

PCLP2

Laboratorul 12

1. **Standard Template Library (STL).** *STL* este o bibliotecă generică de structuri de date și algoritmi pentru limbajul C++. STL implementează toate aceste elemente cu ajutorul claselor template. Programul de mai jos ilustrează folosirea clasei `vector` instanțiată pentru obiecte de tip `string`. Parcurgerea elementelor din vector se realizează: a) prin folosirea indicilor elementelor vectorului; b) cu ajutorul unui iterator de tip `const_iterator` care parcurge elementele în ordinea în care sunt stocate în vector; și c) cu ajutorul unui iterator de tip `reverse_iterator` care parcurge elementele în ordine inversă.

```
test_vector.cpp
#include <iostream>
#include <vector>

using namespace std;

main()
{
    vector<string> e;
    e.push_back("Primul element");
    e.push_back("Al doilea element");
    e.push_back("Al treilea element");

    cout << "Citirea elementelor dupa indice:" << endl;
    for(int i=0; i < e.size(); i++)
        cout << e[i] << endl;

    cout << endl << "Citirea elementelor cu iterator:" << endl;
    vector<string>::const_iterator it;
    for(it=e.begin(); it!=e.end(); it++)
        cout << *it << endl;

    cout << endl << "Citirea elementelor in ordine inversa:" << endl;
    vector<string>::reverse_iterator rit;
    for(rit=e.rbegin(); rit!=e.rend(); ++rit)
        cout << *rit << endl;

    return 0;
}
```

2. Clasa `map` din *STL* implementează matricile asociative. Scrieți un program care să păstreze asocieri de forma "student – notă". Încărcați în container 5 elemente, apoi parcurgeți-l cu un iterator și afișați-i conținutul. Folosiți clasa `map<string, int>` pentru a defini obiectul de tip `map` și iteratorul `map<string, int>::iterator` pentru a parcurge containerul. Pentru mai multe informații referitoare la acest subiect, studiați și secțiunea "Containerul asociativ `map`" din capitolul 10 al cursului.