

Übungen zur Vorlesung  
**Spezielle Gebiete zum Software Engineering**  
Sommersemester 2015

**Aufgabenblatt 2: WebComponents, WebWorkers, WebStorage,  
WebSockets**

Erstellen Sie für jede der folgenden Teilaufgaben eine Anwendung (=index.html), die die Verwendung der Web-Komponente demonstriert. Benutzen Sie bei den Aufgaben, bei denen das Polymer-Framework verwendet wird, die von Polymer zur Verfügung gestellten Core-Elemente wie z.B. `<core-ajax>` und Paper-Elemente wie z.B. `<paper-button>`

**Aufgabe 2.1: WebComponents mit „Vanilla“ Javascript (2 Punkte)**

Erstellen Sie nur mit Javascript ohne zusätzliches Framework eine Web-Komponente, die einen als Parameter übergebenen Text in umgekehrter Reihenfolge ausgibt. Des Weiteren soll die Komponente feststellen, ob es sich bei dem Text um ein Palindrom handelt. Das Element soll wie folgt zu benutzen sein:

```
<palindrom-check>Lagerregal</palindrom-check>
```

**Aufgabe 2.2: WebComponents mit Polymer (2 Punkte)**

Installieren Sie Polymer ([www.polymer-project.org](http://www.polymer-project.org)) und erstellen Sie mit Hilfe von Polymer die geforderte Funktionalität der Web-Komponente aus Aufgabe 2.1.

**Aufgabe 2.3: WebComponents und REST-APIs (2 Punkte)**

*OpenThesaurus* ist ein deutschsprachiges Wörterbuch für Synonyme und Assoziationen, für das eine programmierbare API existiert, für die unter

```
http://www.openthesaurus.de/about/api
```

weitere Informationen zu finden sind. Erstellen Sie eine Web-Komponente, die zu einem übergebenen Parameter (als Attribut zu dem Custom-Element) die Synonyme mit Hilfe von *OpenThesaurus* ermittelt und auflistet. Beispiel:

```
<synonym-list word="Hunger"></synonym-list>
```

#### Aufgabe 2.4: Komposition von WebComponents (4 Punkte)

Erstellen Sie mit Polymer jeweils eine Web-Komponente, die die nachfolgenden Funktionalitäten implementieren:

a) Ermittlung von Längengrad und Breitengrad einer als Parameter übergebenen Stadt über die folgende API

```
http://maps.googleapis.com/maps/api/geocode/json?address=<stadt>
```

Beispiel-URL für Minden:

```
http://maps.googleapis.com/maps/api/geocode/json?address=minden
```

Beispiel-Verwendung:

```
<longlat-element>Minden</longlat-element>
```

b) Erstellung einer statischen Landkarte für den als Parameter übergebenen Längengrad und Breitengrad, Zoomfaktor, Höhe und Breite über die folgende API

```
http://maps.googleapis.com/maps/api/staticmap?center=<lat>,<long>&zoom=<zoom>&size=<height>x<width>
```

Beispiel-URL für Minden:

```
http://maps.googleapis.com/maps/api/staticmap?center=52.29,8.89&zoom=10&size=400x300
```

Beispiel-Verwendung:

```
<static-map lat="52.29" long="8.89" zoom="10" height="400" width="300"></static-map>
```

c) Darstellung des aktuellen Wetters und der Vorhersage für den als Parameter übergebenen Längengrad und Breitengrad über die API, die unter

```
http://openweathermap.org/api
```

beschrieben ist. Beispiel-Verwendung:

```
<weather-element lat="52.29" long="8.89"></weather-element>
```

d) Kombinieren Sie nun alle Komponenten aus den vorhergehenden Aufgaben zu einer neuen CityInfo-Webkomponente. Implementieren Sie dazu eine neue Web-Komponente mit dem Custom-Element `<cityinfo>`. Der Benutzer soll in der Lage sein, den Namen einer Stadt in einem Eingabefeld einzugeben, der der neuen Webkomponente als Attribut oder als Textcontent übergeben wird. Die CityInfo-Komponente enthält für den Stadtnamen den Palindrom-Check (Aufgabe 2.2), die Synonyme (Aufgabe 2.3), Ermittlung des Längengrad und Breitengrad aus Teilaufgabe 2.4.a, eine Landkarte aus Teilaufgabe 2.4.b und schließlich die Wetterinformationen aus Teilaufgabe 2.4.c.