**Ada**

//================================================================================================ abort     else     new     return      abs     elsif     not     reverse      abstract(Ada95)     end     null          accept     entry         select      access     exception     of     separate      aliased(Ada95)     exit     or     some(Ada2012)      all         others     subtype      and     for     out     synchronized(Ada2005)      array     function     overriding(Ada2005)          at             tagged(Ada95)          generic     package     task      begin     goto     pragma     terminate      body         private     then          if     procedure     type      case     in     protected(Ada95)          constant     interface(Ada2005)         until(Ada95)          is     raise     use      declare         range          delay     limited     record     when      delta     loop     rem     while      digits         renames     with      do     mod     requeue (Ada 95)     xor

**Delphi**

//================================================================================================

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **and** | **array** | **as** | **asm** | **begin** |
| **case** | class  | const  | constructor  | destructor  |
| **dispinterface** | div  | do  | downto  | else  |
| **end** | except  | exports  | file  | finalization  |
| **finally** | for  | function  | goto  | if  |
| **implementation** | in  | inherited  | initialization  | inline  |
| **interface** | is  | label  | library  | mod  |
| **nil** | not  | object  | of  | or  |
| **out** | packed  | procedure  | program  | property  |
| **raise** | record  | repeat  | resourcestring  | set  |
| **shl** | shr  | string  | then  | threadvar  |
| **to** | try  | type  | unit  | until  |
| **uses** | var  | while  | with  | xor  |

**C#**

//================================================================================================

abstract as base boolbreak byte case catchchar checked class constcontinue decimal default delegatedo double else enumevent explicit extern falsefinally fixed float forforeach goto if implicitIn in (универсальный модификатор) int (Целочисленное значение) interfaceinternal is lock longnamespace new Null Objectoperator out out (универсальный модификатор) overrideparams private protected publicreadonly ref return sbytesealed short sizeof stackallocstatic string struct switchthis throw true trytypeof uint ulong uncheckedunsafe ushort using virtualvoid volatile while

**Java**

//================================================================================================

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **abstract** | **continue** | **for** | **new** | **switch** |
| **assert** | default  | goto\*  | package  | synchronized  |
| **boolean** | do  | if  | private  | this  |
| **break** | double  | implements  | protected  | throw  |
| **byte** | else  | import  | public  | throws  |
| **case** | enum  | instanceof  | return  | transient  |
| **catch** | extends  | int  | short  | try  |
| **char** | final  | interface  | static  | void  |
| **class** | finally  | long  | strictfp  | volatile  |
| **const\*** | float  | native  | super  | while  |

**Oberon-?**

//================================================================================================

ARRAY IS TO BEGIN LOOP TYPE CASE MOD UNTIL CONST MODULE VAR DIV NIL WHILE DO OF WITH ELSE OR ELSIF POINTER END PROCEDURE EXIT RECORD IF REPEAT IMPORT RETURN IN THEN

**Modula-2**

//================================================================================================

**pervasive identifier**=
"ABS" | "BITSET" | "BOOLEAN" | "CARDINAL" |
"CAP" | "CHR" | "CHAR" | "COMPLEX" |
"CMPLX" | "DEC" | "DISPOSE" | "EXCL" |
"FALSE" | "FLOAT" | "HALT" | "HIGH" |
"IM" | "INC" | "INCL" | "INT" |
"INTERRUPTIBLE" | "INTEGER" | "LENGTH" | "LFLOAT" |
"LONGCOMPLEX" | "LONGREAL" | "MAX" | "MIN" |
"NEW" | "NIL" | "ODD" | "ORD" |
"PROC" | "PROTECTION" | "RE" | "REAL" |
"SIZE" | "TRUE" | "TRUNC" | "UNINTERRUPTIBLE" |
"VAL" ;

**keyword**=
"AND" | "ARRAY" | "BEGIN" | "BY" |
"CASE" | "CONST" | "DEFINITION" | "DIV" |
"DO" | "ELSE" | "ELSIF" | "END" |
"EXIT" | "EXCEPT" | "EXPORT" | "FINALLY" |
"FOR" | "FORWARD" | "FROM" | "IF" |
"IMPLEMENTATION" | "IMPORT" | "IN" | "LOOP" |
"MOD" | "MODULE" | "NOT" | "OF" |
"OR" | "PACKEDSET" | "POINTER" | "PROCEDURE" |
"QUALIFIED" | "RECORD" | "REM" | "RETRY" |
"REPEAT" | "RETURN" | "SET" | "THEN" |
"TO" | "TYPE" | "UNTIL" | "VAR" |
"WHILE" | "WITH" ;