

Assignment for the laboratory work № 6. “The STL generic algorithms”

This laboratory concerns the study of the STL generic algorithms. First of all, you should create a class describing some real object of our world (person, car, elementary particle, etc.). Such a class must contain at least two data members: its name (string type) and its characteristic parameter (age, speed, etc) of the double type. Equality operator `==` must be overloaded for names. Create an array of objects of the designed class. Parameter's values may be defined at random. Create a vector of objects of the designed class by applying the parameterized constructor and copying the array's contents into the vector. Develop a class *MyPrint* that would have a function call operator overloaded to print your class data members: the name and parameter.

Further, implement the following steps:

1. Apply the generic algorithm *for_each* to print the vector contents.
2. Find the first object in the vector whose parameter would be greater than a prescribed value. To do this, create your own class for getting predicate function objects, and then apply the generic algorithm *find_if*. Print the object found.
3. Try to find objects having the same names and being placed in your sequence successively. Apply the algorithm *adjacent_find*, and print the first such an object if any.
4. Try to check the equality of the contents of your original array and the vector, applying the algorithm *equal*.
5. Apply the algorithm *search* to establish location of the arbitrary subsequence in the vector range. Define this subsequence as a part of the array. Print the subsequence found.
6. Apply the algorithm *accumulate* in order to calculate the average (mean) value of the parameter data member of the vector's elements. You have to develop your own class *MyBinOp* that provides the relevant binary operation.
7. Create a new vector *v2* of double values with the same size as the original one, and fill in it with the parameter values from the first vector. You should use the algorithm *transform* and create a function object of your own class *MyUnOp* for providing the relevant unary operation (for getting the parameter's values).
8. Replace all old values x_i of the vector *v2* with the new ones $y_i = x_i - m$, where m is the mean value calculated at the step 6. To do this, you should use the algorithm *transform* again, having created before your class *MyFunc* to implement the necessary binary operation.
9. Sort the contents of *v2* in ascending order. Apply the *sort* algorithm and print the result.