
C++

STL

创客学院 荣荣老师

1 / 容器

2 / 迭代器

3 / 算法

C++写一条链表

容器

I 顺序容器:

vector

list

deque

适配器

stack

queue

priority_queue

关联容器:

map

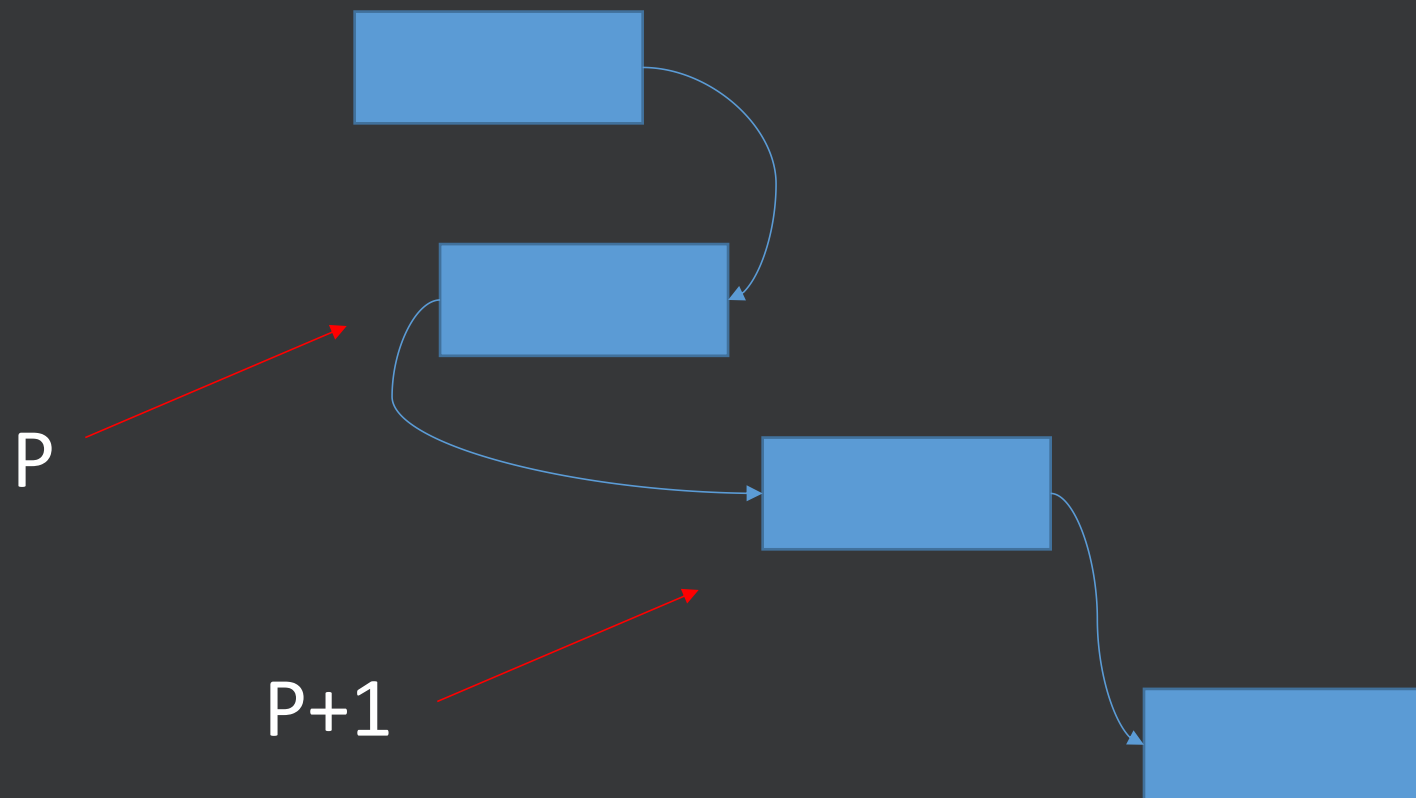
set //tree

multimap

multiset

迭代器

■ iterator:
“神奇的指针”



顺序容器

■ vector: 顺序表

insert();

push_back();

erase();

pop_back();

empty();

begin();

end();

.....

顺序容器

■ list: 链表

insert();

push_back();

erase();

pop_back();

empty();

front();

back();

sort();

顺序容器

- deque: 双端
insert();
push_back();
erase();
pop_back();
empty();
Push_front();

顺序容器

stack:

适配器，它可以将任意类型的序列容器转换为一个堆栈，一般使用`deque`作为支持的序列容器。
元素只能后进先出（LIFO）

`push();`

`top();`

`pop();`注意，出栈操作只是删除栈顶元素，并不返回该元素

顺序容器

queue:

适配器，它可以将任意类型的序列容器转换为一个队列，一般使用deque作为支持的序列容器。
元素只能先进先出（FIFO）

push();

front()/back();

pop();注意，出栈操作只是删除栈顶元素，并不返回该元素

关联容器

■ map:键值对 (key/value) 容器

```
map<string, double> stu;
```

```
insert( make_pair<string, double>( "john" ,95.5) );
```

```
stu[ "keiven" ] = 80.0;
```

```
cout<< "john   : " <<stu[ "john" ] <<endl;
```

```
cout<< "keiven : " <<stu[ "keiven" ] <<endl;
```

关联容器

■ set:

```
set<int> a;
```

```
a.insert(1);
```

```
a.insert(3);
```

```
a.insert(5);
```

```
if(a.end() != a.find(3) )
```

```
    cout<<"have 3"<<endl;
```

```
if(a.end() != a.find(30) )
```

```
    cout<<"have 30"<<endl;
```

算法 `#include<algorithm>`

`sort(b,e)`

`sort(b,e,回掉函数)`

`unique(b,e);` //使用该算法前,要先对元素进行排序

`find_if(b,e,谓词);`

`count_if(b,e,谓词);`

`for_each(b,e,回掉函数);`

扫一扫，获取更多信息



THANK YOU