

Application and Development of Computer Software Development Technology

Xiong Chuixiang

Hankou University, Wuhan city, Hubei Province, China

Keywords: computer software; development technology; application;

Abstract: Computer software development technology involves high-level programming languages such as VB. It flexibly uses programming knowledge such as constants, variables, operators, expressions, functions, basic control statements, algorithms and data structures to achieve the target output through input, and finally completes the realization of software functions. Starting from the basic knowledge of computer software development, this paper discusses the basic operation and future development trend of computer software development.

1. Introduction

Under the background of “Made in China 2025” and “Internet+XX Project”, the degree of informationization is deepening day by day. All walks of life are increasingly using computers to process information and complete related work. The benefits of informationization and digitalization are enormous, and many tasks that used to cost a lot of manpower and material resources can now be easily completed. In mechanical engineering, civil engineering, chemical engineering and other engineering fields as information processing tools, the computer seems to have a more magical power, several times or even dozens of times to improve the design efficiency, greatly improve the quality of the project, collaborative design, virtual design and simulation omnipotent, this magic power comes from software! It has become an indisputable fact that the problems of domain science are mathematically, numerically and programmatically. With the development of computer software technology, software has entered the work, study and life of every engineer and technicians. It can be said that the efficiency of leaving the software enterprise will be greatly reduced or even paralyzed. This paper describes some basic technologies in software development for engineering design and some related knowledge needed to be mastered by software developers. It also discusses some basic characteristics of software and briefly introduces the application of software technology in engineering.

2. Advanced Programming Language

With the advent of computers, people have designed a computer language to manipulate computers. The original language was machine language. Machine language opens up two areas in memory: data area to store data and instruction area to store instructions. The CPU starts from the first address in the instruction area and executes instructions one by one until all instructions are executed. The general instruction format is as follows:

Table 1 instruction schematic

Operation code	Operands	Operands	Remarks
00000100	10100001	00101110	Binary instruction
04H	A1H	2EH	Hexadecimal instruction
ADD	AX	2EH	Assembler instruction

The machine can “read” the instructions above, that is, add 46 to the current number in the accumulator. Because the convention operation code “00000100” means “add”, that is, add the two operands “10100001” and “00101110” after it. Among them, the second operand “00101110” is the accumulator code, which stores the operands. Obviously, it is very boring and error-prone to use

binary code for machine programming, which is unacceptable to people. So at the beginning, people use octal number (O) or hexadecimal number (H) to express binary code instructions (when input program, it must be converted into binary code). It was soon found that replacing the agreed instructions with memorable English words made it easier to read and write programs, which led to the birth of assembly language. Obviously, the assembler also needs to be converted into binary code, which is done automatically by the assembler. Assembler has high efficiency and is still in use.

Assembly language is machine-oriented. In assembly language programming, it is necessary to directly arrange the order of action of storage, register and calculator. It is also necessary to know the computer's representation of data conventions (fixed point, floating point, double precision) and so on. For most people, this is not a simple matter. In addition, although assembly language Abstracts some opcodes and registers, it has a close relationship with computers. Because of the difference of instruction length, addressing mode, register number and instruction representation of different computers, assembler is not only not transplantable, but also difficult to understand directly. These shortcomings of assembly language create conditions for the emergence of high-level language.

Machines only know how to execute programs according to machine language instructions, so high-level language programs must be translated into machine language programs through translation. This work is usually done automatically by a translation program. A program that translates one language into another is called a translator (for example, C++ to C). There are two ways to translate high-level language programs into machine language programs: compiling and interpreting. The corresponding translation tools are also called compilers and interpreters.

High-level language programming is also computational data, that is, the input data is calculated by expression to get the output data. But the data, especially the calculation results, are not involved in the calculation. People can only express it in variables. For example, the triangle trilateral length is known for calculating area:

$$S = 0.5 * (a + B + c)$$

$$\text{Area:} = \text{strt} (s*(-a)*(-b)* (s-C))$$

Among them, a, b, c, s, area are variables, 0.5 is constant, “+” and “*” are operators, and sqrt is square root function, which constitute the expression. “:=” is an assignment number. Expressions are sequences of constants, variables, function calls, and operators. Obviously, variables in an expression are assigned before they are computed. When writing a program, the variable at the left end of the assignment number may not have a value, and even if it has a value, it will be assigned a new value after executing the calculation. The advantage of writing programs with variables is that they are more general. The above statement can calculate not only the triangular areas of 3, 4, 5 trilateral lengths, but also other triangular areas (as long as the correct values of a, b, C are input). Therefore, as long as the calculation has a clear mathematical expression (mathematical model), programming is relatively easy.

Assignment and function call are the basic means for program language to change the value of variables. Different languages use different assignment numbers, such as PASCAL language assignment number is “:=”, while in C, Java, Visual Basic (VB) and other languages, assignment number is “=”.

The statements in the program correspond to one or more instructions of the computer. In some languages, expressions can not be divided into statements, such as PASCAL; some languages have no concept of statements, only expressions, such as LISP; and some languages have expressions and statements, such as C. In C language, I + + is not only an expression, but also a statement called expression statement.

3. VB Language Foundation

VB is one of the commonly used advanced computer software programming languages. Any programming language has its own data type. VB not only provides abundant standard data types, but also allows users to define their own data types. VB system defines six standard data types,

including date type, numerical type, character type, logic type, object type and variant type. Among them, numerical type includes single precision type, integer type, long integer type, double precision type and byte type.

