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
                                              Option -c : nur compilieren
g++ -Wall -c rechteck2.cpp -o rechteck2.o
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: rechteck3

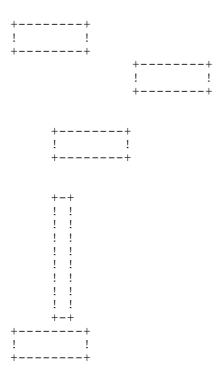
Die Klasse Rechteck soll wie folgt erweitert werden:

- Als zusätzliche Eigenschaft kennt jedes Rechteckobjekt seine Position (die linke obere Ecke). Die Konstruktoren initialisieren diese Position mit (0, 0).
- Die Methode void bewegDichUm(int deltaX, int deltaY) ändert die Position um deltaX bzw. deltaY. Da auch negative Verschiebungen möglich seien sollen, muss sichergestellt werden, dass dadurch die Positionswerte nicht negativ werden.
- Die Methode zeigDich () muss jetzt die Position berücksichtigen, also eine entsprechende Anzahl Leerzeilen bzw. Leerspalten ausgeben. Es soll nicht mit ANSI-Escape Sequenzen gearbeitet werden.

Beispiel:

```
int main()
  Rechteck r(3, 10);
  r.zeigDich();
  r.bewegDichUm(15, 0);
  r.zeigDich();
  r.bewegDichUm(-10, 2);
  r.zeigDich();
  r.drehDich90Grad();
  r.zeigDich();
  r.bewegDichUm(-10, -10);
  r.drehDich90Grad();
  r.zeigDich();
```

Ausgabe:





Aufgabe: rechteck4

Erweitern Sie das Projekt wie folgt:

einen Konstruktor zum Instanziieren 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.zeiqDich();
```

Ausgabe:

