

知识点1【类型转换】（了解）

1、上行、下行转换

2、static_cast静态类型转换

3、dynamic_cast静态类型转换

4、const_cast常量转换

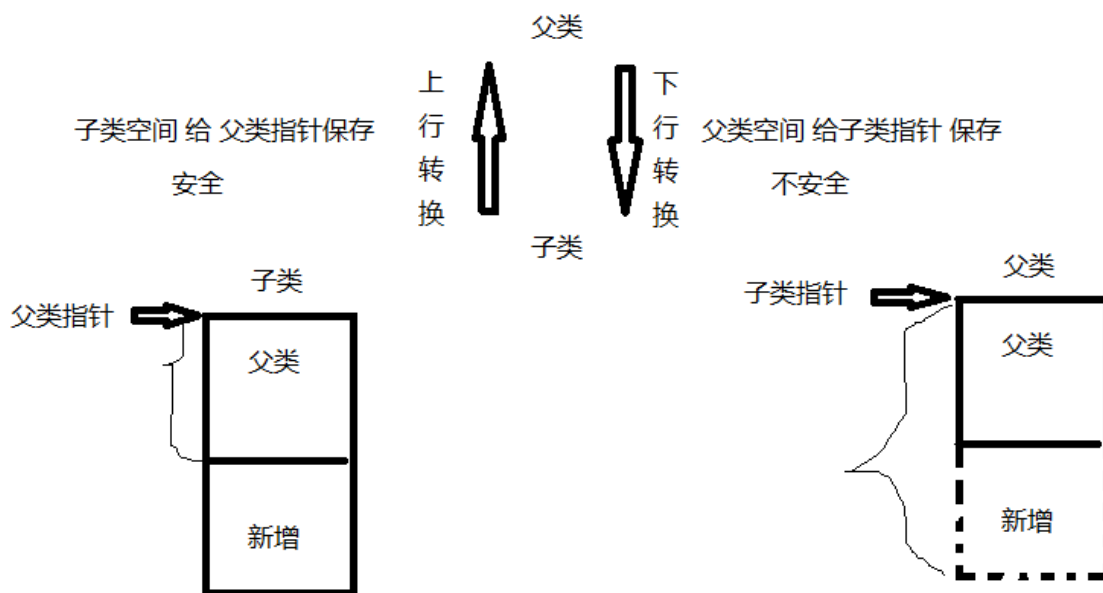
1、将const修饰的指针或引用 转换成 非const （支持）

2、将非const修饰的指针或引用 转换成 const （支持）

5、重新解释转换(reinterpret_cast)（最不安全）

知识点1【类型转换】（了解）

1、上行、下行转换



2、static_cast静态类型转换

```
1 class Base{};  
2 class Son:public Base{};  
3 class Other{};
```

用于类层次结构中基类（父类）和派生类（子类）之间指针或引用的转换。

基本类型：支持

```
1 int num = static_cast<int>(3.14); //ok
```

上行转换：支持 安全

```
1 Base *p = static_cast<Base *>(new Son);
```

下行转换：支持 (不安全)

```
1 Son *p2 = static_cast<Son *>(new Base);
```

不相关类型转换：不支持

```
1 Base *p3 = static_cast<Base *>(new Other); //err
```

3、dynamic_cast静态类型转换

dynamic_cast主要用于类层次间的上行转换和下行转换

基本类型：不支持

```
1 int num = dynamic_cast<int>(3.14); //err
```

上行转换：支持

```
1 Base *p1 = dynamic_cast<Base *>(new Son); //ok
```

下行转换：不支持 (不安全)

```
1 Son *p2 = dynamic_cast<Son *>(new Base); //err
```

不相关类型转换：不支持

```
1 Base *p3 = dynamic_cast<Base *>(new Other); //err
```

4、const_cast常量转换

1、将const修饰的指针或引用 转换成 非const (支持)

```
1 const int *p1;  
2 int *p2 = const_cast<int *>(p1);  
3  
4 const int &ob = 10;  
5 int &ob1 = const_cast<int &>(ob);
```

2、将非const修饰的指针或引用 转换成 const (支持)

```
1 int *p3;  
2 const *p4 = const_cast<int *>(p3);  
3  
4 int data = 10;  
5 const int &ob2 = const_cast<const int &>(data);
```

5、重新解释转换(reinterpret_cast) (最不安全)

```
44 void test04()
45 {
46     //基本类型: 不支持
47     int num = reinterpret_cast<int>(3.14f);
48     //基本类型指针: 支持
49     float *q;
50     int *p = reinterpret_cast<int *>(q);
51     //上行转换: 支持
52     Base *p1 = reinterpret_cast<Base *>(new Son);
53     //下行转换: 支持
54     Son *p2 = reinterpret_cast<Son *>(new Base);
55     //不相关类型转换: 支持
56     Base *p3 = reinterpret_cast<Base *>(new Other);
57 }
```