

Konzept zu Aufgabe 10

Aufgabenstellung:

1. Das Weihnachtsbaummuster ist rekursiv definiert:
Das Muster der Ordnung 1 besteht aus der Zeile „0 1“.
2. Das Muster der Ordnung $n + 1$ entsteht aus dem Muster der Ordnung n , indem jede Zeile $s_1 \dots s_t$ des Musters der Ordnung n durch die Zeilen:
 $s_{2t} \dots s_{2t+1} s_{2t+2} \dots s_{2t+3} \dots s_{2t+4} \dots s_{2t+5}$
ersetzt (für $t = 1$ entfällt die erste Zeile). Diese Rekursion liefert alle 2^n Bitstrings der Länge n in einer jahreszeitgemäßen Anordnung!
3. Stellen Sie das Weihnachtsbaummuster dar! Wie Sie das tun ist Ihnen weitestgehend freigestellt. Swing oder Swing Applet sind m.E. gute Möglichkeiten. „Brennende“ Kerzen sind nicht gefordert.

Überlegung:

Was wird gebraucht?

ChristmasTreePattern:

Diese Klasse besitzt 2 Methoden die nach dem in der Aufgabenstellung vorgegebenen Muster einen Weihnachtsbaummuster aus Einsen und Nullen formt. Dies geht rekursiv von statten, die Methode `getChrisTreeLines()` ruft sich solange selbst auf bis n und somit auch der Baum, die gewünschte Größe haben. Die genutzten Datenstrukturen für die Strings und Zeilen sind im Grunde 2 ArrayLists die für die Zeilen hält Elemente über `List<Strings>`, die wiederum Elemente über Strings hält. Das erleichtert das iterieren ungemein. Die Methode `createChristree()` ist der Aufrufer der rekursiven Methode und Formatiert den letztendlichen Baum mit HTML-Tags. In den Kommentaren finden sich weitere angaben zum Ablauf der Rekursion.

ChristmasTreeView:

Die View erstellt nur einen JFrame, der eine JEditorPane beinhaltet. Diese wird gebraucht um die HTML-Tags umzusetzen und sie ist nicht veränderbar. Somit kann der Baum damit zentriert und entsprechend dem Pattern dargestellt werden.

MainApp:

Diese Klasse dient wie immer als Einstiegspunkt in das Programm und beinhaltet die Main-Methode. Hier kann man auch die Größe der Elemente des Baums angeben.

UML:

Siehe a10 package.