

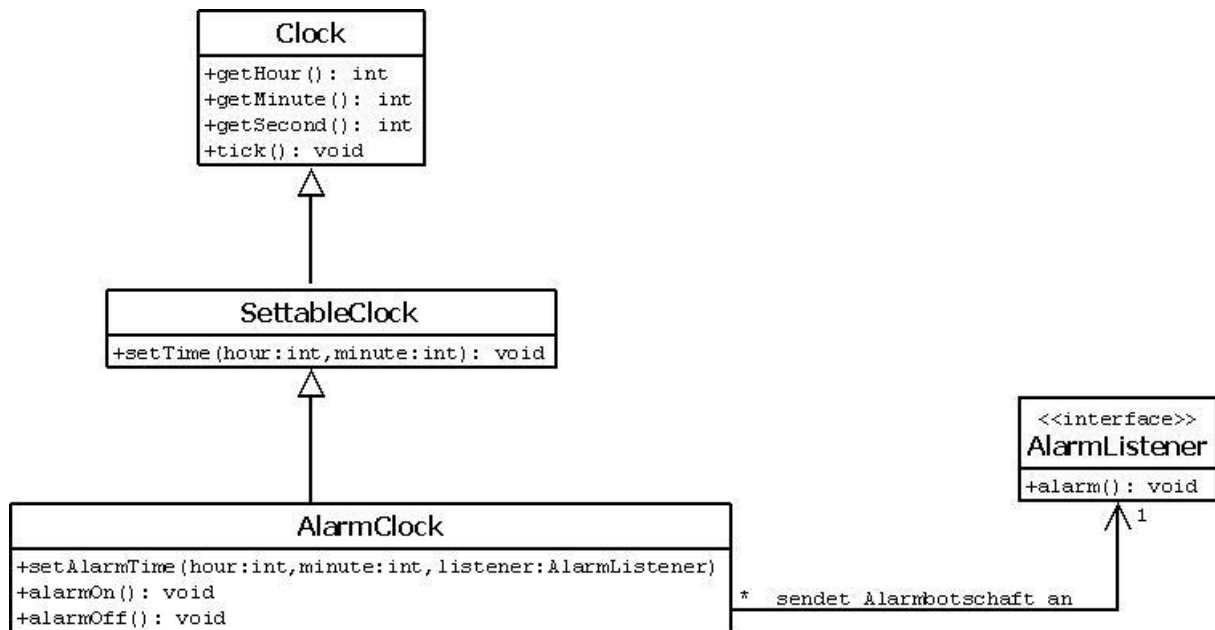
## Übungsblatt 2

### Hinweis:

- Zur Lösung dieser Aufgabe bilden Sie 2-er Teams.

### 2.1 Clock Hierarchie (Design by Contract, Liskov Prinzip)

Gegeben ist folgende Klassenhierarchie:



- **get** Methoden sind selbsterklärend
  - **tick** schaltet die Uhr um eine Sekunde weiter
  - **setTime** setzt die Uhrzeit auf neue Zeit
  - **setAlarmTime** stellt die Alarmzeit ein. Außerdem wird ein Objekt übergeben, das das **AlarmListener** interface implementiert hat. Wenn der Alarm an ist und die Alarmzeit erreicht, sendet die **AlarmClock** dem **Listener** Objekt eine Minute lang eine **alarm** Botschaft.
  - **alarmOn** bzw. **alarmOff** schaltet den Alarm ein oder aus.
- Entwerfen Sie pre und post Conditions sowie mögliche Klasseninvarianten gemäß dem Design-by-Contract Prinzip für alle Klassen der Klassenhierarchie. Verwenden Sie hierzu zunächst die Pseudo Syntax **precondition**, **postcondition** und **invariant**. Bei der Formulierung der Bedingungen halten Sie immer das Liskov Prinzip ein!
- Lösungshinweis: Denken Sie an die Zustände(!) der Uhr.**
- Im zweiten Teil der Aufgabe sollen Sie diese Klassenhierarchie mit Hilfe von Java realisieren. Dazu lesen Sie sich zunächst den Artikel <http://docs.oracle.com/javase/8/docs/technotes/guides/language/assert.html> durch. Implementieren Sie nun die gesamte Klassenhierarchie sowie die pre-, postcondition und invariant in Java gemäß dem vorgegeben Artikel.

- c. Im letzten Teil der Aufgabe überlegen Sie sich nun pre-, postconditions sowie invarianten für Ihr in Übungsblatt 1 redesigntes Spiel `VierGewinnt`. Sie müssen die Bedingungen nicht implementieren. Es reicht, wenn Sie diese in PseudoCode Syntax angeben.

## 2.2 Time HTML Konvertierung (Dependency Injection)

Gegeben sind die Klassen `TimeDisplay` und `DefaultTimeProvider` zur Konvertierung der aktuellen Zeit in HTML (siehe Server).

Implementieren Sie einen notwendigen JUnit Test, der testet, ob im Falle von Mitternacht der HTML String `"<span class=\"tinyBoldText\">Midnight</span>"` korrekt durch die Methode `getCurrentTimeAsHtmlFragment` zurückgeliefert wird.

Welche Problematik ergibt sich? Wie können Sie diese lösen? Führen Sie Refactoring Maßnahmen durch, beschreiben welche und begründen Sie, wieso Sie dies durchgeführt haben.