

USE OF PROGRAMMING LANGUAGES IN SOLVING TECHNICAL PROBLEMS

Axmedova Barno Abdievna

Senior teacher, Karshi Engineering Economics Institute, **UZBEKISTAN**

E-mail address: finaledition2@mail.ru

ABSTRACT

This article provides recommendations for the use of modern programming languages in the field of higher education technical specialties, as well as to solve these problems on the example of the C ++ programming language. In addition, the C ++ programming language describes the array, its types, capabilities, array problems and their programming solutions.

Keywords: C ++, low-level programming language, high-level programming language, operators, array, multidimensional array, static array, dynamic array, functions, alloc, malloc, size of, null, new.

INTRODUCTION

In the modernization of the education system, first of all, education is broad it is necessary to understand that democratization, humanization, as well as the fact that the education system can meet the laws of development.

The use of computer and information technology in the performance of technical specialization guarantees high efficiency of the results of the given tasks. Computer and information technology require a certain level of knowledge and skills from their users to communicate. The high level of this knowledge and skills determines the quality level of communication. The solution to this problem is one of the most pressing issues in the field of computer science, information technology and its teaching methods in primary, secondary special and higher education.

Decades ago, the science of informatics, whose main task was the basics of programming, today is divided into programming languages, information technology, network technology, teaching technology and a number of other independent disciplines, based on the modern requirements of education. Although all of these disciplines are logically closely related to each other, they are designed to study a specific field.

MATERIALS AND METHODS

Any program that requires a computer requires the development of special software. For the problems that need to be solved with the help of computers - the fact that the development of programs using old programming tools is not economically justified, and the complexity of the program development process creates the need for new programming languages. This problem is solved by experienced programmers or groups of programmers. The main task of teachers of the basics of programming in educational institutions was to train such programmers, to effectively organize the process, such as teaching them the necessary information (programming basics, programming languages, information technology, etc.) to use them in their future careers.

Teaching a number of programming languages to students of technical specialties and related fields of study for use in their professional activities is also provided for in a number of normative documents of the DTS (State Education Standard). According to them, in the teaching of programming languages, the teaching of fundamental concepts related to issues such as the representation and processing of data in different forms is of paramount importance.

Earlier generations of computers were able to perform only the simplest arithmetic and logic operations during operation. By now, a computer will need to have a compatible program installed in its memory to solve complex problems. It is known that a program is a specific sequence of machine codes that controls the application of a specific computing tool. Hundreds of programming languages have been created to facilitate the software creation process. All programming languages can be divided into two categories:

- Low-level programming languages;
- High-level programming languages;

The C ++ programming language is associated with the Unix operating system, and many of the programs used in this system are written in C ++. The C ++ programming language allows you to take full advantage of all the features of the machine. The C ++ language is common in terms of purpose, it occupies an intermediate position between high-level programming languages and Assembler. Program control operators meet the requirements of structured programming. It does not have input and output, dynamic memory allocation, multitasking, parallel computing. These tools are implemented through external functions [1].

Programming in C ++ allows you to understand the operation of a computer and its operating system, so it is intended for programmers, but the C ++ programming language is not associated with any system or machine. In C ++, a program is an action taken to get results on data. In the program, actions are given by operators, and data is performed by identifying and describing objects. Each object used in the program must be described. The description associates several characteristics with the object. These characteristics are: type, designation, memory class, area of motion, initial values, and so on.

An array is an ordered set of one-category, finite values. Examples of arrays are vectors, matrices, which are known from the mathematics course. Arrays are usually divided into one-dimensional and multidimensional, dynamic and static types. An array is said to be one-dimensional if its element can be referenced by an index. The indexes of array elements in C ++ programming languages always start from zero (not all at once).

Multidimensional static arrays. In C ++, there are no restrictions on the type of an array element, but these types must be of a finite size. Because the compiler must be able to calculate how much space (bytes) of the array takes up memory. In particular, the component of an array can be an array ("vector of vectors"), resulting in a two-dimensional array called a matrix. If the element of the matrix is also a vector, three-dimensional arrays are formed. In this way it is possible to create arrays of arbitrary size depending on the problem to be solved. In a two-dimensional array, the first index indicates the number of rows and the second the number of columns.

Working with dynamic arrays. The disadvantages of static arrays are that their dimensions must be known in advance, moreover, these dimensions are limited by the size of the memory segment allocated to the given ones. On the other hand, by declaring an array large enough, the allocated memory may not be fully utilized in solving the problem. These shortcomings can be overcome by using dynamic arrays, as they allow you to create arrays of the size you need

during program execution and eliminate them when you no longer need them. You can use the malloc (), calloc () functions, or the new operator to allocate memory to dynamic arrays. The delete operator is used to free up memory allocated to a dynamic object [2].

RESULTS AND PRACTICAL APPLICATIONS

The above features are located in the << alloc.h >> library. The syntax of the Malloc () function is in the form Void * malloc (size_t size), which separates a continuous byte-sized area from the stack of memory. If memory allocation is successful, the malloc () function returns the start address of the allocated area. If the required memory allocation is successful, the function returns a NULL value.

