**Şcoală:** Colegiul Naţional „Spiru Haret”

**Profil:** Real

**Specializare:** Matematică Informatică Engleză Bilingv

**Materie:** Informatică

**Avantajele utilizării STL**

**Profesor coordonator:** Boian Flavius

**Elev:** Sandu Denisa

Târgu-Jiu

2014

**Cuprins**

|  |  |
| --- | --- |
| 1. Argument | Pag 3 |
| 1. Introducere | Pag 4 |
| 1. Codul sursa | Pag 5 |
| 1. Concluzie | Pag 28 |
| 1. Bibliografie | Pag 29 |

**Argument**

Motivul pentru care am ales să prezint avantajele utilizării STL (Standard Template Library) este că sunt de părere ca funcţiile predefinite pot schimba în totalitate viziunea despre programarea într-un limbaj.

Biblioteca standard şi funcţiile ei reuşesc să comprime codul, îmbunătăţind considerabil timpul de implementare si reducând bug-urile ce se infiltrează în momentul în care sursa este complexă. Pe lângă simplitaea pe care o aduc, funcţiile STL prezintă întotdeauna performanţă deosebită atât din punct de vedere al timpului de execuţie, cât şi al memoriei utilizate, dovedindu-se astfel că înlocuirea structurilor complicate, standard, cu funcţii STL este o decizie pertinentă, cu precădere în cazul pregătirii pentru olimpiade şi concursuri şcolare.

Astfel, am hotărât să abordez acest subiect pentru a explica în detaliu performanţele ce se obţin înlocuind codul scris în limbaj simplu cu funcţiile predefinite STL, realizând o comparaţie între cele două situaţii şi prezentând avantajele şi dezavantajele pe care le deţine fiecare dintre cele două opţiuni.

**Introducere**

***Standard Template Library*** (STL) este o biblioteza generica, ce furnizeaza solutii pentru manipularea colectiilor de date, precum si algoritmi eficienti care actioneaza asupra acestor date. Biblioteca STL este parte integranta a Bibliotecii Standard C++, imbogatind limbajul C++ cu instrumente extrem de utilizate in practica: containerele generice si algoritmii generici.  
Lucrarea mea de atestat va cuprinde, in general, modalitatile deosebit de practice prin care STL-ul se aplica in programare, inlocuind algoritmi clasici predati in timpul orelor de informatica si va explica modalitatile de optimizare a codurilor respective.

**Informatii generale**

* ***Biblioteca Standard C++*** contine un set de clase, functii si macrodefinitii care extind nucleul limbajului.
* Scopul ***STL*** este de a oferi algoritmi eficienti si colectii de date care sa fie independente de structurile de date particulare.
* Componentele de baza ale STL sunt: clasele container, iteratorii si algoritmii.
* ***Algoritmii*** s-au implementat in STL ca functii generice care actioneaza asupra elementelor unei colectii de elemente (fie acesta tablou de tip C sau container C++). Eficienta algoritmilor este garantata. S-au depus eforturi deosebite pentru ca viteza de lucru a algoritmilor STL sa fie la fel de mare cu a celor mai optimi algoritmi scrisi "de mana".
* Algoritmii constituie o componenta fundamentala a STL. SUnt functii sablon ai caror parametri sunt iterarori. Cu ajutorul iteratorilor se realizeaza independenta algoritmilor fata de tipurile de date procesate. Un acelasi algoritm poate lucra cu diferite tipuri de containere predefinite ori cu containere definite de noi insine, in particular cu tablouri, cu conditia ca specificatiile iteratorilor sa fie respectate.
* Algoritmii ofera servicii de baza, cum ar fi: *copiere, modificare, mutare, cautare, sortare sau procesare numerica.*

