C++标准库实现很多的容器，包括二叉树和散列表

### 向量容器

* 1. 向量vector

std::vector<type> vectorName;

#include <iostream>

#include <string>

#include <vector>

int main()

{

std::vector<std::string> names;

names.push\_back("Darbuly");

names.push\_back("DBL");

for(int i=0;i<names.size();i++)

{

std::cout<<names[i]<<"\n";

}

return 0;

}

### 迭代器

for(int i=0;i<names.size();i++)

{

std::cout<<names[i]<<"\n";

}

问题：我们仍把它视为一个C++数组来对待

1. iterator各种迭代器
2. 操作符：\* ++ == ！= =
3. 每一种容器都必须提供自己的迭代器，接口相同，型号不同。

#include <iostream>

#include <string>

#include <vector>

int main()

{

std::vector<std::string> names;

names.push\_back("Darbuly");

names.push\_back("DBL");

std::vector<std::string>::iterator iter = names.begin();

while(iter!=names.end())

{

std::cout<< \*iter<<"\n";

++iter;

}

return 0;

}

1. 算法
   1. 排序算法algorithm

#include <iostream>

#include <string>

#include <vector>

#include <algorithm>

int main()

{

std::vector<std::string> names;

names.push\_back("Darbuly");

names.push\_back("Lola");

names.push\_back("Dingding");

names.push\_back("Joy");

names.push\_back("Micheal");

names.push\_back("Lucy");

names.push\_back("Lilei");

std::sort(names.begin(),names.end());

std::vector<std::string>::iterator iter = names.begin();

while(iter!=names.end())

{

std::cout<< \*iter<<"\n";

++iter;

}

return 0;

}