Sub - logically complete code fragment, isolated from other software units. Sub can have their own internal data, variables, functions, and so on. It is possible to organize data exchange between the subroutine and other software units. Sub has its own name that can be identified, but which are not due any value (number, symbols, etc.).
Sub operator begins and ends operator SUBROUTINE END (END SUBROUTINE). Sub can be called from another program units, as well as it can cause another program unit. To call subroutines, use the CALL.

Example. HW make routine to output to the console greeting «HELLO, WORLD».

```
SUBROUTINE HW ()
WRITE (*, *) 'HELLO, WORLD'
END SUBROUTINE HW
```

Calling this subroutine from another program unit can be made as follows.

```
CALL HW ()
```

In Fortran direct or indirect recursive call software units (call itself off) is not possible.
Function (or function pidprohama) as routine - a logically complete code fragment, isolated from other software units. The function can have their own internal data, variables, functions, and so on. It is also possible to organize data exchange between function and other software units. It has its own name that can be identified. But the function of different routines that her name associated with the value that the function returns a point challenge. Thus, the function - typed programming unit - it is subject to the rules of appointment naming data types and the rule of silence for data types.
The function begins operator [TYPE] FUNCTION and ends with the operator END (END FUNCTION). If [TYPE] function is specified, it is automatically assigned by the rules of default data types. The function can be called from another program units, as well as it can cause another program unit.
To distinguish between user functions that it develops and compiles user and Fortran standard features that are designed, compiled and placed in libraries FORTRFNu developer compiler.

Example. Using standard trigonometric functions Fortran, make routine to determine the function z = sin (x) + cos (y).

```
FUNCTION Z (X, Y)
Z = SIN (X) + COS (Y)
END FUNCTION Z
```

The name of the function due Z value of type REAL, which, as a result of the calculation is returned to the point of call. So different from call to call subroutines.

Example. Assign value of variable W square function z from the previous example, if x = 0,3 and y = 0,2.

```
W = Z (.3, .2) ** 2
```