

# Praktikumsaufgaben für VW6

Programmierung interaktiver Systeme im Wintersemester 2023/24  
Prof. Dr. Martin Weigel

[README](#) • [JavaDoc](#) • [JUnit Tests](#)

Wählen Sie zum Bearbeiten dieser Aufgabe entweder ihr MVC-Projekt aus Po3 **oder** Po4 – ihre MVC Version des TicTacToe-Spieles oder ihre MVC Version von PokePong. Das TicTacToe-Spiel ist kurzfristig etwas weniger Arbeit. Das Pong-Spiel wird aber einen wichtigen Teil in der übernächsten Praktikumsübung spielen. Der Mehraufwand kann sich also lohnen.

Versuchen Sie alle Aufgabenteile zu bearbeiten. Eine teilweise dokumentierte und teilweise getestete Abgabe zählt in diesem Praktikum mehr als eine vollständige Dokumentation ohne Tests. Ziel der Praktika ist es schließlich, dass Sie sich mit allen für das Projekt benötigten Konzepten vertraut machen.

## 1 README

---

Schreiben Sie eine `README.md` Datei für das Programm mit Markdown. Versuchen Sie dabei möglichst viele der Checkboxes für das Programmierprojekt zu erfüllen (Slide 7). Ihre README sollte daher folgendes enthalten:

1. Eine Kurzbeschreibung der Anwendung
2. Einen oder mehrere Screenshots
3. Alle verwendeten Bibliotheken als Link
4. Eine Beschreibung wie das Programm zu starten ist
5. Ein Beispiel wie das Model in der JShell getestet werden kann: Erstellen eines Objektes, Aufrufen einer Funktion und Prüfung, dass sich die Daten im Model geändert haben.

## 2 JavaDoc

---

JavaDoc ist ein Werkzeug zum Dokumentieren von Java-Code. Dokumentieren Sie das von ihnen gewählte Spiel mit JavaDoc. Nutzen Sie hierzu die entsprechenden Tags von JavaDoc, um ihre Klassen und Methoden vollständig zu dokumentieren.

Dokumentieren Sie alle Pakete, Interfaces und Klassen. Dokumentieren Sie dafür auch alle Variablen und Methoden welche außerhalb der Klasse sichtbar sind (`public`, `protected` und ohne Zugriffsrecht `Modifizierer`). Auch das Dokumentieren von `private`-Elementen kann sinnvoll sein, um sich selbst oder andere Entwickler auf einen bestimmten Aspekt hinzuweisen und so den Code einfacher verständlich zu machen.

Generieren Sie sich die HTML Dokumentation mithilfe der Kommandozeile oder mit IntelliJ. Bringen Sie ihren dokumentierten Quellcode und die HTML Dokumentation zu ihrem Praktikumstermin mit.

## 3 JUnit Tests

---

Unit Tests spielen eine wichtige Rolle beim Testen von Software-Anwendungen. Eine *Unit* steht dabei für eine Verhaltenseinheit (z.B. eine bestimmte Funktion welche eine Klasse/Methode erfüllen soll). Nutzen Sie das gleiche Projekt wie in den obigen Aufgaben und schreiben Sie Tests für das *Model* der Anwendung. Controller und View brauchen bei uns nicht getestet werden. Überlegen Sie sich im Vorfeld, welches Verhalten Sie testen müssen. Dokumentieren Sie häufig vorkommendes Verhalten und auch ein paar Sonderfälle. Versuchen Sie eine hohe Line Coverage ( $\geq 90\%$ ) des Model-Codes zu erreichen.



In ihrem späteren Projekt wird die Test-Coverage in die Bewertung einfließen. Die Test-Coverage gibt an wie viele ihrer Programmierzeilen von den Unit-Tests

durchlaufen werden (s. Vorlesung). Sie können sich diese Coverage in IntelliJ mithilfe von Run '...' with Coverage berechnen lassen.