

- Wie bei Objektzeigern Verwendung als **boolescher Wert** möglich
- **Aber:** Anders als bei Objektzeigern **keine Zeigerarithmetik**
- **Adressoperator & <Ausdruck>:**
Anwendung auf eine Funktion liefert einen Zeiger auf die Funktion
- **Funktionsaufruf <Ausdruck> (<tatsächliche Argumente*>):**
Bisher war der <Ausdruck> für uns lediglich ein Funktionsbezeichner, in Wahrheit wird hier ein Ausdruck erwartet, der zu einem **Zeiger auf eine Funktion** auswertet => **keine Dereferenzierung** für Aufruf erforderlich
- **Implizite unäre Konvertierungen:** Implizite Konvertierung einer Funktion in einen Zeiger auf die Funktion

```
int add(int l, int r) {  
    return l + r;  
}
```

```
int (*f)(int, int) = add;      /* OK */ implizite Konvertierung  
int (*f)(int, int) = &add;    /* OK */ explizite Konvertierung  
int result = f(5,6);          /* result == 11 */  
int result = add(5,6);        /* result == 11 */  
int result = (&add)(5,6);     /* result == 11 */  
int equals = (add == &add);   /* equals == 1  */
```