

УДК 004.424

Danil Arkhypov<sup>1</sup>, Nataliia Zhukova<sup>2</sup>

<sup>1</sup>student of group CST-220, National University «Zaporizhzhia Polytechnic»

<sup>2</sup>PhD (Philology), assistant prof. National University «Zaporizhzhia Polytechnic»

## STRUCTURED PROGRAMMING

Structured programming (sometimes known as *modular programming*) is a programming paradigm that facilitates the creation of programs with readable code and reusable components. All modern programming languages support structured programming, but the mechanisms of support, like the syntax of the programming languages, varies.

Where modules or elements of code can be reused from a library, it may also be possible to build structured code using modules written in different languages, as long as they can obey a common module interface or application program interface (API) specification. However, when modules are reused, it's possible to compromise data security and governance, so it's important to define and enforce a privacy policy controlling the use of modules that bring with them implicit data access rights.

Structured programming encourages dividing an application program into a hierarchy of modules or autonomous elements, which may, in turn, contain other such elements. Within each element, code may be further structured using blocks of related logic designed to improve readability and maintainability. These may include “case”, which tests a variable against a set of values; “Repeat”, “while” and “for”, which construct loops that continue until a condition is met. In all structured programming languages, an unconditional transfer of control, or “goto” statement, is deprecated and sometimes unavailable.

The base composition of any type of structured programming includes three fundamental elements, sequencing, selection, and repetition.

Sequencing has to do with the logical sequence provided by the statements in the program. As they are executed, each step in the sequence must logically progress to the next without producing any undesirable effects.

Selection allows singling out any number of statements to execute in the program. These statements will contain certain keywords that can identify the sequence as a logically ordered executable. These terms are “if,” “then,” “endif,” or “switch.”

As a program proceeds, a select statement continues to be active until the program gets to the point where some other action needs to take place. The keywords include “repeat,” “for,” or “do...until.” The repetition factor dictates instructions to the program about how long to continue the operation before requesting further instructions.

Depending on the purpose and function of the program, the exact nature of structured programming will vary. For example, most forms of structured programming will have a single entry point but may have more than one exit point. In another form, called modular programming, the creation of modules within the overall structure of the program will interact with one another, depending on the type of code that is executed.

The advantages of structured programming are as follows:

1. It is user friendly and easy to understand.
2. Similar to English vocabulary of words and symbols.
3. It is easier to learn.
4. They require less time to write.
5. They are easier to maintain.

These are mainly problem oriented rather than machine based.

Programs written in a higher level language can be translated into many machine languages and therefore can run on any computer for which there exists an appropriate translator.

It is independent of the machine on which it is used i.e. programs developed in high level languages can be run on any computer.

The disadvantages of structured programming are the following:

1. A high level language has to be translated into the machine language by the translator and thus a price in computer time is paid.

2. The object code generated by a translator might be inefficient compared to an equivalent assembly language program.

Data types are proceeds in many functions in a structured program. When changes occur in those data types, the corresponding change must be made to every location that acts on those data types within the program. This is really a very time consuming task if the program is very large.

Let us consider the case of software development in which several programmers work as a team on an application. In a structured program, each programmer is assigned to build a specific set of functions and data types. Since different programmers handle separate functions that have mutually shared data types. Other programmers in the team must reflect the changes in data types done by the programmer in data type handled. Otherwise, it requires rewriting several functions.

Structured programming is a major paradigm that helps to develop a program using a set of modules or functions. Structured programming began the theoretical study of programming design. It forced programmers to approach programming from the conceptual point of view, making them design programs before they wrote code. It cleared up spaghetti code and forced the concept of correctness into the program.