

Blockwoche: Web Programming Lab

Übung 1: Interface

```
class App {  
    createUser(name: string): User {  
        return {  
            id: 123,  
            name: name,  
            settings: {  
                settingA: 123,  
                settingB: 456,  
            }  
        }  
    }  
}
```

- Erstellt für «User» und «Settings» Interfaces
- Branch in GitHub Repo: interface

- Mögliche Lösung:

<https://gist.github.com/anschuermann/34003f5c02a3289f29a66ec2b1ea14a7>

Übung 2: Generics

```
interface KeyPair {  
    key: string;  
    value: string;  
}  
  
let kv1: KeyPair = { key:"key", value:"value" };  
let kv2: KeyPair = { key:1, value:"Steve" };  
let kv3: KeyPair = { key:1, value:12345 };
```

- Korrigiere die Kompilierungsfehler mit Generics
- Branch in GitHub Repo: generics
- Mögliche Lösung:
<https://gist.github.com/anschuermann/e7f2a5d53442effcaea9788928193d22>

Übung 3: Two-for-one

- Erhalte einen Namen als Input und gib folgendes auf der Konsole aus (X ist der Name):

«One for X, one for me.»

- Wenn der Name fehlt, gib folgendes aus:

«One for you, one for me.»

- Branch in GitHub Repo: two-for-one

- Mögliche Lösung:

<https://gist.github.com/anschuermann/6e851609ef851f91c0bf9c9239c1f747>

Übung 4: Charles

Charles gibt immer dieselben Antworten. Erhalte eine Message an Charles als Input und gib seine Antwort auf der Konsole aus:

- Charles antwortet 'Sure.' wenn Du ihm eine Frage stellst.
- Er antwortet 'Whoa, chill out!' wenn Du ihn anschreist.
- Er sagt 'Fine. Be that way!' wenn Du ihn in deiner Nachricht anschweigst.
- Er antwortet 'Whatever.' zu allem anderen.
- Branch in GitHub Repo: charles
- Mögliche Lösung:

<https://gist.github.com/anschuermann/281d2e928d55d7fcb7e40fa3af667257>

Übung 5: Classlist

Erhalte Informationen zu Studenten und ihren jeweiligen Klassen und erstelle daraus eine Klassenliste. Implementiere folgende Funktionen:

- Student hinzufügen (Name und Klasse)
- Liste aller Studenten einer Klasse ausgeben
- Sortierte Liste aller Studenten in allen Klassen ausgeben. Klassen starten mit 1, 2, 3 etc., die Studenten sind alphabetisch nach Name sortiert.
- Branch in GitHub Repo: classlist
- Mögliche Lösung:
<https://gist.github.com/anschuermann/a2b42ee5db983b311abaca5fce017b9c>

Zusatz-Übung: Math-Bot

Erstelle einen Chat-Bot in TypeScript, der Math-Aufgaben aus Textanfragen lösen kann:

- Beispiel: «Was ist 7 minus 5?» \rightarrow 2
- Unterstützte Operationen: Addition, Subtraktion, Multiplikation, Division
- Der Chat-Bot weist Anfragen zurück, die er nicht beantworten kann
- Branch in GitHub Repo: mathbot

- Mögliche Lösung:

<https://gist.github.com/anschuermann/5942c16c787dad8daa1b5d85de50eafc>