3 "Bench" Programming Language

Friday, January 30, 2026 3:29 PM

Basic Components
- literal — 5 7.2 "hello"
- operators — + * /
- Expressions — 5 + 9 * (3 - 2)
- Variables
- Assignment.
- Conditionals — if swich.
- Repetition — while.

"Functions"
"O.O.P"
"Generics"

- <u>The "PUCK 26.1" Programing Language:</u>

```
% This is a comment
PROCEDURE main ( )
  WRITE ( "Hello" , "World" ) ;
  x := 2 + 2 ;
  y := 3 * 12 / 7.5 ;
  p := ( x > 0 ) AND ~ ( y <= 30 ) ;
  WRITE ( x * 100 )
END main .
```

- Keywords are UPPERCASE
- := Assignment
- Space separation
- Literals
- Operators.
  - ~ negation
  - AND OR Logical operators.
- ; is a "Separator" no a "terminator"
- name repeats.

```
% FizzBuzz
PROCEDURE FizzBuzz ( n )
    IF n MOD 3 = 0 THEN
        IF n MOD 5 # 0 THEN
            WRITE ( "Fizz" )
        ELSE
            WRITE ( "FizzBuzz" )
        END
    ELSE
        IF n MOD 5 = 0 THEN
            WRITE ( "Buzz" )
        END
    END
```

- IF THEN ELSE ELSIF
- MOD modulus
- DIV Integer Division
- =
- #

```
    WRITE ( "Buzz" )
  END
END
END FizzBuzz .
```

```
% Functions and Loops
FUNCTION fibo ( n )
  x := 1 ;  y := 0 ;  c := 3 ;
  WHILE c < n DO
    x := x + y ;
    y := x - y ;
    c := c + 1
  END
RETURN x END fibo .
```

- indentation is not important
- WHILE
- FUNCTION
- RETURN is part of END

x [1 1 2 3 5
y [0 1 1 2 3
c [

n = 6

```
% Greatest Common Denominator
FUNCTION gcd ( a , b )

  WHILE a > b DO
    a := a - b
  ELSIF b > a DO
    b := b - a
  END

RETURN a END gcd .
```

- WHILE has an ELSIF
- conditions are tested sequentially until one is true
  - the corresponding block is executed.

```
% Newlines are just Whitespace
PROCEDURE foo ( s )
  res := True      ;
  IF ( s = "A" ) OR
     ( s = "B" )
  THEN
    WRITE
    (
       "Hello" & "World"
    )
  END
END foo .
```

- & concatenation.

o—o— EOF