

# Vorkurs Programmierung

## – Übungsblatt 9 (Tag 12) –

### Fachgruppe für Computergraphik und Multimediasysteme

Andreas Kolb, Nicolas Cuntz, Felix Heide

#### Aufgabe 1 (Rechtecke)

Es soll im Folgenden ein Programm geschrieben werden, das eine Liste von Rechtecken mit unterschiedlichen Positionen und Dimensionen im Terminal darstellt. Die Rechtecksliste wird zu Beginn über Terminal-Eingaben eingelesen und in eine dynamische Struktur gespeichert. Der Bereich, in dem Rechtecke gezeichnet werden, ist  $40 \times 20$  ASCII-Zeichen groß. Jedes Zeichen in diesem Bereich kann entweder ein 0-Zeichen oder ein Leerzeichen sein.

Die Rechtecke in der Liste werden in der Reihenfolge gezeichnet, in der sie eingelesen worden sind. Jedes Rechteck kann ein *additives* oder ein *subtraktives* sein. Additiv bedeutet, dass das Rechteck mit 0-Zeichen gefüllt wird. Subtraktiv bedeutet, dass das Rechteck ausgestanzt wird und somit mit Leerzeichen gefüllt wird. Nach Ablauf des Programms ergibt sich beispielsweise das folgende Bild:

```

00000000000000000000000000000000000000000000000000000000000000000000
0
0
0 00000000000000000000000000000000000000000000000000000000000000000000
0 00000000000000000000000000000000000000000000000000000000000000000000
0 00000000000000000000000000000000000000000000000000000000000000000000
0 00000000000000000000000000000000000000000000000000000000000000000000
0 00000000000000000000000000000000000000000000000000000000000000000000
0
0 00000000000000000000000000000000000000000000000000000000000000000000
0
0 00000000000000000000000000000000000000000000000000000000000000000000
0
0 00000000000000000000000000000000000000000000000000000000000000000000
0
0 00000000000000000000000000000000000000000000000000000000000000000000
0
0 00000000000000000000000000000000000000000000000000000000000000000000
0
0 00000000000000000000000000000000000000000000000000000000000000000000
0
0
00000000000000000000000000000000000000000000000000000000000000000000

```

Gehen Sie bei der Programmierung folgendermaßen vor:

- In der ersten Phase werden alle Rechtecksdaten über eine Kommandozeileingabe eingelesen. Hierbei sollen pro Rechteck zwei Integer-Koordinaten (x, y), zwei integer-Ausdehnungsgrößen (w, h) eingelesen sowie ein Datum, das beschreibt, ob das Rechteck additiv oder subtraktiv gezeichnet wird. Erstellen Sie eine Klasse, die ein Rechteck beschreibt und durch Zeigerverkettung zu einer Liste hinzugefügt werden kann.

Die Eingabe kann beispielsweise in der folgenden Weise erfolgen. Die gezeigte Eingabe entspricht dem oben abgebildeten Resultat.

```
Rechteck hinzutun? [j/n] j
x? 0 y? 0 w? 40 h? 20
Additiv/subtraktiv? [a/s] a
```

```
Rechteck hinzutun? [j/n] j
x? 1 y? 1 w? 38 h? 18
Additiv/subtraktiv? [a/s] s
```

```
Rechteck hinzutun? [j/n] j
x? 4 y? 3 w? 16 h? 5
Additiv/subtraktiv? [a/s] a
```

```
Rechteck hinzutun? [j/n] j
x? 20 y? 8 w? 17 h? 10
Additiv/subtraktiv? [a/s] a
```

```
Rechteck hinzutun? [j/n] j
x? 15 y? 5 w? 10 h? 10
Additiv/subtraktiv? [a/s] s
```

```
Rechteck hinzutun? [j/n] n
```

- b) In der zweiten Phase schreiben Sie eine Routine, welche die Rechtecke im Terminal ausgibt. Gehen Sie hierfür alle Ausgabekoordinaten durch, und prüfen Sie für jede Koordinate, welche Rechtecke in welcher Reihenfolge zum Ausgabezeichen beitragen.

Hinweis: Es ist sinnvoll, in der Rechtecksklasse eine Funktion hinzuzufügen, die prüft, ob eine gegebene Position im Rechteck enthalten ist oder nicht.

- c) Zum Schluss sollten Sie dafür sorgen, dass der Speicher für dynamisch angelegte Rechtecke vor Programmabbruch freigegeben wird.