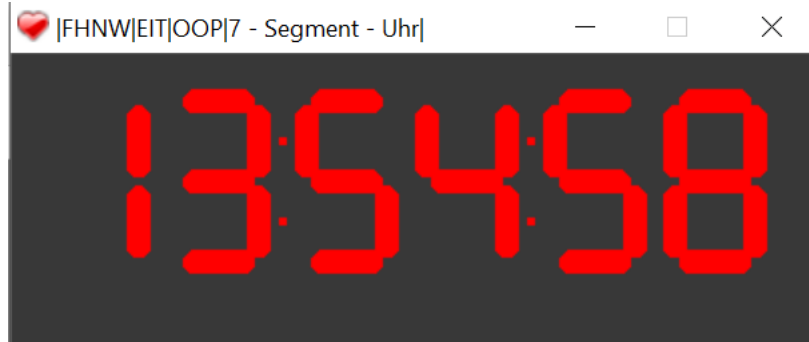


Übung : Vererbung - Siebensegment Uhr

Ziel dieser Übung ist es, ein Programm zu erstellen, das die aktuelle Zeit mittels Siebensegment-Anzeige darstellt

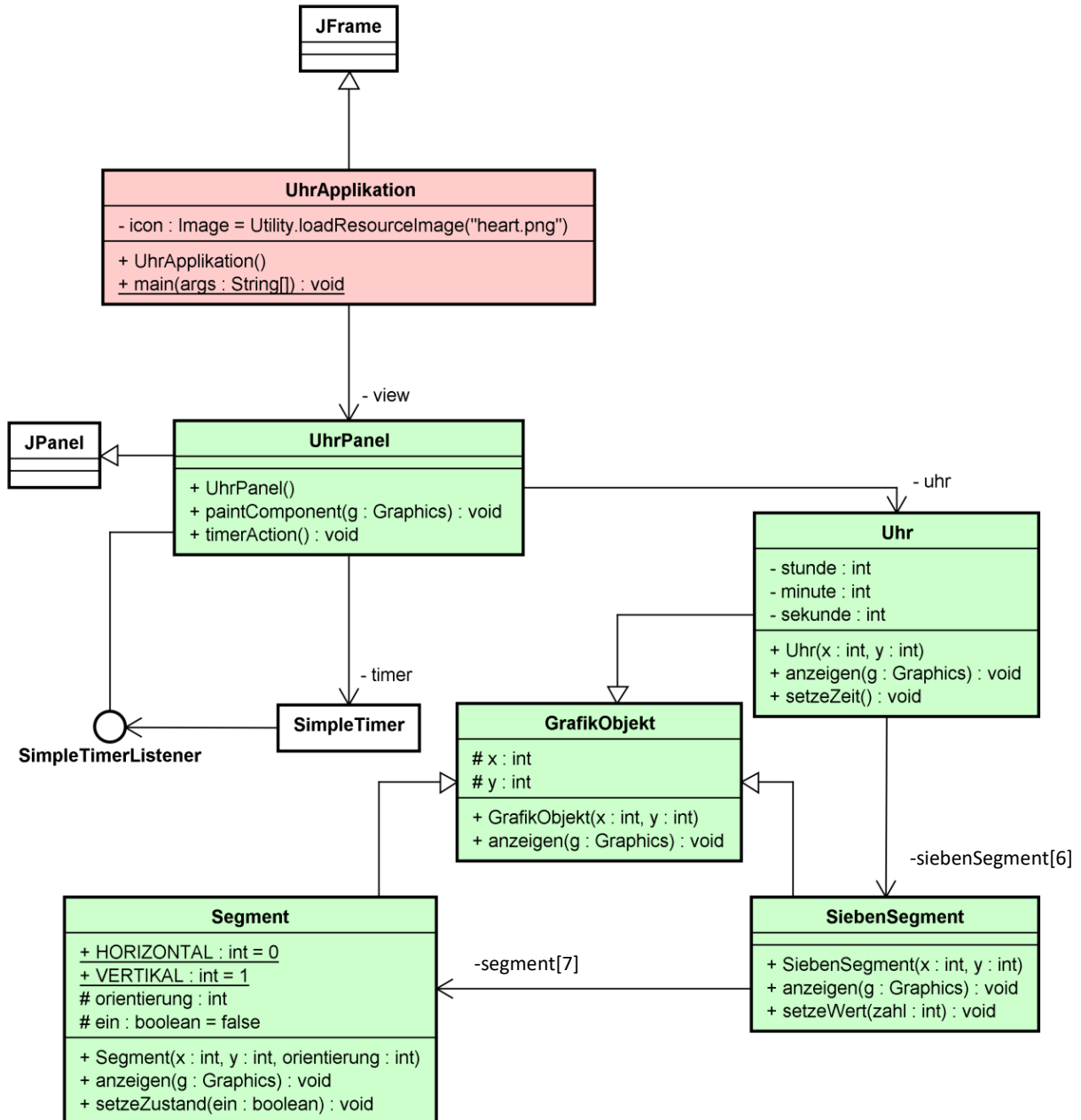


Die Struktur des Codes sieht folgendermassen aus:

- Die Klasse **UhrPanel** erweitert **JPanel** und implementiert den **SimpleTimerListener**. **UhrPanel** hat ein Objekt der Klasse **Uhr**.
- Das **Panel** erzeugt weiter ein Objekt **timer** der Klasse **SimpleTimer**. Der Timer sorgt dafür, dass im Intervall von einer Sekunde die Zeit gesetzt und neu gezeichnet wird
- Die Klasse **Uhr** ist ein **GrafikObjekt** und implementiert die zur Darstellung notwendige Anzeige und verfügt über die Methode **setzeZeit()**.
- Die Ziffern sind Instanzen der Klasse **SiebenSegment**. Diese ist ein **GrafikObjekt** und verfügt über die Methoden **setzeWert()** und **anzeigen()**. Die Methode **setzeWert()** erlaubt die darzustellende Zahl festzulegen. Mit **anzeigen()** wird die Ziffer an der bei der Instanziierung definierten Position gezeichnet.
- Die einzelnen Segmente der Klasse **SiebenSegment** sind Instanzen der Klasse **Segment**. Die Klasse **Segment** ist ein **GrafikObjekt** und erlaubt an gewünschter Position ein horizontales oder vertikales Segment zu erzeugen. Sie verfügt neben der Methode **anzeigen()** über eine Methode **setzeZustand()**, mit der die Farbe des Segments von schwarz (aus) auf rot (ein) gewechselt werden kann.

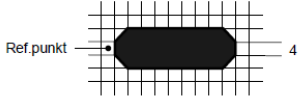
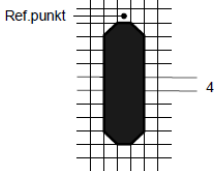
Aufgabe 1: Grundgerüst

Zunächst machen wir uns mit dem Grundgerüst des Programms anhand des folgenden Klassendiagramms vertraut. **Ergänzen Sie allenfalls fehlende Attribute.**



Aufgabe 2: Klasse Segment

Die Klasse **Segment** erlaubt ein Segment mit gewünschter Ausrichtung und Referenzpunkt x, y zu erzeugen. Sie verfügt über die im Klassendiagramm genannten Attribute und Methoden.

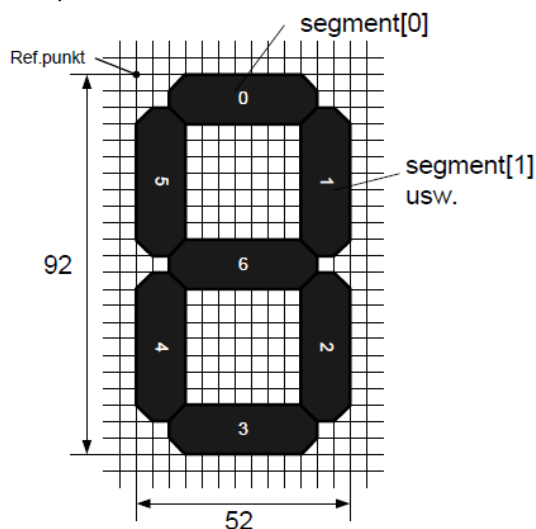
<code>new Segment(x, y, Segment.HORIZONTAL);</code>	<code>new Segment(x, y, Segment.VERTIKAL);</code>
	

- Implementieren Sie die Methoden der Klasse entsprechend obiger Angaben und der Dokumentation im Code. Achten Sie auf den Referenzpunkt.
- Testen Sie die Klasse in dem Sie in **UhrPanel** ein horizontales und vertikales Segment erzeugen und zur Darstellung bringen. Kontrollieren Sie ob der Referenzpunkt an der richtigen Stelle zu liegen kommt.

Aufgabe 3: Klasse SiebenSegment

Die Klasse baut Siebensegment-Anzeigen an gewünschter Stelle. Sie verwendet dazu Objekte der Klasse Segment, um an entsprechender Stelle x, y vertikale oder horizontale Segmente zu erzeugen.

