

**Modul 63081: Grundpraktikum Programmierung**

durchgeführt vom  
Lehrgebiet Kommunikationsnetze  
im Wintersemester 2025/26

# Aufgabenstellung



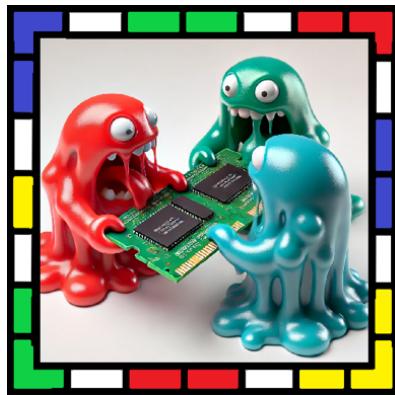
## Inhaltsverzeichnis

<b>1 Programmierpraktikum Wintersemester 2025/26</b>	<b>3</b>
1.1 Terminplan . . . . .	4
1.2 Betreuung . . . . .	5
<b>2 Abgabe und Bewertung Ihres Programms</b>	<b>6</b>
2.1 Namensschema . . . . .	9
<b>3 Dokumentationsrichtlinien</b>	<b>10</b>
3.1 Einführung in das Programm (PDF-Dokumentation) . . . . .	10
3.2 Javadoc-Dokumentation . . . . .	10
3.3 Kommentierung des Programmcodes . . . . .	11

## 1 Programmierpraktikum Wintersemester 2025/26

Herzlich willkommen zum Grundpraktikum Programmierung (Modul 63081) im Wintersemester 2025/26.

Das Praktikum wird in diesem Semester vom Lehrgebiet Kommunikationsnetze von Prof. Dr.-Ing. habil. Dr. h.c. Herwig Unger durchgeführt. Die weiteren Betreuer sind Timothy Harrison, Marcel Schaible und Andrea Frank.



Thema wird das Spiel RINGwars sein. RINGwars ist ein Strategiespiel für zwei Spieler, die auf einem Ring von Knoten gegeneinander antreten. Ziel des Spiels ist, alle Knoten des Rings zu erobern oder den Gegner zu eliminieren.

Ihre Aufgabe wird es sein, einen Agenten für das Spiel RINGwars zu schreiben, der mit der zur Verfügung gestellten RINGwars GUI interagiert. Für die Entwicklung werden Sie die plattformunabhängige Programmiersprache Java verwenden.

Die genauen Anforderungen der Aufgabe entnehmen Sie bitte dem Dokument „RINGwars – A Code-able Game“, das Sie in Moodle finden. Dieses Dokument ist sowohl in englischer als auch in deutscher Sprache verfügbar.

Für die Umsetzung der Aufgabe haben Sie gut drei Monate Zeit. Aber da die Zeit nie so schnell verrinnt wie beim Programmieren und Testen, empfehlen wir Ihnen, zügig mit der Bearbeitung zu beginnen.

Wir wünschen Ihnen viel Erfolg und Freude bei der Arbeit!

## 1.1 Terminplan

- **19. September 2025: Ausgabe der Programmieraufgabe**

- **1. Oktober 2025: letzte Möglichkeit für eine Abmeldung**

Abmeldungen müssen schriftlich an das Prüfungsamt gerichtet werden. Eine Abmeldung per Mail ist ausreichend ([pruefungsamt.mathinf@fernuni-hagen.de](mailto:pruefungsamt.mathinf@fernuni-hagen.de)).

- **1. Oktober 2025: allgemeiner Arbeitsbeginn**

- **2. Oktober 2025, 15.00 Uhr: Eröffnungsveranstaltung in Zoom**

An diesem Termin werden wir ein Video-Treffen veranstalten, in dessen eher informellen Rahmen werden wir gemeinsam mit Ihnen Fragen zur Aufgabenstellung, etc. besprechen können. Die Teilnahme an diesem Treffen ist freiwillig. Den Zoom-Link können Sie der Moodle-Plattform entnehmen.

- **17. Oktober 2025,**

**7. November 2025,**

**21. November 2025 und**

- **19. Dezember 2025, jeweils 18.00 Uhr: Online-Sprechstunden in Zoom**

In den Online-Sprechstunden haben Sie die Möglichkeit, Ihre aufgetretenen Fragen und Probleme mit den Betreuern zu besprechen. Die Teilnahme an all diesen Treffen ist freiwillig. Den Zoom-Link können Sie der Moodle-Plattform entnehmen.

