//类模板成员函数类外实现

#include<iostream>

#include<string>

using namespace std;

template<class T1,class T2>

class person {

public:

person(T1 name, T2 age); /\*{

this->mmame = name;

this->mage = age;

}\*/

void showp(); /\*{

cout << "name=" << this->mmame << " " << "age=" << this->mage << endl;

}\*/

T1 mmame;

T2 mage;

};

//构造函数类外实现

template<class T1,class T2>

person<T1, T2>::person(T1 name, T2 age) {

this->mmame = name;

this->mage = age;

}

//成员函数的类外实现

template<class T1,class T2>

void person<T1, T2>::showp() {

cout << "name=" << this->mmame << " " << "age=" << this->mage < endl;

}

void test01() {

person<string, int>p("sad", 20);

p.showp();

}

int main() {

test01();

system("pause");

return 0;

}