Generic functions

Raluca Oana Scarlatescu

The programs complexity, their high quality and the short time requested for writing and implementing them ask now for a new approach that is oriented to the reusing of the algorithms or parts of them, after rebuilding them with an elevate degree of standardisation.

Frequently a programmer performs the same logic with different types. In this case the standardisation process of the algorithms refers to the data types associated with the functions' parameters. Sometimes, the generality of the algorithms can be obtained using functions that allow reusing programs by adding a small piece of type-specific code. The choice of the applied method depends on the facility of the programming language (in C/C++, by example, there are some well- known techniques to do this: using pointers, processing directive, templates or virtual functions). Some of these suppose that the data types are known at the processing moment.

The paper introduces another solution in order to build generic functions, when the data types are unknown. The solution uses a matrix of standard types. The functions are written such as they include the correct behaviour for each considered type in the matrix. During the execution, after the identification of the data type, the function will choose the part of the algorithm adapted for it. Some examples are included in the paper.

The method «Programming by steps» [1] needs generic macro-functions in order to work efficiently. The paper presents a solution for the memory allocation that is necessary to be used for the parameters of a generic function, when the main application doesn't know the type of the parameters, but only the number of them. The main application will read the parameters' values from a database during the execution and will apply the same process for each function included in the logical flow, indifferently to the number or types of their parameters. The paper refers to a generic function necessary for reading the information from a database table using a programming language, other that the database' one.

The paper defines a "cell" in a table as the intersection between a record and a column where the elemental information is memorised, it describes the cell's characteristics and different approaches in order to retrieve the information, and it represents graphic the cell. A generic function is built, in order to read the data of the cell and to scatter it in the function parameter. The paper contains examples built in SQLServer2000 as database management system and C/C++ as programming language. An inherited class is presented, that uses a generic function in order to read data from a cell, indifferently of the data type, and to solve some problems occurred using the parent class (CRecordSet).

The method «Programming by steps» extends the notion of a generic function at a conceptual level: a function becomes an independent part of the algorithm that has to be executed without importance on how it runs or what it generates. Using the same structures and data memorised for the method «Programming by steps», but a different main application, the graphical representation of the logical flow is obtained. The generic functions are drawn, together with their parameters, linked in the logical succession foreseen by the flow. The conclusions refer to the domain of applicability of the generic functions.

References

[1] R.O. Scarlatescu, Programming by steps. Acta Cybernetica, Volume 16, Number 2, 2003, p. 293-314.