//类模板与友元

#include<iostream>

#include<string>

using namespace std;

//通过全局函数打印person信息

template<class T1,class T2>

class person;

//提前说明

//类外实现

template<class T1, class T2>

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

cout << "类外实现name=" << p.mname << " 类外实现age=" << p.mage << endl;

}

template <class T1,class T2>

class person {

//全局函数，类内实现

friend void printp(person<T1, T2>p) {

cout << "name=" << p.mname << " age=" << p.mage << endl;

}

//全局函数类外实现

//空模板的参数列表

//如果全局函数是类外实现，要让编译器提前知道这个函数存在

friend void printp2<>(person<T1, T2>p);

public:

person(T1 name, T2 age) {

this->mname = name;

this-> mage = age;

}

private:

T1 mname;

T2 mage;

};

//1.全局函数再类内实现

void test01() {

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

printp(p);

}

void test02() {

person<string, int>p("qrq", 21);

printp2(p);

}

int main() {

test01();

test02();

system("pause");

return 0;

}