基础6. 各种类型转换

①隐式类型转换 <<C++ Primer>> P141

②强制类型转换

编译期间进行

a. static_cast 静态类型转换

1. 基本类型转换 告诉其它人我知道这样做不危险
2. 任意类型指针转换都可以用 void* 指针做媒介
3. 存在继承且没有多态类的之间类型转换

   上行(子→父) √ 下行(父→子) X 内存可缩窄,但无法扩张
4. 存在多态类之间的类型转换

   上行(子→父) √ 多态消失 下行(父→子) X

   本质:虚表改变
5. 存在继承且没有多态类指针/引用之间转换:随便转但下行不安全
6. 存在多态类指针/引用之间转换:随便转但是某些下行不安全

转基类本身不安全,但是转基类的指针安全

b. dynamic_cast 动态类型转换 仅支持含虚函数的类

解决下行不安全的问题 运行时RTTI验证

将基类型的指针/引用安全地转成其派生类的指针/引用

<<C++ Primer>>

C. const_cast 静态类型转换 P145 P209

去掉底层const,但是修改会出现段错误

```
const char* pc = "hello world";
char* p = const_cast<char *>(pc);
p[1] = 'm';
```
段错误

作用：函数重载（避免代码重复）

d. reinterpret_cast 静态类型转换

为运算对象的位模式提供低层次上的重新解释

1、reinterpret_cast<type>(expression)

其中type与expression至少有一个是引用/指针

可以进行进制的转换 10→ 0xa

2、指针之间转换可直接使用，无需媒介 void*

3、无法修改const属性 cv