As you can see from the syntax, the function returns a value of type void. In practice, it is necessary to allocate memory for the object. To do this, use a specific type of void conversion technology. For example, allocating space to an array of length 3 for an entire type can be done as follows:

```
int * pint = (int *) malloc (3 * sizeof (int));
```

Unlike the malloc function, the Calloc () function initializes array elements with a value of 0 in addition to allocating space for the array. The syntax of this function is: Void * calloc (size_t num, size_t size); In the view, the num parameter tells you how many elements are in the allotted area, the size of each element. The Free () memory release function has a single parameter that is an indicator of the partition to be erased: Void free (void * block); Having the Free () function in the void type allows you to disable any type of memory. The following program performs the creation, assignment, and deletion of a 20-digit dynamic array.

```
#include <iostream.h>
#include <alloc.h>
int main () {
int * pvector;
if ((pvector = (int *) malloc (20 * sizeof (int))) == NULL) {
Cout << "memory is not enough";
return 1; } }
// fill in the allocated memory area
for (int i = 0; i <10; i ++) * (pvector + i) = I;
// hop vector elements
for (int i = 0; i <10; i ++) cout << * (pvector + i) << endl;
// return (delete) the allocated memory piece
free (pvector);
return 0;}
```

Using the New operator, when allocating memory to an array, the number of objects in square brackets is displayed after the object type. For example, to allocate space to an array of 20 numbers of the whole type pVector = new int [20]; expression must be written. Conversely, you will need to give the delete [] pVector instruction to free up memory allocated in this way. Int ** a to create a two-dimensional dynamic array; The "pointer to pointer" view is used.

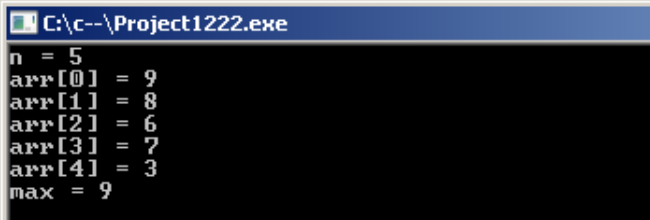
Depending on the number of rows in the other array, the array of pointers must be allocated from dynamic memory: a = new int * [m] // where m is the number of rows in the array. : for (int i = 0; i <m; i ++) a [i] = new int [n]; // n is the number of columns. It should be noted that each line of the dynamic array can be located in different places in memory [3].

When deleting a two-dimensional array, first each element (string) of the array, then the array itself, is lost. for (i = 0; i <m; i ++)
delete [] a [i]; delete [] a.

Below we will look at another problem with arrays, finding the largest of the array elements:

```
1. #include <iostream>
2. using namespace std;
3. int main ()
4. {
5. int * arr;
6. int size;
7. cout << "n =";
8. cin >> size;
9. if (size <= 0) {
10. cerr << "Invalid size" << endl;
11. return 1;
12.}
13. arr = new int [size];
14. for (int i = 0; i <size; i ++)  
{
15. cout << "arr [" << i << "] =";
16. cin >> arr [i];
17.}
18. int max = arr [0];
19. for (int i = 1; i <size; i ++)  
{
20. if (arr [i]> max) {
21. max = arr [i];
22.}}
23. cout << "max =" << max << endl;
24. delete [] arr; // free memory
25. return 0;
26.}
```

RESULT



```
C:\c--\Project1222.exe
n = 5
arr[0] = 9
arr[1] = 8
arr[2] = 6
arr[3] = 7
arr[4] = 3
max = 9
```

CONCLUSIONS AND SUGGESTIONS

One of the most pressing problems in the higher education system is the formation and development of knowledge and skills of students in technical specialties on new programming technologies, as well as the analysis and improvement of teaching methods in computer science and information technology, algorithmic and programming languages [5].

Working in the C ++ programming language is more convenient and more versatile than other programming languages. Through this article, you will gain the necessary knowledge and skills about the structure of a C ++ program, character description, algorithm and program concept, data input and output operators, and how to work with arrays and strings in a program. C ++ program is a tool that works in a Windows environment and is easy to program, automates programming on the computer, reduces errors and simplifies the work of the programmer. The

use of arrays in the C ++ programming language has several advantages over other programming languages. How easy it is to work with arrays in some computational problems. If we take into account their space in the memory, we can see that the speed of the program has increased several times.

Based on the above, this article provides general information about arrays (one-dimensional and multidimensional), their memory layout, sorting, sorting, associating with functions, and using them in applications. In this way, it will be possible to effectively solve the problems of technical specialties using programming languages.

REFERENCES

1. Nazirov Sh.A., Qobulov R.V., Bobojonov M.R., Rakhmanov Q.S. C and C ++ languages. - Tashkent: "Heir", 2013. - 488-p.
2. Information technology in technical systems [Text of lectures] Part 2: textbook / M.M. Kadirov. - Tashkent: "National Society of Philosophers of Uzbekistan" Publishing House, 2019. - 306 p.
3. B. Straustrop. "Summer C ++ programming. "Binom press. - 2015. - p .369.
4. Kjell Backman. Structured Programming with C ++. BookBoon. - USA, 2012. - P.246.
5. <https://dastur.uz>