**Codul Sursă**

<head>

<link rel="stylesheet" type="text/css" href="style.css" />

<title>Avantajele utilizarii STL</title>

</head>

<body>

<div id="container">

<div id="headerimg6">

<ul id="menuu">

<li>

<a href="index.html">Introducere</a>

</li>

<li>

<a>Sortare</a>

<ul>

<li><a href="sortare.html">Sortarea clasica</a></li>

<li><a href="sort.html">STL Sort</a></li>

<li><a href="sortall.html">Compararea complexitatii</a></li>

</ul>

</li>

<li>

<a>Cautare binara</a>

<ul>

<li><a href="bin1.html">Cautare binara clasica</a></li>

<li><a href="bin2.html">STL: Algoritmul "lower\_bound"</a></li>

<li><a href="bin3.html">STL: Algoritmul "upper\_bound"</a></li>

</ul>

</li>

</ul>

</div>

<div id="header"></div>

<div id="content">

<h1>Introducere</h1>

<B><I>Standard Template Library</i></b> (STL) este o biblioteza generica, ce furnizeaza solutii pentru manipularea colectiilor de date, precum si algoritmi eficienti care actioneaza asupra acestor date. Biblioteca STL este parte integranta a Bibliotecii Standard C++, imbogatind limbajul C++ cu instrumente extrem de utilizate in practica: containerele generice si algoritmii generici.<br>

Lucrarea mea de atestat va cuprinde, in general, modalitatile deosebit de practice prin care STL-ul se aplica in programare, inlocuind algoritmi clasici predati in timpul orelor de informatica si va explica modalitatile de optimizare a codurilor respective.

<h1>Informatii generale</h1>

<ul>

<li> <b><i>Biblioteca Standard C++</i></b> contine un set de clase, functii si macrodefinitii care extind nucleul limbajului. </li>

<li> Scopul <b><i>STL</b></i> este de a oferi algoritmi eficienti si colectii de date care sa fie independente de structurile de date particulare.</li>

<li>Componentele de baza ale STL sunt: clasele container, iteratorii si algoritmii.</li>

<li><B><i>Algoritmii</i></b> s-au implementat in STL ca functii generice care actioneaza asupra elementelor unei colectii de elemente (fie acesta tablou de tip C sau container C++). Eficienta algoritmilor este garantata. S-au depus eforturi deosebite pentru ca viteza de lucru a algoritmilor STL sa fie la fel de mare cu a celor mai optimi algoritmi scrisi "de mana".</li>

<li>Algoritmii constituie o componenta fundamentala a STL. SUnt functii sablon ai caror parametri sunt iterarori. Cu ajutorul iteratorilor se realizeaza independenta algoritmilor fata de tipurile de date procesate. Un acelasi algoritm poate lucra cu diferite tipuri de containere predefinite ori cu containere definite de noi insine, in particular cu tablouri, cu conditia ca specificatiile iteratorilor sa fie respectate. </li>

<li>Algoritmii ofera servicii de baza, cum ar fi: <i>copiere, modificare, mutare, cautare, sortare sau procesare numerica.</i> </li>

</ul>

<br /><br /><br />

</div>

<div id="sitebar">

<h1>Citate</h1>

<div id="menuaffi">

<a class="fade">Biblioteca Standard<br>in viziunea marilor programatori</a></div>

<img src="http://i1112.photobucket.com/albums/k481/kiimblebee/1\_zps009afadd.png"></img><br />

<hr>

<img src="http://i1112.photobucket.com/albums/k481/kiimblebee/2\_zps8fdd5fa8.png"></img>

<hr>

</div>

<div id="title">

<i></i>

</div>

</div>

<div id="footer">

<div id="footer\_holder">

<div id="footerbar\_1">

AVANTAJELE UTILIZARII STL - ATESTAT 2014 - SANDU DENISA

</div>

</div></div>

</body>

</html>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<link rel="stylesheet" type="text/css" href="style.css" />

<title>Avantajele utilizarii STL</title>

</head>

<body>

<div id="container">

<div id="headerimg6">

<ul id="menuu">

<li>

<a href="index.html">Introducere</a>

</li>

<li>

<a>Sortare</a>

<ul>

<li><a href="sortare.html">Sortarea clasica</a></li>

<li><a href="sort.html">STL Sort</a></li>

<li><a href="sortall.html">Compararea complexitatii</a></li>

</ul>

</li>

<li>

<a>Cautare binara</a>

<ul>

<li><a href="bin1.html">Cautare binara clasica</a></li>

<li><a href="bin2.html">STL: Algoritmul "lower\_bound"</a></li>

<li><a href="bin3.html">STL: Algoritmul "upper\_bound"</a></li>

</ul>

</li>

</ul>

</div>

<div id="header"></div>

<div id="content">

<h1>Algoritmi clasici de sortare</h1>

<center><font size="3"><b>SORTARE PRIN INSERARE</b></font><br>

<img src="http://upload.wikimedia.org/wikipedia/commons/0/0f/Insertion-sort-example-300px.gif"></img><br>

