ობიექტზე ორიენტირებული დაპროგრამება E335

ილიას სახელმწიფო უნივერსიტეტი

ერეკლე მაღრაძე

ლექცია 2

თემები

- მასივები
- ვექტორები
- სტრიქონები

მასივები

ტიპი	მასივის სახელი	
int	a	
	a[0]	10
	a[1]	20
	a[2]	-3
	a[3]	5
	a[4]	0
	a[5]	456

ელემენტის ინდექსი

- მასივი array წარმოადგენს ერთი და იგივე ტიპის მეხსიერების უჯრედების ერთობლიობას. ამ ერთობლიობას აქვს ერთი სახელი მაგ. a და თითოეულ ელემენტზე წვდომა ხდება ელემენტის ინდექსის მიხედვით.
- ელემენტების ინდექსაცია იწყება 0-დან.
- ელემენტის ბოლო ინდექსი 1-ით ნაკლებია მასივის ზომაზე.

მასივები

- 1 განზომილებიანი მასივები a[n]
- 2 და მეტ განზომილებიანი მასივები a[n][m][k]
- მასივების ზომები n,m,k წინასწარ უნდა იყოს განსაზღვრული

მასივების ინიციალიზაცია და შევსება I

```cpp
#include <iostream>
#include <iomanip>
using namespace std;

int main(){
        int n[10];
        for (int i=0;i<10; i++)
                n[i]=i+10;
        cout <<"Element"<<setw(13)<<"Value"<<endl;
        for (int j=0;j<10;j++)
                cout<<setw(7)<<j<<setw(13)<<n[j]<<endl;
        return 0;
}
```

```
/s/C/CPP > ./a.out
Element        Value
      0           10
      1           11
      2           12
      3           13
      4           14
      5           15
      6           16
      7           17
      8           18
      9           19
/s/C/CPP >
```

მასივის ინიციალიზაცია II

```
1  #include <iostream>
2  #include <iomanip>
3  using namespace std;
4
5  int main()
6  {
7     // n მასივისთვის ინიციალიზაციის სიის გამოყენება
8     int n[ 10 ] = { 32, 27, 64, 18, 95, 14, 90, 70, 60, 37 };
9
10    cout << "Element" << setw( 13 ) << "Value" << endl;
11
12    // თითოეული ელემენტის გამოტანა ეკრანზე
13    for ( int i = 0; i < 10; i++ )
14       cout << setw( 7 ) << i << setw( 13 ) << n[ i ] << endl;
15
16    return 0;
17 }
```

```
//მასივის აღწერის კიდევ ერთი მეთოდი - მეხსიერების 0-ებით შევსება
int n[10]={0};

//ზომის წინასწარ არ მქონე მასივის ელემენტებით ინიციალიზაცია
int n[]={1, 2, 3, 4, 5};
```

მასივების ინიციალიზაცია III

```
1   #include <iostream>
2   #include <iomanip>
3   using namespace std;
4
5   int main()
6   {
7      // constant variable can be used to specify array size
8      const int arraySize = 10; // must initialize in declaration
9
10     int s[ arraySize ]; // array s has 10 elements
11
12     for ( int i = 0; i < arraySize; i++ ) // set the values
13        s[ i ] = 2 + 2 * i;
14
15     cout << "Element" << setw( 13 ) << "Value" << endl;
16
17     // output contents of array s in tabular format
18     for ( int j = 0; j < arraySize; j++ )
19        cout << setw( 7 ) << j << setw( 13 ) << s[ j ] << endl;
20
21  }
22
```

მასივების ინიციალიზაცია IV

```
// Initializing multidimensional arrays.
#include <iostream>
using namespace std;

void printArray( const int [][ 3 ] ); // prototype
const int rows = 2;
const int columns = 3;

int main()
{
   int array1[ rows ][ columns ] = { { 1, 2, 3 }, { 4, 5, 6 } };
   int array2[ rows ][ columns ] = { 1, 2, 3, 4, 5 };
   int array3[ rows ][ columns ] = { { 1, 2 }, { 4 } };

   cout << "Values in array1 by row are:" << endl;
   printArray( array1 );

   cout << "\nValues in array2 by row are:" << endl;
   printArray( array2 );

   cout << "\nValues in array3 by row are:" << endl;
   printArray( array3 );

}

// output array with two rows and three columns
void printArray( const int a[][ columns ] )
{
   // loop through array's rows
   for ( int i = 0; i < rows; i++ )
   {
      // loop through columns of current row
      for ( int j = 0; j < columns; j++ )
         cout << a[ i ][ j ] << ' ';

      cout << endl; // start new line of output
   }
}
```

