//vector存放自定义数据类型

#include<iostream>

#include<string>

using namespace std;

#include<vector>

#include<algorithm>

class person {

public:

person(string name, int age) {

this->mname = name;

this->mage = age;

}

string mname;

int mage;

};

void test01() {

vector<person>v;

person p1("we1", 10);

person p2("we2", 20);

person p3("we3", 30);

person p4("we4", 40);

person p5("we5", 50);

//向容器中添加数据

v.push\_back(p1);

v.push\_back(p2);

v.push\_back(p3);

v.push\_back(p4);

v.push\_back(p5);

//遍历容器中的数据

for (vector<person>::iterator it = v.begin(); it != v.end(); it++)

{

//cout << "name=" << (\*it).mname << " age=" << (\*it).mage << endl;

cout << "name=" << it->mname << " age=" << it->mage << endl;

}

}

//存放自定义的数据类型的指针

void test02() {

vector<person\*>v;

person p1("we1", 10);

person p2("we2", 20);

person p3("we3", 30);

person p4("we4", 40);

person p5("we5", 50);

//向容器中添加数据

v.push\_back(&p1);

v.push\_back(&p2);

v.push\_back(&p3);

v.push\_back(&p4);

v.push\_back(&p5);

//遍历容器中的数据

for (vector<person\*>::iterator it = v.begin(); it != v.end(); it++) {

cout << "2name=" << (\*it)->mname << " 2age=" << (\*it)->mage << endl;

}

}

int main() {

test01();

test02();

system("pause");

return 0;

}