- **4. Januar 2026, 23.59 Uhr: Abgabe Ihrer Lösung**

Ihre Lösung (Programm samt Dokumentation) muss bis zu diesem Zeitpunkt bei uns eingegangen sein.

- **2. bis 12. Februar 2026: virtuelle Präsenzphase in Zoom**

Innerhalb dieses Zeitraums findet die virtuelle Präsenzphase in Form von Videokonferenzen statt. Die Gespräche werden in Gruppen á 10 Teilnehmenden und den Betreuern durchgeführt. Vorab werden wir dazu einen individuellen Termin mit Ihnen vereinbaren. Planen Sie bitte ½ Tag für diesen Termin ein. Die Teilnahme an der virtuellen Präsenzphase ist Pflicht.

- **13. Februar 2026, 14.00 Uhr: Abschlussveranstaltung im Zoom**

Die Teilnahme an der Abschlussveranstaltung ist nicht verpflichtend.

Weitere Informationen zu den Abgabe- und Bewertungsmodalitäten finden Sie in Kapitel 2.

## 1.2 Betreuung

Während des Semesters stehen Ihnen Herr Timothy Harrison, Herr Marcel Schaible und Frau Andrea Frank als Betreuer in fachlichen Fragen zur Seite. Für alles Organisatorische und bei sonstigen Problemen können Sie Frau Frank ansprechen.

**Moodle** Die Moodle Lernumgebung des Kurses ist während des gesamten Semesters Ihr zentraler Anlaufpunkt für alle aktuellen Information rund um das Programmierpraktikum:

<https://moodle.fernuni-hagen.de/course/view.php?id=4122>

Hier finden Sie u.a. die unten erwähnten Foren sowie den aktuellen Terminplan.

**Foren** Die Betreuung des Praktikums erfolgt überwiegend über die folgenden zwei Moodle Foren:

- **Ankündigungs-Forum:** Hier werden wir Ankündigungen und allgemeine Informationen wie z.B. Änderungen des organisatorischen Ablaufs bekannt geben. Bitte schauen Sie mindestens einmal pro Woche in dieses Forum, um sich über aktuelle Mitteilungen zu informieren.
- **Diskussions-Forum:** Wenn Sie inhaltliche oder organisatorische Fragen zum Praktikum haben oder anderen Studierenden eine Hilfestellung geben möchten, so gehen Sie bitte in dieses Forum. Hier können alle praktikumsbezogenen Probleme diskutiert werden.

Ihre Mitarbeit in dem Diskussions-Forum ist wichtig. In den vergangenen Jahren fand hier immer wieder ein intensiver Austausch statt, von dem alle Beteiligten profitieren konnten.

**Mail** Die Praktikumsbetreuung ist über die folgende E-Mail-Adresse zu erreichen:

[propra@fernuni-hagen.de](mailto:propra@fernuni-hagen.de)

Fragen, die programmtechnischer Natur sind, werden allerdings nicht per E-Mail beantwortet. Bitte nutzen Sie hierfür das Diskussions-Forum in Moodle. Meistens haben mehrere Teilnehmer das gleiche Problem und so können sich alle an der Diskussion beteiligen und Hilfe bekommen.

Bitte stellen Sie Ihre Fragen bei Problemen, Unklarheiten, etc. frühzeitig(!) - vor allem vor dem Abgabetermin. Dies ist besser als eine ungültige Abgabe zu riskieren.

## 2 Abgabe und Bewertung Ihres Programms

Im Rahmen dieses Programmierpraktikums müssen Sie folgende Anforderungen erfüllen:

1. Sie müssen eigenständig ein Programm implementieren, welches alle Anforderungen erfüllt, die sich aus der Aufgabenstellung ergeben. Gruppenlösungen sind nicht zulässig.
2. Sie müssen ein Java-Projekt erstellen und dabei folgende Punkte berücksichtigen:

- Benennen Sie Ihr Java-Projekt exakt wie in Kapitel 2.1 beschrieben nach dem Schema:

RINGwars\_<Matr-Nr>\_<Nachname>\_<Vorname>

- Die Quellcodedatei mit der **main**-Methode muss ebenfalls diesem Namensschema entsprechen.
- Sie dürfen lediglich eigene Klassen und die Java-Standardbibliothek verwenden.
- Erzeugen Sie eine ausführbare JAR-Datei, die sich durch folgenden Aufruf starten lässt:

java -jar RINGwars\_<Matr-Nr>\_<Nachname>\_<Vorname>.jar <Step#> <Agent Name>

- Legen Sie Dokumentation und Javadoc wie in Kapitel 3 beschrieben ab.
- Fügen Sie die unterschriebene Selbstständigkeitserklärung bei.
- Packen Sie das gesamte Projekt als ZIP-Datei, benannt nach demselben Schema.