- Schreiben Sie den Konstruktor der Klasse **SiebenSegment** aufgrund folgender Vorgabe:



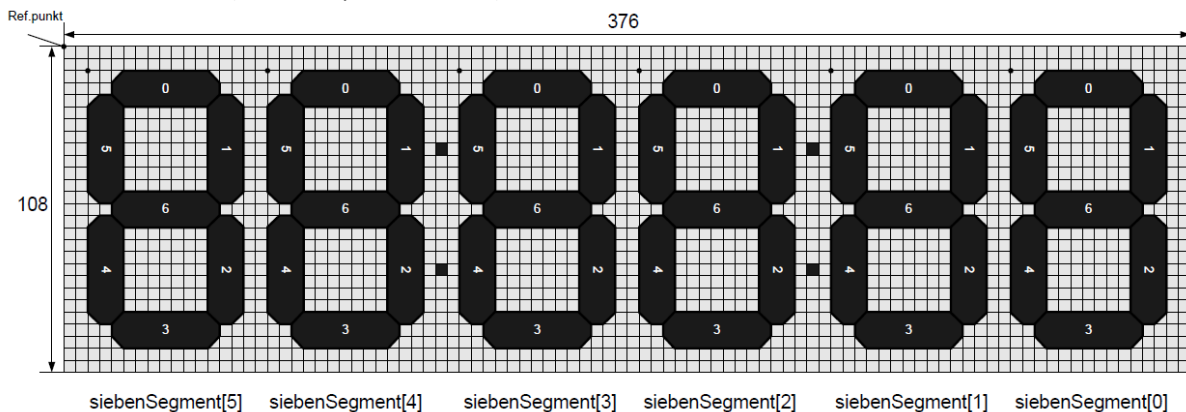
In allen Abbildungen ist das Raster (4 px Abstand zw. Linien) als Hilfe gedacht und soll nicht gezeichnet werden! **Bitte beachten Sie die Referenzpunkte!**

- Implementieren Sie die Methode **anzeigen()** der Klasse **SiebenSegment** gemäss Beschreibung im Code.
- Implementieren Sie die Methode **setzeWert()** der Klasse gemäss Dokumentation und unter Verwendung der **switch ()**-Anweisung.
- Testen Sie die Klasse **SiebenSegment** indem sie im **UhrPanel** eine Ziffer an der Position 40,40 erzeugen und dafür sorgen, dass sie dargestellt wird

Aufgabe 4: Klasse Uhr

Die Klasse **Uhr** implementiert die zur Darstellung notwendige Anzeige. Jedes Mal wird beim erneuten Ablauf des Timers in **timerAction()** mittels **setzeZeit()** die Zeit aktualisiert und via **repaint()** dafür sorgt, dass durch **paintComponent()** die aktuelle Zeit dargestellt wird.

- a) Implementieren Sie den Konstruktor der Klasse **Uhr** aufgrund nachfolgender Skizze. Beachten Sie wiederum den Referenzpunkt und die Reihenfolge der siebenSegment Elemente ! (Raster 4 px zw. Linien)



- b) Schreiben Sie die Methode **anzeigen()** gemäss Beschreibung im Code.
- c) Implementieren Sie die Methode **setzeZeit()**. Stunde, Minute und Sekunde können dabei wie folgt mittels der Klasse **GregorianCalendar** bestimmt werden
(**import** java.util.GregorianCalendar; **import** java.util.Calendar;):

```
GregorianCalendar calendar = new GregorianCalendar();
stunde = calendar.get(Calendar.HOUR_OF_DAY);
minute = calendar.get(Calendar.MINUTE);
sekunde = calendar.get(Calendar.SECOND);
```

Setzen Sie dann den Wert der sechs Ziffern entsprechend der Zeit

Aufgabe 5: SimpleTimer

Im letzten Schritt wollen wir unserer Uhr noch Leben einhauchen, indem wir dafür sorgen, dass die Uhr instanziiert und im Sekundentakt zum Neuzeichnen aufgerufen wird. Wir bedienen uns dazu der Klasse **SimpleTimer**, die einen einfachen Timer implementiert und bereits vorgegeben ist. Die einzige Methode, die das Interface **SimpleTimerListener** spezifiziert, ist die Methode **timerAction()**, die dann regelmässig im Timer-Intervall aufgerufen wird.

- a) Instanziiieren Sie im Konstruktor von **UhrPanel** einen **SimpleTimer** mit Bezeichnung **timer**
`timer = new SimpleTimer(1000, this);`
und starten sie ihn.
- b) Ergänzen Sie die Methode **timerAction()** derart, dass die Uhr alle Sekunde neu gesetzt und gezeichnet wird.