

7. Software engineering

Software development environment

Software development environments, also known as Integrated Development Environments (IDE), provide programmers with various tools that are needed to create computer programs. Here are some of the tools and facilities offered by a typical software development environment:

Facility	Use
Editor	Allows a programmer to enter, format and edit source code
Compiler	Converts source code into executable machine code. Once compiled, a program can be run at any time
Interpreter	Converts each line of source code into machine code, and executes it as each line of code is run. The conversion process is performed each time the program needs to be run
Linker	A program which allows previously compiled code, from software libraries, to be linked together
Loader	A program which loads previously compiled code into memory
Debugger	A program which helps locate, identify and rectify errors in a program
Trace	A facility which displays the order in which the lines of a program are executed, and possibly the values of variables as the program is being run
Break point	A facility which interrupts a program on a specific line of code, allowing the programmer to compare the values of variables against expected values. The program code can then usually be executed one line at a time. This is called <i>single-stepping</i>
Variable watch	A facility that displays the current value of any variable. The value can be 'watched' as the program code is single-stepped to see the effects of the code on the variable. Alternatively a variable watch may be set, which will interrupt the program flow if the watched variable reaches a specified value
Memory inspector	A facility which will display the contents of a section of memory
Error diagnostics	Used when a program fails to compile or to run. Error messages are displayed to help the programmer diagnose what has gone wrong

Libraries

A library is a collection of commonly used private functions and subprograms. Examples of private functions include standard mathematical operations such as square root, random number generators. Examples of subprograms include standard input/output routines, such as saving data to disk. These functions and subprograms can be called from within your program at any stage, but only when the appropriate library has been linked.

The advantages of using standard libraries are that:

- related private functions and subprograms are stored in the same location
- time is saved as the programmer can simply use the private functions and programs stored in a library
- subroutines contained in a library have already been tested, so they should work reliably and not need further testing
- programs will contain less code and will therefore be easier to maintain

Most computer languages use standard libraries, although it is also possible to create your own custom libraries.