//类模板对象做函数参数

#include<iostream>

#include<string>

using namespace std;

template<class T1,class T2>

class person {

public:

person(T1 name, T2 age) {

this->Mname = name;

this->Mage = age;

}

void showp() {

cout<<"name="<<this->Mname<<"age="<<this->Mage<<endl;

}

T1 Mname;

T2 Mage;

};

//1.指定传入类型

void printp1(person<string,int>&p) {

p.showp();

}

void test01() {

person<string,int>p("asd",100);

printp1(p);

}

//2.参数模板化

template<class T1, class T2>

void printp2(person<T1,T2>&p) {

p.showp();

cout << "t1=" << typeid(T1).name() << endl;

cout << "t2=" << typeid(T2).name() << endl;

}

void test02() {

person<string,int>p("qwe",99);

printp2(p);

}

//3.整个类模板化

template<class T>

void printp3(T &p) {

p.showp();

cout << "t=" << typeid(T).name() << endl;

}

void test03() {

person<string,int>p("zxc",29);

printp3(p);

}

int main() {

test01();

test02();

test03();

system("pause");

return 0;

}