

1 静态转换

2 动态转换

3 常量转换

4 重新解释转换

5 总结

1 静态转换

`static_cast<待转换的类型>(待转换的数据)`

- `static_cast`可以用来转换基本的内置数据类型 `int char double...`
- `static_cast`不能转换没有发生继承关系之间的类
- `static_cast`可以用来转换发送继承关系之间的类,但是不保存安全性
- 不能用来转换指针

```
1 #define _CRT_SECURE_NO_WARNINGS
2 #include <iostream>
3 #include <string.h>
4 #include <stdlib.h>
5 #include <string>
6 using namespace std;
7
8 void test01()
9 {
10     //static_cast 用来转换内置的数据类型 和c语言的强制类型转换一样
11     int a = 1;
12     char b = 2;
13     double c = 3.14;
14     a = static_cast<int>(b);
15     a = static_cast<int>(c);
16     c = static_cast<double>(a);
17 }
18 class A
19 {
20 public:
21     int a;
```

```

22 };
23 class B:public A
24 {
25 public:
26     int b;
27 };
28 void test02()
29 {
30     A *p1 = new A;
31     B *p2 = new B;
32     //static_cast不能转换没有发生继承关系之间的类
33     //如果两个类之间发生了继承关系,可以类型转换 但是static_cast不会保证转换的安全性
34     p1 = static_cast<A *>(p2);//子转父 向上转换 是安全的
35     p2 = static_cast<B *>(p1);//父转子 向下转换 是不安全的
36 }
37 void test03()
38 {
39     int *p1 = NULL;
40     char *p2 = NULL;
41     //static_cast不能用来转指针
42     //p1 = static_cast<int *>(p2);
43
44 }
45
46 int main() {
47
48
49     return 0;
50 }

```

2 动态转换

- 不能用于转换基本的数据类型
- 可以用于转换发送继关系之间的类,保证转换是安全的 子转父是可以的
- 如果发生了多态,子转父和父转子总是安全的

```

1 //动态转换 dynamic_cast
2 void test04()
3 {
4     //动态转换不能转内置的基本数据类型

```

```

5  int a = 1;
6  char b = 2;
7  //a = dynamic_cast<int>(b);
8
9  }
10 void test05()
11 {
12     A *p1 = new A;
13     B *p2 = new B;
14     //dynamic_cast不能用于没有发生继承关系之间的类转换
15     //dynamic_cast可以用于发生继承关系之间的类号转换
16
17     p1 = dynamic_cast<A *>(p2); //子转父 是安全的
18     //p2 = dynamic_cast<B*>(p1); //父转子 不安全 不允许
19
20 }

```

3 常量转换

- `const_cast` 一般用来加const或去除const

```

1  //const转换
2  void test06()
3  {
4      int *p1 = NULL;
5      const int *p2 = NULL;
6      //int *p1 = static_cast<int *>(p2);
7      p1 = const_cast<int *>(p2);
8      p2 = const_cast< const int *>(p1);
9  }

```

4 重新解释转换

`reinterpret_cast`

- 一般用来转换指针 整数和指针之间都可以转换

```

1  void test07()
2  {

```

```
3  int *p = NULL;
4  char *p2 = NULL;
5  p = reinterpret_cast<int *>(p2);
6  p2 = reinterpret_cast<char *>(p);
7  int c = 0;
8  c = reinterpret_cast<int>(p2);
9 }
```

5 总结

static_cast 一般用来转换内置的基本数据类型

dynamic_cast 一般用来转发生继承关系之间的自定义的数据类型

const_cast 一般用来转换加const和去除const

reinterpret_cast 一般用来转指针