

# Praktikum Datenbanken und Webtechnologien

## Aufgabenblatt Nr. 2: Dart Teil I

Prof. Dr. Nane Kratzke

### 1 Hausaufgabe: Installieren Sie sich den Dart Editor (10 Punkte)

- Installieren Sie sich den Dart Editor. Beweisen Sie dies, indem sie zu diesem Praktikum ihren Laptop mitbringen. **(5 Punkte)**
- Implementieren Sie bitte eine Command Line Application "Hello World" in Dart. **(5 Punkte)**

### 2 Hausaufgabe: Listen funktional verarbeiten mit Dart (20 Punkte)

Gegeben sei eine Liste von ganzzahligen zufälligen Werten:

```
1 import 'dart:math';
2 main() {
3   final Random r = new Random();
4   final values = new Iterable.generate(1000, (int i) => r.nextInt(i + 1)).toList();
5 }
```

Die folgenden Fragen dürfen Sie nur beantworten indem sie die Funktion `map()`, `where()`, `reduce()`, `length`, `join()` sowie alle in Dart verfügbaren Operatoren nutzen.

- Geben Sie an wieviele durch 7 teilbare Werte in der Liste `values` existieren **(5 Punkte)**.
- Bestimmen Sie den Durchschnittswert der Liste `values` **(5 Punkte)**.
- Berechnen Sie den durchschnittlichen Wert aller ungeraden Werte der Liste `values` **(5 Punkte)**.
- Geben Sie alle Vielfachen von 138 aus der Liste `values` in einer HTML Liste auf der Konsole aus **(5 Punkte)**.

**Hinweis:** Die Wirkungsweise der Listenmethoden `map()`, `where()`, `reduce()`, `length`, `join()` ebenso wie für `Random` und `Iterable.new` finden sie sicherlich in der API von Dart (<https://api.dartlang.org>).

### 3 Hausaufgabe: HTML Tabellen erzeugen mit Dart (20 Punkte + 10 Extra-Punkte)

Gegeben ist folgende Dart Datenstruktur. Gesucht ist eine Funktion `String toHTML(List<List<String>>)`,

```
1 void main() {  
2   final table = [  
3     ["1", "2", "3"],  
4     ["4", "5", "6"],  
5     ["7", "8", "9"]  
6   ];  
7   print(toHTML(table));  
8 }
```

die folgende Ausgabe auf der Konsole erzeugt (und auch für beliebig andere Datenstrukturen oben angegebener Art funktioniert).

```
1 <table>  
2 <tr><td>1</td><td>2</td><td>3</td></tr>  
3 <tr><td>4</td><td>5</td><td>6</td></tr>  
4 <tr><td>7</td><td>8</td><td>9</td></tr>  
5 </table>
```

- Sie lösen das Problem mittels klassischen Kontrollstrukturen (wie bspw. for Schleifen). **(15 Punkte)**
- Sie nutzen ergänzend Methoden wie `map()` und `forEach()` sowie Lambdafunktionen **(5 Punkte)**
- Finden Sie eine Variante in der Sie nur noch Immutables in `toHTML` nutzen **(10 Extra-Punkte)**

## 4 Praktikumsaufgabe: Implementieren Sie das Sieb von Eratosthenes (50 Punkte)

Bestimmen Sie die Primzahlen bis 1000 nach dem Prinzip des Siebs von Eratosthenes

[https://de.wikipedia.org/wiki/Sieb\\_des\\_Eratosthenes](https://de.wikipedia.org/wiki/Sieb_des_Eratosthenes)

Folgendes Programm

```
1 main() {  
2   print("Die Primzahlen von 1 bis 1000 lauten: ${prims(1000)}");  
3 }
```

soll folgende Ausgabe erzeugen:

```
1 Die Primzahlen von 1 bis 1000 lauten: [2, 3, 5, 7, 11, 13, 17, 19, ..., 997]
```

**Hinweis:** Ihnen ggf. fehlende mathematische Funktionen wie `floor` und `sqrt` finden sie sicherlich in der API von Dart (<https://api.dartlang.org>).

Sie erhalten Punkte wie folgt:

- Sie bestimmen alle Primzahlen nach der oben angegebenen Methode mittels einer **typisierten** Methode `prims` (**volle 50 Punkte**).
- Sie bestimmen alle Primzahlen nach der oben angegebenen Methode aber sie kapseln diese Funktionalität nicht in einer Methode `prims` (**40 Punkte**).
- Sie bestimmen Primzahlen nach der oben angegebenen Methode, aber es sind zu wenige (**nur noch 35 Punkte**).
- Sie bestimmen Primzahlen nach der oben angegebenen Methode, aber es sind zu viele (**nur noch 25 Punkte**, da einige Primzahlen dann keine Primzahlen sein können).
- Sie können das Sieb von Erathosthenes nicht implementieren, aber sie schaffen es zumindest für eine gegebene Zahl `n` mittels einer Methode `bool isPrim(int)` zu prüfen, ob es sich bei `n` um eine Primzahl handelt (**nur noch 15 Punkte**).