<b>Complexitate</b>

<ul>

<li>Caz favorabil: O(n);</li>

<li>Caz nefavorabil: O(n^2);</li>

<li>Complexitate medie: O(n^2);</li>

</ul>

<img src="http://i1112.photobucket.com/albums/k481/kiimblebee/insertion\_zps5f4bc4e5.png" border="1"></img></center>

<hr>

<center><font size="3"><b>SORTARE PRIN SELECTIE</b></font><br><br>

<table>

<tr>

<td>

<center><img src="http://upload.wikimedia.org/wikipedia/commons/9/94/Selection-Sort-Animation.gif"></img></center>

</td>

<td>

<center>

<b>Complexitate</b>

<ul>

<li>Caz favorabil: O(n);</li>

<li>Caz nefavorabil: O(n^2);</li>

<li>Complexitate medie: O(n^2);</li>

<br>

<img src="http://i1112.photobucket.com/albums/k481/kiimblebee/selection\_zpse752a41a.png" border="1" align="right"></img>

</ul>

</center>

</td></tr></table>

</center>

<br /><br /><br />

</div>

<div id="sitebar">

<h1>Citate</h1>

<div id="menuaffi">

<a class="fade">Biblioteca Standard<br>in viziunea marilor programatori</a></div>

<img src="http://i1112.photobucket.com/albums/k481/kiimblebee/1\_zps009afadd.png"></img><br />

<hr>

<img src="http://i1112.photobucket.com/albums/k481/kiimblebee/2\_zps8fdd5fa8.png"></img>

<hr>

</div>

<div id="title">

<i></i>

</div>

</div>

<div id="footer">

<div id="footer\_holder">

<div id="footerbar\_1">

Atestat informatica - 2014 - Sandu Denisa

</div>

</div></div>

</body>

</html>

<head>

<link rel="stylesheet" type="text/css" href="style.css" />

<title>Avantajele utilizarii STL</title>

</head>

<body>

<div id="container">

<div id="headerimg6">

<ul id="menuu">

<li>

<a href="index.html">Introducere</a>

</li>

<li>

<a>Sortare</a>

<ul>

<li><a href="sortare.html">Sortarea clasica</a></li>

<li><a href="sort.html">STL Sort</a></li>

<li><a href="sortall.html">Compararea complexitatii</a></li>

</ul>

</li>

<li>

<a>Cautare binara</a>

<ul>

<li><a href="bin1.html">Cautare binara clasica</a></li>

<li><a href="bin2.html">STL: Algoritmul "lower\_bound"</a></li>

<li><a href="bin3.html">STL: Algoritmul "upper\_bound"</a></li>

</ul>

</li>

</ul>

</div>

<div id="header"></div>

<div id="content">

<h1>STL Sort</h1>

<B><I>Sort</i></b> este un algoritm predefinit cu o complexitate la executie asemanatoare quicksort-ului. Daca algoritmii bine-cunoscuti de sortare prezentati <a href="sortare.php">aici</a> au avantajul de a necesita un cod de dimensiuni relativ reduse si dezavantajul unei complexitati ridicate, algoritmii eficienti asemenea quicksort-ului au o implementare dificila. <br><br>

<center>

<font size="3"><b>QUICKSORT</b></font></br>

<table>

<tr>

<td>

<center><img src="http://upload.wikimedia.org/wikipedia/commons/6/6a/Sorting\_quicksort\_anim.gif"></img></center>

</td>

<td>

<center>

<b>Complexitate</b>

<ul>

<font size="2">

<li>Caz favorabil: O(n);</li>

<li>Caz nefavorabil: O(n^2);</li>

<li>Complexitate medie: O(n log n);</li>

<br>

</font>

</ul>

</center>

</td></tr></table>

Codul necesar sortarii recursive unui vector cu n elemente utilizand metoda quicksort este urmatorul:

<br><br>

<center><img src="http://i1112.photobucket.com/albums/k481/kiimblebee/quick\_zps2338d07c.png" border="1" bordercolor="#00FF00"></img></center>

<hr color="#7a7a7a">

<center>

<font size="3"><b>STL SORT</b></font></br>

<table>

<tr>

<td><img src="http://upload.wikimedia.org/wikipedia/commons/thumb/a/af/Quicksort-diagram.svg/300px-Quicksort-diagram.svg.png"></img>

