

## Übungsblatt 4

### Aufgabe 11 Eine leichtgewichtige DOM Implementierung

Nutzen Sie das *Composite Pattern* um ein HTML Dokument zu erstellen. Hierbei soll es folgende Arten von Knoten geben:

- **Element** stellt ein HTML Element dar (hat möglicherweise Kindelemente) und besitzt Eigenschaften wie **TagName** und **Parent**
- **TextNode** stellt einen Textknoten dar (hat keine Kindelemente) und verfügt über die Eigenschaft **Text**
- **Comment** stellt einen Kommentar dar (hat keine Kindelemente), der über die Eigenschaft **Text** gelesen oder gesetzt werden kann
- **Document** enthält max. ein HTML Element, auf das über die Eigenschaft **Root** zugegriffen werden kann

Alle Knoten erben von einer Klasse **Node** und besitzen eine Methode *ToHtml()*. Der Aufruf von *ToHtml()* einer Instanz von *Document* würde z.B. *ToHtml()* von *root*, falls vorhanden, aufrufen.

Fügen Sie eine API hinzu, so dass bei einem neuen Dokument der *Root* Knoten gesetzt werden kann. Außerdem soll es möglich sein zu jedem Knoten weitere Knoten hinzuzufügen.

### Aufgabe 12 Flyweight für Namensräume

Jedes HTML Element ist einem Namensraum geordnet. Zu jedem Namesraum existieren verschiedene Informationen, z.B.

- Präfix (z.B. xml)
- Url (z.B. <http://www.w3.org/XML/1998/namespace>)
- Version (z.B. 1.1)

Diese Informationen sollten nicht wiederholt werden. Statt dessen sollten die Informationen zu Namensräumen in einer *Flyweight Factory* gespeichert werden, welche dazu dient, vorhandene Räume auszulesen oder neue Räume aufzunehmen. Die Information soll dann in einem *Shared Flyweight Object* referenziert werden.

Zeichnen Sie ein UML Objektdiagramm. Dieses soll einen Zeitpunkt des Programms zeigen, bei dem drei HTML Elemente mit zwei Namespaces (html, svg) vorhanden sind.

### ! Wichtig

Alle Diagramme können über Anwendungen (z.B. PlantUML, yUML, Visual Studio, ...) oder auch per Hand gezeichnet werden.