

3.20. Инициализация родителя наследником и запрет преобразования в обратную сторону. То же самое со ссылками и указателями.

```
1 struct Base {
2     int a;
3 };
4
5 struct Derived: public Base {
6     int b;
7 };
8
9 int main() {
10     Base b;
11     Derived d;
12
13     Base b0 = d; // так можно
14     Derived d0 = b; // так нельзя
15
16     Base& b1 = d; // так можно
17     Derived& d1 = b; // так нельзя, будет СЕ
18
19     // Чтобы всё-таки привести объект типа Base к типу Derived,
20     // нужно воспользоваться static_cast
21
22
23     Derived& d2 = static_cast<Derived&>(b1);
24     // Всё пройдет успешно, так как под b1 на самом деле лежит объект типа Derived
25
26     Derived& d3 = static_cast<Derived&>(b);
27     // Никакой ошибки не выведется, однако будет UB, так как под b на самом деле не лежит
    объект типа Derived
28
29     Base* b4 = &d;
30     Derived* d4 = static_cast<Derived*>(b4);
31     // так можно
32
33     Derived* d5 = &b;
34     Base* b6 = static_cast<Base*>(d5);
35     // так нельзя
36     return 0;
37 }
38 }
```