Sub data - a software unit that is designed for input (operator nakshtalt DATA), placed in the common areas of the memory access is enabled for software from different units. Typically, these large amounts of data, organize accessed in any other way would be too difficult.

Sub data begins operator BLOCK DATA, comprising operators describing data types, operator describing common memory blocks, DATA operators to set values. Sub completed data provider END (END BLOCK DATA).

As routine, it has its own name that can be identified, but which are not due any value (number, symbols, etc.). Given its purpose, routine data can be called from other software units, and they do not make sense to call another program unit.

Example. Sub data. Typical start and finish.

```
BLOCK DATA DAT
...
END BLOCK DATA DAT
```

Module - specific software unit, which entered into Fortran support capabilities of object-oriented programming. The main purpose of the module - expanding outwardly visible objects.

For procedural programming paradigm, program units have their own local facilities, seating is limited vidimosti program unit. They can not see beyond. Data such facilities can be transferred to another program unit, but her name remain the local. To explicitly declared in the module objects can extend vidimosti names in selected or all software units. So, if you can refer to objects by their names, the software eliminates the need for efforts to arrange Tsikh data objects between software units.

Because of these characteristics, the time of conventional software compilation units, modules must pass preprotsesornu processing. Because the project they should be placed either in a separate file (s) or file with the main program, and, before its beginning.

The module begins operator MODULE, can have any operators, ending the operator END (END MODULE). It has its own name that can be identified, but which are not due any value (number, symbols, etc.). The module can be called from other software units, but can be asotsyyovanyy in the main program, subroutine, function or another module carrier USE.

Example. Write MOD module for advanced zone of visibility variable ALPHA, BETA, array DELTA (2, 20).

```
MODULE MOD
REAL ALPHA, BETA
DIMENSION DELTA (2, 20)
END MODULE
```

In the program units where the planned use of variable ALPHA, BETA, array DELTA (with all or some of them), you must perform the association to the module via the operator MOD USE.

```
USE MOD
```