

Übung 03: Objekthierarchien

Abgabetermin: 23. 3. 2017, 8:15

Name: _____

Matrikelnummer: _____

Informatik: G1 (Marr) G2 (Prähofer) G3 (Prähofer) G4 (Löberbauer)

WIN: G1 (Khalil) G2 (Hummel) G3 (Khalil)

| Aufgabe | Punkte | abzugeben schriftlich | abzugeben elektronisch | korr. | Punkte |
|---------|--------|---|------------------------|--------------------------|--------|
| Übung 3 | 24 | Java-Programm, Screenshots von den Testläufen | Java-Programm | <input type="checkbox"/> | |

Übung 03: Seitenlayout

(24 Punkte)

Beim Setzen einer Zeitung ist eine wichtige Aufgabe, die unterschiedlichen Texte in mehrere Spalten und Zeilen anzuordnen. Wir wollen diese Aufgabe unterstützen, indem wir ein Klassensystem für die Darstellung und die Anordnung von Textblöcken einer Seite implementieren und diese Anordnung graphisch ausgeben.

Anmerkung: Ausrichtung der Texte und optimale Anordnung ist nicht Teil der Aufgabe.

a) Klassenhierarchie von Textblöcken

(14 Punkte)

Eine Seite sei als hierarchische Anordnung von Blöcken organisiert. Es gibt:

- Textblöcke für einzelne Textzeilen (speichern einen einzeiligen String)
- Blöcke, die mehrere Zeilen beinhalten, wobei jede Zeile wieder ein beliebiger Block sein kann
- Blöcke, die mehrere Spalten beinhalten, wobei jede Spalte wieder ein beliebiger Block sein kann

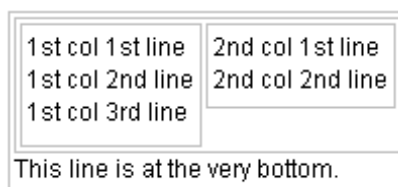
Für die Ausgabe verwenden wir die Klasse `Window` (siehe Anhang für Erklärungen zur Verwendung von `Window`). Mit `Window` werden die Blöcke an einer bestimmten `x/y`-Position ausgegeben. Dabei soll

- ein einfacher Textblock seinen Text mit der Methode `Window.drawText` an der entsprechenden Position ausgeben
- ein Mehrzeilenblock seine Zeilenblöcke untereinander ausgeben
- ein Mehrspaltenblock seine Spaltenblöcke nebeneinander ausgeben

Zusätzlich sollen bei Mehrzeilenblöcken und Mehrspaltenblöcken mit

`Window.drawRectangle(x, y, w, h, Color.LIGHT_GRAY);`

ein Rahmen in hellgrauer Farbe gezeichnet und ein kleiner Abstand eingefügt werden. Eine Ausgabe einer Anordnung von Blöcken könnte folgend aussehen:



Für die richtige Positionierung der Blöcke muss man die Weite und Höhe der Blöcke bestimmen. Implementieren Sie für die Blöcke Methoden `width()` und `height()` wie folgt:

- für einen einfachen Textblock kann die Breite und Höhe auf Basis des Strings mit den Methoden `getTextWidth` und `int getTextHeight` von `Window` bestimmt werden
- für einen Mehrzeilenblock ergibt sich die Breite aus dem Maximum der Breiten der Unterblöcke und die Höhe als Summe der Höhen der Unterblöcke (plus benötigter Abstände)
- für einen Mehrspaltenblock ergibt sich die Breite als Summe der Breiten der Unterblöcke und die Höhe aus dem Maximum der Höhen der Unterblöcke (plus benötigter Abstände)

b) Renderable

(3 Punkte)

Führen Sie ein Interface `Renderable` ein, das die Methoden

- `render(int x, int y)`
- `render()`

für eine graphische Ausgabe deklariert. Dabei soll die erste Methode die Ausgabe an der gegebenen x/y-Position und die zweite die Ausgabe an der linken oberen Ecke durchführen.

c) Factory-Methoden

(4 Punkte)

Implementieren Sie dann eine Klasse `Blocks` mit statischen Factory-Methoden zum Erzeugen der unterschiedlichen Blöcke wie folgt:

- `line(String line)` – erzeugt einen einfachen Textblock
- `lines(Block... blocks)` – erzeugt aus den übergebenen Blöcken einen Mehrzeilenblock
- `lines(String... lines)` – erzeugt aus den übergebenen Textzeilen Textblöcke und fügt diese zu einem Mehrzeilenblock zusammen
- `cols(Block... blocks)` – erzeugt aus den übergebenen Blöcken einen Mehrspaltenblock

Mit diesen Factory-Methoden kann man zum Beispiel die in obiger Abbildung dargestellte Blockstruktur folgend erzeugen:

```
lines(
    cols(
        lines("1st col 1st line", "1st col 2nd line", "1st col 3rd line"),
        lines("2nd col 1st line", "2nd col 2nd line")),
    line("This line is at the very bottom. ")
);
```

d) Test

(3 Punkte)

Testen Sie Ihr Programm, indem Sie mehrere unterschiedliche (mindestens 3) Blockhierarchien (mit mindestens 12 Blöcken) erzeugen und diese graphisch ausgeben.

Hinweise zu Verwendung der Window-Klasse

Die Klasse `Window` bietet eine Reihe von statischen Methoden zur einfachen graphischen Ausgabe auf einem Swing-Fenster. Das Fenster wird dabei automatisch geöffnet. Angebotene Methoden sind unter anderem:

- `drawText(String text, int x, int y)`
- `drawLine (int x1, int y1, int x2, int y2)`
- `drawRectangle (int x, int y, int w, int h)`

zum Zeichnen von Text, Linien und Rechtecken bzw.

- `drawText(String text, int x, int y , Color color)`
- `drawLine (int x1, int y1, int x2, int y2 , Color color)`
- `drawRectangle (int x, int y, int w, int h , Color color)`

zum Zeichnen Text, Linien und Rechtecken in einer bestimmten Farbe und

- `int getTextWidth(String text)`
- `int getTextHeight()`

zum Bestimmen von Breite eines gegebenen Textes und Höhe von Texten.