3. Sowohl das Programm selbst (Code und Kommentare) als auch die Dokumentation (PDF und Javadoc) können wahlweise in Deutsch oder Englisch verfasst werden.

### 4. Mindestanforderungen

Damit Ihre Abgabe als gültig anerkannt und zur Bewertung zugelassen wird, muss Ihr Programm mindestens die folgenden Anforderungen erfüllen:

#### a) Ausführbarkeit

- Das Programm muss ohne zusätzliche Bibliotheken oder externe Abhängigkeiten lauffähig sein.
- Es muss durch den Aufruf

java -jar RINGwars\_<Matr-Nr>\_<Nachname>\_<Vorname>.jar <Step#> <Agent Name>

gestartet werden können.

- Das Programm darf keine interaktive Benutzereingabe verlangen.
- b) Einlesen der Status-Datei
  - Der Agent muss die von der RINGwars-GUI bereitgestellte Status-Datei `<step#>.txt` korrekt einlesen können.
  - Mindestens folgende Informationen müssen verarbeitet werden:
    - aktuelle FERNIE-Zahlen (erste Zeile),
    - Knotenzugehörigkeiten (zweite Zeile),
    - neue verfügbare FERNIES (dritte Zeile),
    - maximale Knotenkapazität (vierte Zeile).
  - Ungültige oder unvollständige Status-Dateien dürfen nicht zum Absturz führen.
- c) Erzeugung einer gültigen Zug-Datei
  - Der Agent muss in jedem Spielzug eine Ausgabedatei `move.txt` im richtigen Format erzeugen:
    - jede Zeile beschreibt eine Aktion mit Knotennummer und Anzahl FERNIES,
    - positive Zahlen = Platzieren (Place),
    - negative Zahlen = Entfernen (Remove),
    - die letzte Zeile muss eine gültige Zuganweisung sein, d.h. keine Leerzeile.
  - Alle Züge müssen innerhalb der Spielregeln legal sein:
    - nur FERNIES von eigenen Knoten dürfen entfernt werden,
    - es dürfen nicht mehr FERNIES entfernt werden, als auf dem Knoten vorhanden sind,
    - es dürfen nicht mehr Fernies platziert werden, als durch „neue FERNIES“ plus entfernte FERNIES verfügbar sind,
    - Fernies dürfen nur auf Knoten innerhalb der Sichtweite gesetzt werden.
- d) Stabilität
  - Das Programm muss ohne unbehandelte Exceptions laufen.
  - Auch in den ersten Spielzügen (bevor der Gegner sichtbar ist) muss der Agent gültige Ausgaben erzeugen.

- Bei unerwarteten Eingaben darf das Programm nicht abstürzen, sondern muss zumindest eine leere `move.txt` erzeugen.

e) Dokumentation

- Eine kurze PDF-Einführung (**ca. 3 - 15 Seiten**) muss vorhanden sein, die Aufbau, Datenstrukturen und grundlegende Strategie des Agenten erläutert.
- Javadoc-Dokumentation für alle nicht-privaten Klassen und Methoden muss vorliegen.
- Der Quellcode muss ausreichend kommentiert sein.

f) Integrität

- Der Agent darf ausschließlich in seinem eigenen Verzeichnis Dateien lesen oder schreiben.
- Jegliche Manipulation von Dateien anderer Agenten oder der GUI gilt als Betrugsvorwurf und führt zum Nichtbestehen.

5. Die Abgabe Ihrer Lösung muss bis zum **4. Januar 2026** erfolgen.

6. Während der virtuellen Präsenzphase sollen Sie den Aufbau Ihres Programms erklären und Fragen der Betreuenden beantworten können.

