

## vector

Вектор `vector` в STL определен как динамический массив с доступом к его элементам по индексу.

```
vector<int> a;  
vector<double> x(5);  
vector<char> c(5, '*');  
vector<int> b(a); // b = a
```

Для любого объекта, который будет храниться в векторе, должен быть определен конструктор по умолчанию. Кроме того, для объекта должны быть определены операторы `<` и `==`.

Для класса вектор определены следующие операторы сравнения:  
`==`, `<`, `<=`, `!=`, `>`, `>=`.

Кроме этого, для класса `vector` определяется оператор индекса `[]`.

- Новые элементы могут включаться с помощью функций `insert()`, `push_back()`, `resize()`, `assign()`.
- Существующие элементы могут удаляться с помощью функций `erase()`, `pop_back()`, `resize()`, `clear()`
- Доступ к отдельным элементам осуществляется с помощью итераторов `begin()`, `end()`, `rbegin()`, `rend()`.
- Манипулирование контейнером, сортировка, поиск в нем и тому подобное возможно с помощью глобальных функций файла - заголовка `<algorithm.h>`.

```
#include <iostream.h>  
#include <vector.h>
```

```
using namespace std;
```

```
int main(void)  
{  
    vector<int> v;  
    for(int i = 0; i < 10; i++)  
    {  
        v.push_back(i);  
    }  
    cout << "size = " << v.size() << "\n";  
    for (int i = 0; i < 10; i++)  
    {  
        cout << v[i] << " ";  
    }  
    cout << endl;  
    for (int i = 0; i < 10; i++)  
    {
```

```

    v[i] = v[i] + v[i];
}
for (int i = 0; i < v.size(); i++)
{
    cout << v[i] << " ";
}
cout << endl;
return 0;
}

```

Доступ к вектору через итератор

```

#include <iostream.h>
#include <vector.h>

using namespace std;

int main(void)
{
    vector<int> v;
    for (int i = 0; i < 10; i++)
    {
        v.push_back(i);
    }
    cout << "size = " << v.size() << "\n";
    vector<int>::iterator p = v.begin();
    while (p != v.end())
    {
        cout << *p << " ";
        p++;
    }
    return 0;
}

```

Вставка и удаление элементов

```

#include <iostream.h>
#include <vector.h>

using namespace std;

int main(void)
{
    vector<int> v(5, 1);
    // ВЫВОД
    for (int i = 0; i < 5; i++)
    {

```

```
    cout << v[i] << " ";
}
cout << endl;
vector<int>::iterator p = v.begin();
p += 2;
// вставить 10 элементов со значением 9
v.insert(p, 10, 9);
//ВЫВОД
p = v.begin();
while (p != v.end())
{
    cout << *p << " ";
    p++;
}
// удалить вставленные элементы
p = v.begin();
p += 2;
v.erase (p, p + 10);
// ВЫВОД
p = v.begin();
while (p != v.end())
{
    cout << *p << " ";
    p++;
}
return 0;
}
```