</td>

<td><font size="2">Sortarea din STL are astfel un avantaj urias: dimensiunile reduse ale codului. Utilizat pe vectori, codul presupune doua etape:

<ul><li>1. Includerea bibliotecii < algorithm >. </li>

<li>2. Scrierea algoritmului: sort(v+1, v+n+1);</li> </ul>

Acest algoritm sorteaza vectorul "v" cu "n" elemente in ordine crescatoare, intr-o complexitate O(n log n).</font></td></tr></table>

<br /><br /><br />

</div>

<div id="sitebar">

<h1>Citate</h1>

<div id="menuaffi">

<a class="fade">Biblioteca Standard<br>in viziunea marilor programatori</a></div>

<img src="http://i1112.photobucket.com/albums/k481/kiimblebee/1\_zps009afadd.png"></img><br />

<hr>

<img src="http://i1112.photobucket.com/albums/k481/kiimblebee/2\_zps8fdd5fa8.png"></img>

<hr>

</div>

<div id="title">

<i></i>

</div>

</div>

<div id="footer">

<div id="footer\_holder">

<div id="footerbar\_1">

AVANTAJELE UTILIZARII STL - ATESTAT 2014 - SANDU DENISA

</div>

</div></div>

</body>

</html>

<head>

<link rel="stylesheet" type="text/css" href="style.css" />

<title>Avantajele utilizarii STL</title>

</head>

<body>

<div id="container">

<div id="headerimg6">

<ul id="menuu">

<li>

<a href="index.html">Introducere</a>

</li>

<li>

<a>Sortare</a>

<ul>

<li><a href="sortare.html">Sortarea clasica</a></li>

<li><a href="sort.html">STL Sort</a></li>

<li><a href="sortall.html">Compararea complexitatii</a></li>

</ul>

</li>

<li>

<a>Cautare binara</a>

<ul>

<li><a href="bin1.html">Cautare binara clasica</a></li>

<li><a href="bin2.html">STL: Algoritmul "lower\_bound"</a></li>

<li><a href="bin3.html">STL: Algoritmul "upper\_bound"</a></li>

</ul>

</li>

</ul>

</div>

<div id="header"></div>

<div id="content">

<h1>Comparatii realizate intre <br>sortarea didactica si STL sort</h1>

Pe langa avantajul unui cod stabil si usor de scris, STL sort este un template scris dupa algoritmul de sortare "quicksort" - asa cum spune si numele, un algoritm rapid.

<hr>

<center><b><i><font size="4">Random</font></i></b><br>

Pentru numere alese la intamplare, sortarea prin interschimbare si STL sort se descurca astfel:

<table border="5" bordercolor=#636363>

<tr>

<td> <center><b>INTERSCHIMBARE</b></center> </td>

<td> <center><b>STL Sort</b></center> </td>

</tr>

<tr><td>

<img src="insertion-sort.gif"></img>

</td>

<td>

<img src="quick-sort.gif"></img></td></tr>

</table></center>

<hr>

<center><b><i><font size="4">Aproape sortate</font></i></b><br>

Pentru numere aproape sortate deja, sortarea prin interschimbare si STL sort se descurca astfel:

<table border="5" bordercolor=#636363>

<tr>

<td> <center><b>INTERSCHIMBARE</b></center> </td>

<td> <center><b>STL Sort</b></center> </td>

</tr>

<tr><td>

<img src="insertion-sort2.gif"></img>

</td>

<td>

<img src="quick-sort2.gif"></img></td></tr>

</table></center>

<hr>

<center><b><i><font size="4">Inversate</font></i></b><br>

Pentru numere aflate in ordine inversa, sortarea prin interschimbare si STL sort se descurca astfel:

<table border="5" bordercolor=#636363>

<tr>

<td> <center><b>INTERSCHIMBARE</b></center> </td>

<td> <center><b>STL Sort</b></center> </td>

</tr>

<tr><td>

<img src="insertion-sort3.gif"></img>

</td>

<td>

<img src="quick-sort3.gif"></img></td></tr>

</table></center>

<hr>

<center><b><i><font size="4">Cateva numere nesortate</font></i></b><br>

Cand exista doar cateva numere care nu sunt sortate si apar unice, sortarea prin interschimbare si STL sort se descurca astfel:

