable only!

Declaration of variable must be introduced here! **If NOT → ERROR**

How to declare variables:

<name> : type ;

type is:

char

integer

boolean

real

string

...

# Input

```
readln (<variable>);
```

**Eg.:** Write program which reads integer number and writes the same number.

# Expressions

Infix notation

operand **operator** operand

Priority of **operator**

- |                    |                    |
|--------------------|--------------------|
| 1. not             | <b>the highest</b> |
| 2. * / div mod and |                    |
| 3. + - or          |                    |
| 4. = < > <> <= =>  | <b>the lowest</b>  |

If you aren't sure of priority use parenthesis

**Eg.** Write program which reads two real number and adds them together and writes the result.

# Conditions

Examples of conditions:

$5 = 6$  .... **false**

$5 \leq 6$  .... **true**

$(5 \text{ div } 2) \geq 1$  .... **true**

$a \neq b$  .... depend on value a and b

$a = \text{true}, b = \text{false}$

$(a \text{ and } b)$  .... **false**

Result of the condition is **true** or **false**.

# Alternative

```
if <condition> then block;
```

```
    block:
```

```
        command
```

```
    or
```

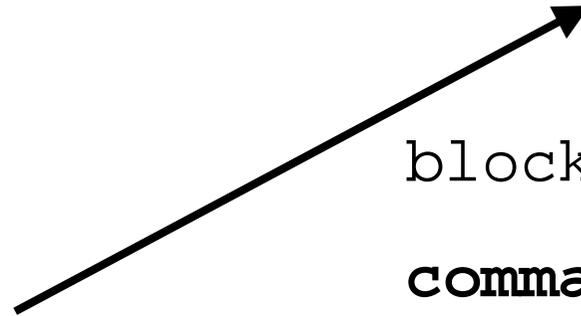
```
    begin
```

```
        commands;
```

```
    end
```

# Alternative 2

```
if <condition> then block else block;
```



```
block:  
command  
  
or  
begin  
  
    commands ;  
  
end
```

Be careful **NO** semicolon  
in front of `else`!

**Eg.** Write program which reads integer number and if the number is small then 10 then writes letter A otherwise writes letter B.

# Cycle FOR

```
for <variable> := <expression> to  
    <expression> do block;
```

```
for <variable> := <expression> downto  
    <expression> do block;
```

```
    block:
```

```
        command
```

```
    or
```

```
        begin
```

```
            commands;
```

```
        end
```

# Cycle FOR

**Eg.** Write program which reads two integers numbers and sum all numbers between these two numbers.

# Cycle WHILE

```
while <condition> do block;
```

```
    block:
```

```
        command
```

```
    or
```

```
        begin
```

```
            commands ;
```

```
        end
```

# Cycle WHILE

**Eg.:** Write program which read one integer number (limit) and next read so many number how is the limit.

# Array

Declaration of array must be in `var` part of program.

1 dimension

```
array [ordinal type] of type;
```

2 dimensions

```
array [ordinal type, ordinal type] of type;
```

# Ordinal type

For each ordinal type must by define functions

predecessor            (pred)

successor              (succ)

ordinal                 (ord)

**In Pascal:**

integer

boolean

char

# Array

**Eg.:** Write program which defined array with 10 elements. Next reads all elements of the array (from keyboard). The output will be sum all odd and even elements. (Program writes two numbers).

# Constants

**program** <name> ;

**var**

...

**const**

...

**begin**

...

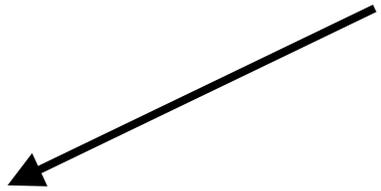
**end.**

Definition of constants must be introduced here!

How to define constant:  
<name> = value;

Eg.:

pi = 3.14;



**That's all for today!**



**Thank you for your  
attention!**