

## Einführung in C++

Übersetzen und starten Sie das gegebene Programm `... \UE09 \Vorgabe \rechteck1.cpp` und machen Sie sich mit der Funktionsweise vertraut.

Beachten Sie:

- alle Methoden im Beispiel sind inline-Methoden
- C++ kennt den Typ `bool`

### Aufgabe: rechteck2

Teilen Sie die Klasse `Rechteck` auf in eine Klassendeklarationsdatei und eine Klassendefinitiondatei. Die Anwendung `rechteck2.cpp` soll zunächst die `main`-Funktion aus `rechteck1.cpp` übernehmen.

Die drei Dateien werden wie folgt compiliert und gelinkt:

```
g++ -Wall -c Rechteck.cpp -o Rechteck.o
g++ -Wall -c rechteck2.cpp -o rechteck2.o ← Option -c : nur compilieren

g++ -o rechteck2 Rechteck.o rechteck2.o ← Linker-Aufruf
```

Nun sollte alles funktionieren wie gehabt.

Erweitern Sie die Klasse `Rechteck` um die Methode `drehDich90Grad`, sodass dies möglich ist:

```
int main()
{
    Rechteck r1(7, 4);

    r1.zeigDich();

    r1.drehDich90Grad();

    r1.zeigDich();
}
```

Ausgabe:

```
+---+
!   !
!   !
!   !
!   !
!   !
+---+
+-----+
!       !
!       !
!       !
+-----+
```

**ACHTUNG:** Die Methode `drehDich90Grad` soll den Zustand eines `Rechteck`-Objekts ändern! Schreiben Sie keine neue Ausgabe-Methode; `zeigDich` soll nicht verändert werden.



**Aufgabe: rechteck4**

Erweitern Sie das Projekt wie folgt:

- einen Konstruktor zum Instanzieren von Quadraten hinzufügen:  
`Rechteck(int kantenLaenge);`
- Einen weiteren Konstruktor für Rechtecke, der es erlaubt das „Eckzeichen“ zu definieren:  
`Rechteck(int hoehe, int breite, char eckChar);`

Das Eckzeichen soll im Heap gespeichert werden. Als Attribut muss daher ein Zeiger angelegt werden. Der Konstruktor erzeugt dann das Heapobjekt und initialisiert den Zeiger.

**Achtung:** ALLE Konstruktoren müssen jetzt das neue Attribut initialisieren. Der Standardwert soll das '+'-Zeichen sein.

Verwenden Sie hier die Konstruktor-Initialisierungsliste.

- Die Methode `zeigDich()` muss leicht modifiziert werden, damit das Eckzeichen auch angezeigt wird.
- Ein Destruktor muss definiert werden, der das Heap-Eckzeichen freigibt. Zusätzlich soll der Destruktor die Objekte individuell verabschieden - siehe unten.

Folgende Anweisungen erzeugen dann die gezeigte Ausgabe:

```
{
    Rechteck a(5);
    a.zeigDich();
    {
        Rechteck b(4, 8, 'X');
        b.bewegDichUm(5, 1);
        b.zeigDich();

        Rechteck c(3, 14, 'o');
        c.bewegDichUm(10, 1);
        c.zeigDich();
    }
}
```

Ausgabe:

```
+---+
!   !
!   !
!   !
+---+
```

```
X-----X
!       !
!       !
X-----X
```

```
o-----o
!       !
o-----o
```

Tschüss 3x14 "o"-Rechteck!

Tschüss 4x8 "X"-Rechteck!

Tschüss 5x5 "+"-Quadrat!