<table border="5" bordercolor=#636363>

<tr>

<td> <center><b>INTERSCHIMBARE</b></center> </td>

<td> <center><b>STL Sort</b></center> </td>

</tr>

<tr><td>

<img src="insertion-sort4.gif"></img>

</td>

<td>

<img src="quick-sort4.gif"></img></td></tr>

</table></center>

<br /><br /><br />

</div>

<div id="sitebar">

<h1>Citate</h1>

<div id="menuaffi">

<a class="fade">Biblioteca Standard<br>in viziunea marilor programatori</a></div>

<img src="http://i1112.photobucket.com/albums/k481/kiimblebee/1\_zps009afadd.png"></img><br />

<hr>

<img src="http://i1112.photobucket.com/albums/k481/kiimblebee/2\_zps8fdd5fa8.png"></img>

<hr>

</div>

<div id="title">

<i></i>

</div>

</div>

<div id="footer">

<div id="footer\_holder">

<div id="footerbar\_1">

AVANTAJELE UTILIZARII STL - ATESTAT 2014 - SANDU DENISA

</div>

</div></div>

</body>

</html>

<head>

<link rel="stylesheet" type="text/css" href="style.css" />

<title>Avantajele utilizarii STL</title>

</head>

<body>

<div id="container">

<div id="headerimg6">

<ul id="menuu">

<li>

<a href="index.html">Introducere</a>

</li>

<li>

<a>Sortare</a>

<ul>

<li><a href="sortare.html">Sortarea clasica</a></li>

<li><a href="sort.html">STL Sort</a></li>

<li><a href="sortall.html">Compararea complexitatii</a></li>

</ul>

</li>

<li>

<a>Cautare binara</a>

<ul>

<li><a href="bin1.html">Cautare binara clasica</a></li>

<li><a href="bin2.html">STL: Algoritmul "lower\_bound"</a></li>

<li><a href="bin3.html">STL: Algoritmul "upper\_bound"</a></li>

</ul>

</li>

</ul>

</div>

<div id="header"></div>

<div id="content">

<h1>Cautarea binara clasica</h1>

Algoritmul de cautare binara este un algoritm de cautare folosit pentru a gasi un element intr-o lista ordonata (tablou unidimensional/vector). Algoritmul functioneaza pe baza tehnicii divide et impera. Valoarea cautata este comparata cu cea a elementului din mijlocul listei. Daca e egala cu cea a acelui element, algoritmul se termina. Daca e mai mare decat acea valoare, algoritmul se reia, de la mijlocul listei pana la sfarsit, iar daca e mai mica, algoritmul se reia pentru elementele de la inceputul listei pana la mijloc. Intrucat la fiecare pas cardinalul multimii de elemente in care se efectueaza cautarea se injumatateste, algoritmul are complexitate logaritmica.

<br>

<center>

<img src="http://flylib.com/books/2/300/1/html/2/images/08fig02.jpg"> </img>

</center>

<br>

In imaginea de mai sus este descris procedeul prin care algoritmul lucreaza pentru a gasi intr-un vector sortat elementul de valoare 76. <br>

O functie recursiva care realizeaza cautarea binara a unui element intr-un vector sortat este realizata in codul urmator:<br><br>

<center>

<img src="http://i1112.photobucket.com/albums/k481/kiimblebee/bin\_zpsc30e8513.png" border="1"></img><br><br>

<b>Complexitate</b>

<ul>

<li>Caz favorabil: O(1);</li>

<li>Caz nefavorabil: O(log n);</li>

<li>Complexitate medie: O(log n);</li>

</ul>

</center>

<br /><br /><br />

</div>

<div id="sitebar">

<h1>Citate</h1>

<div id="menuaffi">

<a class="fade">Biblioteca Standard<br>in viziunea marilor programatori</a></div>

<img src="http://i1112.photobucket.com/albums/k481/kiimblebee/1\_zps009afadd.png"></img><br />

<hr>

<img src="http://i1112.photobucket.com/albums/k481/kiimblebee/2\_zps8fdd5fa8.png"></img>