There are two forms of direct constant and sign constant in VB. They are string constant, numeric constant, Boolean constant and daily constant. In addition, they can be defined by octal and hexadecimal numbers. Symbolic constants are constants represented by symbols, including both system-defined constants and user-defined constants. Variables are the amount of value that can be changed when a program is running. Programs often use and store data through variables. Each variable corresponds to a given memory unit, which can store a data, and the data is typed, so the variables are typed. Variables are usually declared before they are used. By declaring, the program knows the name of the variable and the data type of the variable, so that the VB system can allocate memory units for it and use it.

There are various expressions in computer programming languages. Expressions consist of operators and operands. Since there are various types of operands, there are corresponding types of operators and expressions. In VB, the following operators and expressions are shown in Table 2:

Table 2 Operators and Expressions in VB

Arithmetic operator	Arithmetic expression
Relational operator	Relational expression
Logical Operator	Logical expression
String operator	String expression
Date operator	date expression

Function is a kind of special operation. There are all kinds of functions in advanced programming language, which brings great convenience to programming. There are two kinds of functions in VB, namely internal function and custom function. User-defined functions are functions defined by users according to their own programming needs. These functions generally have special functions and can not be found in internal functions. Internal functions are defined functions of VB system, also known as standard functions. These functions are very convenient to use. Users do not need to understand the internal processing process of functions, just give the name of functions and appropriate parameters. VB provides a large number of internal functions, which can be divided into: mathematical operation function, string function, data type conversion function, date and time function and format output function.

Like other programming languages, VB language has its own coding rules. The main coding rules of VB program code are:

- (1) In VB code, letters are case-insensitive.
- (2) VB automatically converts keywords to uppercase letters, and the rest are all lowercase letters.
- (3) Customize variable name, procedure name and function name. VB is based on the first-second definition.
- (4) Multiple statements can be written in one line, separated by colons. For example, t = a: a = b: B = t
- (5) If a statement cannot be written in one line, it can be divided into several lines followed by a continuation character (a space followed by an underscore).

4. Basic control structure

Like other advanced programming languages, VB has three basic structured programming structures: sequence structure, selection structure and loop structure, which are the basis of programming.

Sequential structure is the simplest and most commonly used program structure. The program of this structure is executed sequentially according to the order in which statements appear. The main statements used in sequential structured programs are assignment statements and input/output

statements. If a program does not have the ability to output data, it is meaningless. VB provides a variety of methods to output data, such as Print method, label control and text box control. There are also many ways to input data in VB, the most common way is to use text box, in addition, you can input through the input dialog box.

Selection Statement of Selection Structure Conditional Statement If... Then... Else, If... Then... Nested and multi-branch conditional selection statements of Else If and IF conditional statements. VB also provides a variety of statements for designing loop structure programs. Among them, do is the most commonly used one. Loop and For... Next statement. In addition to Stop, End and Exit statements, there are Go To and With statements which are often used in programs.

5. Algorithms and Data Structure

When there is data representation, data operations should be considered, and these operations can be effectively integrated to complete various tasks, that is, algorithms. The algorithm is a description of the steps of solving a particular problem and a finite sequence of instructions. Each instruction represents one or more operations. The algorithm should have the following characteristics:

(1) Finiteness: The algorithm must always end after a finite step (for any valid input value), and each step can be completed in a finite time.

(2) Certainty: Every instruction in the algorithm must have a definite meaning without ambiguity. Under any conditions, the algorithm has only one execution path, that is, the same output can only be obtained for the same input.

(3) Feasibility: The algorithm is feasible, that is to say, the operations described in the algorithm can be implemented by a finite number of basic operations that have been implemented.

(4) Input: The algorithm has zero or more inputs, which are taken from a specific set of data objects.

(5) Output: The algorithm has one or more outputs, which have a specific relationship with the input.

The algorithm describes the process of solving a problem. There are many kinds of algorithms to describe the same problem. There are many kinds of problems and more algorithms. Generally, algorithms can be divided into two categories: numerical algorithm and non-numerical algorithm. Common algorithms include accumulation, multiplication, primes, exhaustion, recursion, maximum or minimum.

The commonly used data structures are linear table, tree, binary tree and graph. The operation of data is defined on the logical structure of data, but the concrete implementation of the operation should be carried out on the storage structure. Generally, there are the following common operations:

(1) Search. Retrieval is to find nodes satisfying certain conditions in the data structure. Generally, given the value of a field, a node with the value of that field is found.

(2) Insert. Add new nodes to the data structure.

(3) Delete. Remove the specified node from the data structure.

(4) Update. Change the value of one or more fields of the specified node.

(5) Sorting. The deduction share is rearranged in the order specified by its species. For example, increase or decrease.

6. Conclusion

The basic course of computer software technology application in university courses generally refers to data structure. For the application of computer software technology, interface design, graphics design, file and database technology are only offered in computer-related professional courses, rather than as compulsory courses for students of other engineering colleges. That is to say, the application of computer software technology only learns the data structure, and has preliminary foundation of software writing. However, the knowledge in the field of AI in-depth learning has not

been popularized in universities. In the future, this field will become a new field of software technology innovation.

References

- [1] Euro Gaoquan. Application and Development Trend of Computer Software Development Technology [J]. Electronic Technology and Software Engineering, 2019 (12): 47.
- [2] Zhang Huan. Development and Application of Computer Software Technology [J]. Information and Computer (Theoretical Edition), 2019 (12): 13-14.
- [3] Liao Chunxiu. Application and Trend of Computer Software Development Technology [J]. Electronic Technology and Software Engineering, 2019 (11): 36.