

// (2.1) static_cast: 静态转换, 大家就理解成“正常转换”, 编译的时候就会进行类型转换的检查。
// 代码中要保证转换的安全性和正确性, static_cast含义跟c语言中的强制类型转换这种感觉差不多。
// c风格的强制类型转换, 以及 编译器能够进行的隐式类型转换, 都可以用static_cast类显式完成。

在编译时就执行转换

```
// 可用于:  
// a) 相关类型转换, 比如整型和实型之间的转换:  
// double f = 100.34f;  
// int i = (int)f; // c风格的  
// int i2 = static_cast<int>(f); // c++风格的类型转换。  
// b) 后续学习类的时候子类转成父类类型 (继承关系), 也能用这个static_cast转。  
// class A {}  
// class B : public A {}  
// B b;  
// A a = static_cast<A>(b); // 把子类转成父类的对象。  
// A a;  
// B b = static_cast<B>(a);  
// c) void *与其他类型指针之间的转换, void *: 无类型指针: 可以指向任何指针类型 (万能指针);  
// int i = 10;  
// int *p = &i;  
// void *q = static_cast<void *>(p); // int *转成void *  
// int *db = static_cast<int *>(q);
```