<hr>

</div>

<div id="title">

<i></i>

</div>

</div>

<div id="footer">

<div id="footer\_holder">

<div id="footerbar\_1">

AVANTAJELE UTILIZARII STL - ATESTAT 2014 - SANDU DENISA

</div>

</div></div>

</body>

</html>

<head>

<link rel="stylesheet" type="text/css" href="style.css" />

<title>Avantajele utilizarii STL</title>

</head>

<body>

<div id="container">

<div id="headerimg6">

<ul id="menuu">

<li>

<a href="index.html">Introducere</a>

</li>

<li>

<a>Sortare</a>

<ul>

<li><a href="sortare.html">Sortarea clasica</a></li>

<li><a href="sort.html">STL Sort</a></li>

<li><a href="sortall.html">Compararea complexitatii</a></li>

</ul>

</li>

<li>

<a>Cautare binara</a>

<ul>

<li><a href="bin1.html">Cautare binara clasica</a></li>

<li><a href="bin2.html">STL: Algoritmul "lower\_bound"</a></li>

<li><a href="bin3.html">STL: Algoritmul "upper\_bound"</a></li>

</ul>

</li>

</ul>

</div>

<div id="header"></div>

<div id="content">

<h1>STL: Algoritmul "lower\_bound"</h1>

<b>Utilizare pe vectori</b>

<br>

Secventa urmatoare utilizeaza lower\_bound pentru a determina pozitiile pe care se pot insera valorile 5 si 4 in sirul a, cu pastrarea relatiei de ordine.

<ul>

<font size="2">int a[] = { 2, 3, 5, 5, 5, 8, 10 };<br>

int \* p = lower\_bound (a, a+7, 5};<br>

cout << "pozitia " << p-a << " " << "valoarea actuala " << \*p;<br>

p = lower\_bound (a, a+7, 4);<br>

cout<< " pozitia " << p-a << " " << "valoarea actuala " << \*p;

</font></ul>

<b>Timpul de executie</b> al algoritmului, in functie de lungimea vectorului, variaza astfel:<br>

<img src="http://www.pvk.ca/images/2012-07-30-binary-search-is-a-pathological-case-for-caches/bsearch-first-cycles.png"></img>

<br>

</div>

<div id="sitebar">

<h1>Citate</h1>

<div id="menuaffi">

<a class="fade">Biblioteca Standard<br>in viziunea marilor programatori</a></div>

<img src="http://i1112.photobucket.com/albums/k481/kiimblebee/1\_zps009afadd.png"></img><br />

<hr>

<img src="http://i1112.photobucket.com/albums/k481/kiimblebee/2\_zps8fdd5fa8.png"></img>

<hr>

</div>

<div id="title">

<i></i>

</div>

</div>

<div id="footer">

<div id="footer\_holder">

<div id="footerbar\_1">

AVANTAJELE UTILIZARII STL - ATESTAT 2014 - SANDU DENISA

</div>

</div></div>

</body>

</html>

<head>

<link rel="stylesheet" type="text/css" href="style.css" />

<title>Avantajele utilizarii STL</title>

</head>

<body>

<div id="container">

<div id="headerimg6">

<ul id="menuu">

<li>

<a href="index.html">Introducere</a>

</li>

<li>

<a>Sortare</a>

<ul>

<li><a href="sortare.html">Sortarea clasica</a></li>

<li><a href="sort.html">STL Sort</a></li>

<li><a href="sortall.html">Compararea complexitatii</a></li>

</ul>

</li>

<li>

<a>Cautare binara</a>

<ul>

<li><a href="bin1.html">Cautare binara clasica</a></li>

<li><a href="bin2.html">STL: Algoritmul "lower\_bound"</a></li>

<li><a href="bin3.html">STL: Algoritmul "upper\_bound"</a></li>

</ul>

</li>

</ul>

</div>

<div id="header"></div>

<div id="content">

<h1>STL: Algoritmul "upper\_bound"</h1>

<b>Utilizare pe vectori</b>

<br>

Secventa urmatoare utilizeaza upper\_bound pentru a determina pozitiile pe care se pot insera valorile 5 si 6 in sirul a, cu pastrarea relatiei de ordine.

