

Итератор

Синтаксис: `data_structure <type>:: iterator name;`

Итераторът е помощна структура данни, използвана за обхождане на линейни структури от STL (vector, map, list). Дефинира се като вида структура, нейния тип и ключовата дума iterator и се използва директно към обекти, съответстващи на дефиницията.

Пример:

```
vector<int> v;  
vector<int>:: iterator it = v.begin();  
while(it!=v.end())it++;
```

Основни функции, методи и оператори:

<code>*iterator</code>	Връща стойността на полето, сочено от итератора.
<code>++</code> <code>--</code>	Оператори за преместване с една позиция напред/назад.
<code>==</code> <code>!=</code>	Оператори за сравнение
<code>=</code>	Оператор за присвояване
<code>advance(iterator, x)</code>	Измества итератора с <code>x</code> позиции напред.
<code>next(iterator, x)</code>	Връща нов итератор, сочещ с <code>x</code> позиции напред.
<code>prev(iterator, x)</code>	Връща нов итератор, сочещ с <code>x</code> позиции назад.
<code>distance(iterator1, iterator2)</code>	Връща цяло число – броя позиции между <code>iterator1</code> и <code>iterator2</code>

В зависимост от структурата, за която са пригодени, итераторите могат да бъдат няколко вида:

Вид итератор	Структури, за които се използва	Поддържа следните операции:																
<u>Random-access</u> предоставят директен достъп до дадена позиция	std::vector	<table><tr><td>It + n</td><td>It == it1</td></tr><tr><td>it += n</td><td>it != it1</td></tr><tr><td>it++</td><td>it < it1</td></tr><tr><td>++it</td><td>it <= it1</td></tr><tr><td>it – n</td><td>it > it1</td></tr><tr><td>it -=n</td><td>it >= it1</td></tr><tr><td>it--</td><td></td></tr><tr><td>--it</td><td></td></tr></table>	It + n	It == it1	it += n	it != it1	it++	it < it1	++it	it <= it1	it – n	it > it1	it -=n	it >= it1	it--		--it	
It + n	It == it1																	
it += n	it != it1																	
it++	it < it1																	
++it	it <= it1																	
it – n	it > it1																	
it -=n	it >= it1																	
it--																		
--it																		
<u>Bidirectional</u> могат да се предвижват и в двете	std::list	<table><tr><td>It++</td><td>It == it1</td></tr><tr><td>++it</td><td>it != it1</td></tr></table>	It++	It == it1	++it	it != it1												
It++	It == it1																	
++it	it != it1																	

посоки, имат предефиниран next() и prev()	stl::map stl::set	it-- --it	
<u>Forward</u> могат да се предвижват само в една посока, имат предефиниран next()	stl::forward_list	It++ It == it1 ++it it != it1	