

Dynamic Type Casts

1.1. Warum Type Casts?

In objektorientierten Programmen kommen zwei Fälle von Typumwandlungen besonders häufig vor: Type casts entlang der Vererbungshierarchie hinauf und herunter.

Zwei Beispiele anhand folgender Klassenhierarchie:

```
class tool {  
    public:  
    tool();  
    virtual ~tool();  
};
```

```
class hammer: public tool {  
    public:  
    hammer();  
    virtual ~hammer();  
};
```

1.2. Statischer vs. impliziter Type Cast

Beispiel 1: die Vererbungshierarchie hinauf (Generalisierung)

```
void AddToToolList(tool *ptool);
```

```
...  
hammer *phammer;  
phammer = new hammer;  
AddToToolList((tool*)phammer);
```

Zeiger auf abgeleitete Klassen sind auf Zeiger einer passenden Basisklasse zuweisungskompatibel. Damit kann der Type Cast implizit durchgeführt werden, wodurch der Compiler gleichzeitig die Gültigkeit der Beziehung zwischen den Zeigertypen überprüft:

```
void AddToToolList(tool *ptool);
```

```
...  
hammer *phammer;  
phammer = new hammer;  
AddToToolList(phammer);
```

1.3. Dynamischer Type Cast

Beispiel 2: die Vererbungshierarchie herunter (Spezialisierung)

Wenn Referenzen auf abgeleitete Klassen durch generische Programmteile verwaltet werden (z.B. Listen), geht der tatsächliche Typ eines Objekts aus dem statischen Typ im Quellcode nicht mehr hervor.

Ein Type Cast wird notwendig, wenn spezielle Eigenschaften der tatsächlichen Klasse eines Objekts benötigt werden.

Ein statischer Type Cast ist gefährlich, weil der Typ eines Objekts ohne Überprüfung falsch interpretiert werden kann und dann zu unvorhersehbarem Programmverhalten führt.

Ein dynamischer Type Cast überprüft zur Laufzeit den tatsächlichen Typ des Objekts. Stimmt dieser mit dem Type Cast nicht überein, ist das Ergebnis des Type Casts eines Zeigers NULL.

```
tool *ptool;  
...  
hammer *phammer;  
phammer = dynamic_cast<hammer*>(ptool);  
if (phammer!=NULL) {  
    phammer->...  
}
```

1.4. Fazit

1. Wenn ein statischer Type Cast ausgelassen werden kann, weil die Zeiger zuweisungskompatibel sein sollten, soll der Type Cast ausgelassen werden.
2. Wenn ein Type Cast nicht vermieden werden kann, weil er die Vererbungshierarchie entlang herunter geht, soll ein dynamischer Type Cast verwendet werden.
3. Wenn trotz allem ein statischer Type Cast benötigt wird, so sollte die Notwendigkeit in einem Kommentar begründet werden.