<ul>

<font size="2">int a[] = { 2, 3, 5, 5, 5, 8, 10 };<br>

int \* p = upper\_bound (a, a+7, 5};<br>

cout << "pozitia " << p-a << " " << "valoarea actuala " << \*p;<br>

p = upper\_bound (a, a+7, 6);<br>

cout<< " pozitia " << p-a << " " << "valoarea actuala " << \*p;

</font></ul>

<b>Executia</b> algoritmului pe caz general poate fi descrisa astfel:<br><br>

<center><img src="http://darcy.rsgc.on.ca/ACES/ICS3U/images/BinarySearchAnimation.gif"></img></center>

<br>

</div>

<div id="sitebar">

<h1>Citate</h1>

<div id="menuaffi">

<a class="fade">Biblioteca Standard<br>in viziunea marilor programatori</a></div>

<img src="http://i1112.photobucket.com/albums/k481/kiimblebee/1\_zps009afadd.png"></img><br />

<hr>

<img src="http://i1112.photobucket.com/albums/k481/kiimblebee/2\_zps8fdd5fa8.png"></img>

<hr>

</div>

<div id="title">

<i></i>

</div>

</div>

<div id="footer">

<div id="footer\_holder">

<div id="footerbar\_1">

AVANTAJELE UTILIZARII STL - ATESTAT 2014 - SANDU DENISA

</div>

</div></div>

</body>

</html>

**CSS**

body {

color : #343434;

font-family : "Trebuchet MS",Arial,verdana;

font-size : 0.9em;

text-align : left;

margin : 0;

padding : 0 0 0;

background-color: #636363;

background-image: url("patt7.jpg");

}

a {

color : #BBBBBB;

text-decoration : none;

}

a:visited {

color : #BBBBBB;

text-decoration : none;

}

a:hover {

color : #99AC75;

text-decoration : none;

}

#container {

margin : auto;

margin-top : 0;

width : 900px;

background-repeat : repeat-x;

}

b {

color: #788B54;

}

#menuu {

width : 400px;

margin : 0px;

padding : 10px 0 0 0;

list-style : none;

position: absolute;

left: 35px;

background-color: #587559 ;

box-shadow : 0 2px 2px #C0C0C0;

border-radius: 10px;

border-left: 7px solid #ffffff;

border-right: 7px solid #ffffff;

}

#menuu li {

float : left;

padding : 0 0 10px 0;

position : relative;

}

#menuu a {

float : left;

height : 20px;

padding : 0 25px;

text-transform : uppercase;

font : bold 12px/20px Arial;

text-decoration : none;

color : #fafafa;

}

#menuu li:hover > a {

color : #fafafa;

}

#menuu li:hover > ul {

display : block;

}

#menuu ul {

list-style : none;

margin : 0;

padding : 0;

display : none;

position : absolute;

top : 28px;

left : 0;

z-index : 99999;

background-color:#191806;

opacity:0.85;

border-radius: 10px;

}

#menuu ul li {

float : none;

margin : 0;

padding : 0;

display : block;

box-shadow : 0 1px 0, 0 2px 0 #C1C1C1;

}

#menuu ul li:last-child {

box-shadow : none;

}

#menuu ul a {

padding : 10px;

height : auto;

line-height : 1;

display : block;

white-space : nowrap;

float : none;

text-transform : none;

}

\*:first-child + html #menuu ul a {

height : 10px;

width : 150px;

}

#menuu ul a:hover {

background : url("http://i1112.photobucket.com/albums/k481/kiimblebee/mb1\_zpsf6f7a206.jpg");

}

#menuu ul li:first-child a {

}

#menuu ul li:first-child a:after {

content : '';

position : absolute;

left : 30px;

top : -8px;

width : 0;

height : 0;

border-left : 5px solid transparent;

border-right : 5px solid transparent;

border-bottom : 8px solid #191806;

}

#menuu ul li:first-child a:hover:after {

border-bottom-color : #B3C68F;

}

#menuu ul li:last-child a {

border-radius: 0px 0px 10px 10px;

}

#menuu:after {

visibility : hidden;

display : block;

font-size : 0;

content : " ";

clear : both;

height : 0;

}

#menuaffi a, #menuaffi a:visited {

text-decoration : none;

text-align : center;

color : #ffffff;

background-color: #303f25;

display : block;

width : 250px;

height : 35px;

padding : 0.7em;

margin : 0.5em auto;

}

#menuaffi a:hover {

text-decoration : none;

text-align : center;

background-color: #303f25;

color : #ffffff;

display : block;

width : 250px;

height : 35px;

padding : 0.7em;

margin : 0.5em auto;

}

#header {

width : 900px;

height : 355px;

background : url("http://i1112.photobucket.com/albums/k481/kiimblebee/header3\_zps303e8190.png") no-repeat;

margin-top : 60px;

position : relative;

border-top : 9px solid #ffffff;

border-radius: 20px 20px 0px 0px;