ვექტორები

- C++ სტანდრატული ბიბლიოთეკის მნიშვნელოვანი კლასია vector
- Vector თანმიმდევრობით განლაგებული ერთი და იგივე ტიპის მონაცემების ერთობლიობა
- ვექტორის ელემენტები შეიძლება იყოს ობიექტები, სტრიქონები, მთელი რიცხვები …
- განაცხადი vector <int> V;
- ვექტორის ზომას გავიგებთ size() მეთოდის საშუალებით
  cout<<V.size()<<endl;
  0

ვექტორები II

- ვექტორში ელემენტის დამატება ხდება ბოლოში
- ელემენტის დამატების მეთოდია push_back()
  მაგ. V.push_back(100); ვექტორის ზომა გახდება 1 - შევამოწმოთ
- Vector <double> Vec(2);
  შექმნის ცარიელ, 2 ელემენტიან ვექტორს
- ვექტორის ელემენტებზე მიმართვისას ვიყენებთ ინდექსს ან at მეთოდს.

სტრიქონები

- სტრიქონისთვის მეხსიერების გამოყოფა ხდება დინამიურად, გამოცხადებისთანავე ხდება მაგ. 15 ბაიტის გამოყოფა, რაც შეიძლება იყოს სხვადასხვა – კომპილატორიდან გამომდინარე
- მეხსიერების დინამიური გამოყოფის ხარჯზე, პროგრამის შესრულების პროცესში სტრიქონის სიგრძე შეიძლება, როგორც შემცირდეს ასევე გაიზარდოს
- სტრიქონებთან სამუშაოდ გვჭირდება ბიბლიოთეკა string
- სტრიქონის ელემენტებზე წვდომა ხორციელდება ინდექსების მიხედვით

```
#include <iostream>
#include <string>
using namespace std;

int main()
        {
                string the_string;
                the_string="პირველი სტრიქონი";
                cout<<the_string.size()<<endl;
                the_string+=" მივადგათ მეორე სტრიქონი";
                cout<<the_string.size()<<endl;
                the_string="შევცვალოთ სტრიქონის ზომა";
                cout<<the_string.size()<<endl;
                return 0;
        }
~
~
```

```
/s/C/CPP > ./a.out
46
112
68
/s/C/CPP >
```

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4  int main(){
5      string Name, lastName;
6      cout << "\nEnter your name and lastname\n";
7      cin >> Name >> lastName;
8      string a = "My name";
9      string b(" is ");
10     string fullName;
11     fullName = a +  b;
12     fullName += Name + " " + lastName;
13     cout << fullName + "!" << endl;
14     return 0;
15 }
```

```
1  //სტრიქონის კლავიატურიდან შეტანის ფუნქცია getline
2
3  #include <iostream>
4  #include <string>
5  using namespace std;
6
7  int main()
8  {
9      cout << "Enter text\n";
10     string words;
11     getline(cin, words);
12
13     cout << "\nEntered text is\n";
14     cout << words << endl;
15     return 0;
16 }
```

```
1  //append ფუნქციის გამოყენება
2
3  #include <iostream>
4  #include <string>
5  using namespace std;
6
7  int main ()
8  {
9      string str = "Nobody is perfect";
10     string s   = "";
11
12     s.append(str,0,6);
13     cout << "s is : " << s << endl;
14
15     s.append(str.begin()+6, str.end());
16     cout << "s is : " << s << endl;
17
18     s.append(3,'!');
19     cout << "s is : " << s << endl;
20     return 0;
21 }
```

```
1  //ფუნქცია  find.
2
3  #include <iostream>
4  #include <string>
5  using namespace std;
6
7  int main ()
8  {
9      string  str("C++ is best language");
10     int pos1, pos2;
11
12     pos1 = str.find ("best");
13     cout << "Word best is found on position " << pos1+1
14          << endl;
15
16     pos2 = str.find ("best",pos1+1);
17     cout << "Word best is found on position " << pos2+1
18          << endl;
19
20     pos1 = str.find('g');
21     cout << "First character 'g' found on position "
22          << pos1 << endl;
23     return 0;
24 }
25
```

```
1  //აღგორითმი  reverse.
2
3  #include <iostream>
4  #include <string>
5  using namespace std;
6
7  int main ()
8  {
9      string str = "Programming Language";
10     reverse(str.begin(), str.end());
11     cout << str << endl;
12     return 0;
13 }
14
```