# stl中pair容器的用法

****pair就是一个结构体，但是比结构体更加得灵活****

****template <class T1, class T2> struct pair //模板， T1, T2,可以是不同的类型****

****构造函数****

#include<iostream>

using namespace std;

#include<string>

//#include <utility>

int main(){

pair<int, string> p1; //default constructor

pair<string, double>p2("zhouyu", 100); // overroad constructor

pair<string, double>p3=(p2);

p2.first = "nobody"; p2.second=20;

cout<<p2.first<<" "<<p2.second<<endl;

cout<<p3.first<<" "<<p3.second<<endl;

return 0;

}

****重载运算符“=”，和makepair的用法****

#include<iostream>

using namespace std;

#include<string>

//#include <utility>

int main(){

pair<string, double> p1 = make\_pair("tianyu", 100);

pair<string, double> p2;

p2=p1; //overoad "="

cout<<p2.first<<" "<<p2.second<<endl;

return 0;

}

pair类的比较函数：

****pair<class first,class second> p;****

****说明：pari的比较是按照字典序比较的，还有就是先比较first,frist的值大的时候，pair就大，如果first相等，再比较second,second大的就pair打，如果first,second都一样，等于就成立。****

****可以验证一下，下面程序输出的结果****

*#include <iostream>*

*using namespace std;*

*int main (){*

*pair<int,char> A (10,'z');*

*pair<int,char> B (90,'a');*

*if (A==B) cout << "foo and bar are equal\n";*

*if (A!=B) cout << "foo and bar are not equal\n";*

*if (A< B) cout << "foo is less than bar\n";*

*if (A>B) cout << "foo is greater than bar\n";*

*if (A<=B) cout << "foo is less than or equal to bar\n";*

*if (A>=B) cout << "foo is greater than or equal to bar\n";*

*return 0;*

*}*