}

#headerimg6 {

list-style : none;

margin-left: 200px;

position : absolute;

height : 165px;

width : 430px;

float : left;

top : 10px;

text-align : left;

font-size : 80%;

color : #b9b9b9;

background : none;

}

#content {

float : left;

padding : 5px 10px 10px 12px;

width : 583px;

color : #808080;

background: #ffffff;

text-align : left;

}

#content h1, #content .commentformtitle {

color: #474745;

font-family: "Comic Sans";

text-align: center;

font-size: 15pt;

letter-spacing:2px;

border-bottom: 1px solid #aaaaaa;

padding:5px;

margin:20px 2px 7px 1px;

}

#sitebar {

float : right;

padding : 6px 13px 3px 10px;

width : 272px;

color : #D6D5CF;

font-family : "Trebuchet MS",Arial,verdana;

font-size : 0.9em;

text-align : center;

border-bottom : 9px solid #ffffff;

border-radius: 0px 0px 20px 0px;

background: #25311c;

opacity:0.9;

}

#sitebar h1 {

color: #ffffff;

font-family:arial;

font-size:10pt;

text-align: center;

text-transform:uppercase;

letter-spacing:6px;

padding:10px;

margin:20px 2px 7px 1px;

}

#title {

position:absolute;

width: 270px;

float : left;

top: 280px;

opacity:0.5;

font-family: "Comic Sans";

font-size: 35pt;

text-align: right;

color: #ffffff;

padding: 10px 0px 0px 270px;

letter-spacing: -4px;

}

#footer {

clear : both;

color : #ffffff;

font-weight : bold;

font-family : "Trebuchet MS",Arial,verdana;

font-size : 12px;

text-align : center;

margin-left : auto;

margin-right : auto;

width : 100%;

height : 40px;

text-align : left;

background : #1e2a16;

opacity:0.7;

}

#footer a, #footer a:visited {

font-family : "Trebuchet MS",Arial,verdana;

font-size : 12px;

}

.footer\_holder {

margin : auto;

padding : 0;

width : 1000px;

height : 50px;

}

#footerbar\_1 {

margin-left : auto;

margin-right : auto;

padding-top : 10px;

width : 600px;

font-weight : bold;

font-family : "Trebuchet MS",Arial,verdana;

font-size : 12px;

text-align : left;

background : none;

}

Textarea {

text-align : left;

color : #353535;

font-family : arial;

font-size : 13px;

padding : 5px;

width : 340px;

border : #dfc6b3 solid 1px;

background : #ecddd2;

}

Input {

margin : 2px;

padding : 2px;

border : #dfc6b3 solid 1px;

background : #ecddd2;

font-size : 80%;

border-radius : 5px;

}

blockquote {

background-color: #EFEFEF;

padding: 10px;

border-radius: 10px;

border: 1px solid #D3D3D3;

quotes: "\201C" "\201D";

}

blockquote:before { content: open-quote; font-weight: bold; }

blockquote:after { content: close-quote; font-weight: bold; }

**Concluzie**

Biblioteca standard C++ are trei mari componente: biblioteca de functii C, cea de stream-uri de intrare / iesire si Standard Template Library (STL). STL se bazeaza pe trei concepte centrale: containeri, iteratori si algoritmi. In schimb se utilizeaza din plin polimorfismul parametric. In C++ numele acestuia este "template".

In limbajul de baza avem la dispozitie tablourile pentru reprezentarea unei secvente indexate de elemente. Dimensiunea unui tablou trebuie fie sa fie cunoscuta la compilare, fie sa fie gestionata explicit de programator prin alocare dinamica. Principalul avantaj al STL este ca ne scapa de aceasta grija. In afara de adresarea indexata obisnuita, secventele STL au si operatia push\_back: adauga un element la sfarsit (si, evident, creste dimensiunea cu 1). Exista si operatia inversa pop\_back: elimina ultimul element. Cele trei secvente STL care suporta aceste operatii sunt vector, deque si list.

Astfel, utilizarea limbajului din Standard Template Library este in mod evident usoara si deosebit de utila.

**Bibliografie**

|  |  |
| --- | --- |
|  | * ***Constantin Gălățan***, C++ Intorucere in Standard Template Library, Editura ALL, 2008  *Emanuela Cerchez*, Programarea in limbajul C/C++ pentru liceu. Volumul al IV-lea: Programare orientata pe obiecte si programare generica cu STL, Editura Polirom, 2013 |