```
//一：RTTI是什么(Run Time  Type Identification):运行时类型识别；
//通过运行时类型识别，程序能够使用基类的指针或者引用来检查这些指针或者引用所指的对象的实际派生类型。
Human *phuman = new Men;
Human &q = *phuman; //*phuman表示指针phuman所指向的对象
//RTTI我们可以把这个称呼看成是一种系统提供给我们的一种能力，或者一种功能。这种功能或者能力是通过2个运算符来体现：
//（1）dynamic_cast运算符：能够将基类的指针或者引用安全的转换为派生类的指针或者引用；
//（2）typeid运算符：返回指针或者引用所指对象的实际类型。
//老师补充：要想让RTTI的两个运算符能够正常工作，那么基类中必须至少要有一个虚函数，不然这两个运算符工作的结果就可能跟我们预测的不一样。
//因为只有虚函数的存在，这两个运算符才会使用指针或者引用所绑定的对象的动态类型（你new的类型）；
```

```
//五：RTTI与虚函数表
//c++中，如果类里含有虚函数。编译器就会对该类产生一个虚函数表。
//虚函数表里有很多项，每一项都是 一个指针。每个指针指向的是这个类里的各个虚函数的入口地址。
//虚函数表项里，第一个表项很特殊，它指向的不是虚函数的入口地址，它指向的实际上是咱们这个类所关联的type_info对象。

Human *phuman = new Men;
const type_info &ty = typeid(*phuman);

//phuman对象里有一个我们看不见的指针，这个指针指向谁，指向的是这个对象所在的类Men里的虚函数表。
```