*Manipulationsversuche mit den eingesandten Programmen, z.B. durch Aufspielen von Computerviren oder anderen Programmkomponenten, die nicht der Lösung der Programmieraufgabe dienen, führen – unabhängig von Schadensersatzansprüchen seitens der FernUniversität – zu einer nichtbestandenen Prüfungsleistung. Das Kopieren von Sourcecode aus ähnlichen Projekten aus dem Internet oder von Kommiliton:innen ist unter keinen Umständen gestattet und wird mit einem sofortigen Ausschluss geahndet. Explizit verboten ist außerdem der Einsatz von KI-gestützten Werkzeugen wie beispielsweise ChatGPT oder Copilot. Eine Verwendung solcher Werkzeuge führt ebenfalls zum sofortigen Ausschluss und somit einer nichtbestandenen Prüfungsleistung. Wir besitzen sowohl die entsprechenden Werkzeuge, als auch die notwendige Sachkenntnis, um solche Verstöße zu erkennen.*

## 2.1 Namensschema

Der Name Ihres Java-Projekts muss dem folgenden Schema entsprechen:

RINGwars\_<Matr-Nr>\_<Nachname>\_<Vorname>

Die Platzhalter <Matr-Nr>, <Nachname> und <Vorname> sind entsprechend zu ersetzen, wobei folgende Punkte zu beachten sind:

- Die Matrikelnummer soll nur aus Ziffern bestehen (also nicht das q enthalten).
- Wenn mehrere Vornamen oder Doppelnamen angegeben werden, sollen diese mit einem Unterstrich   getrennt werden. Der gesamte Name darf also kein Leerzeichen und keinen Bindestrich - enthalten.
- Der Nachname und der Vorname sollen mit Großbuchstaben beginnen.
- Mit Ausnahme des Unterstrichs   sollen keine Sonderzeichen, Umlaute, Akzentzeichen, etc. verwendet werden (also nur die 26 Buchstaben des Alphabets, ä wird wie üblich durch ae ersetzt, usw.).
- Gültige Projektnamen wären z.B.:
  - RINGwars\_1234567\_Mueller\_Rene\_Joerg
  - RINGwars\_9876543\_Meier\_Schmidt\_Francois
  - RINGwars\_815123\_Von\_Wohlgemuth\_Kaethe\_Luise

Die **Quellcodedatei**, welche die main-Methode zum Starten Ihres Programms enthält, muss ebenfalls dem obigen Namensschema entsprechen, also:

RINGwars\_<Matr-Nr>\_<Nachname>\_<Vorname>.java

Und auch die JAR- und ZIP-Dateien sind nach demselben Namensschema zu benennen:

RINGwars\_<Matr-Nr>\_<Nachname>\_<Vorname>.jar

RINGwars\_<Matr-Nr>\_<Nachname>\_<Vorname>.zip

Wir empfehlen Ihnen, auch bei allen weiteren Dateinamen innerhalb Ihres Projekts möglichst auf die Verwendung von potentiell problematischen Sonderzeichen zu verzichten. Insbesondere dürfen Dateinamen keine Sonderzeichen wie z.B. < und > enthalten, die zwar unter Linux, aber nicht unter Windows zulässig sind.

### 3 Dokumentationsrichtlinien

Zu einem Software-Projekt gehört immer auch eine Dokumentation. Daher ist auch die Dokumentation Ihres Projekts ein Bestandteil der Prüfungsleistung und fließt mit in die Gesamtnote ein. Neben dem eigentlichen Programm werden wir uns also auch ausführlich Ihrer Dokumentation widmen. Sie sollten daher bei der Gestaltung Ihrer Programmdokumentation die gleichen Maßstäbe ansetzen, nach denen Sie beispielsweise eine schriftliche Seminarausarbeitung erstellen würden. Dennoch ist die Dokumentation nur einer von mehreren Bestandteilen Ihrer Gesamtleistung, das ausführbare Programm bildet weiterhin den Hauptteil. Orientieren Sie sich daher an der folgenden Empfehlung: Wenden Sie die Zeit und Mühe auf, die für eine Dokumentation nach den im Folgenden beschriebenen Richtlinien erforderlich ist; aber verlieren Sie dabei nicht aus dem Auge, dass die Dokumentation nicht den Kern des Projekts darstellt.

Ihre Dokumentation besteht aus drei Teilen: einer kurzen Einführung in das Programm (PDF-Dokumentation), der Javadoc-Dokumentation und dem kommentierten Programmcode.

#### 3.1 Einführung in das Programm (PDF-Dokumentation)

Beschreiben Sie den Aufbau Ihres Programms (welches sind die wichtigsten benutzten Datenstrukturen und welche Klassen erledigen welche Aufgaben) und begründen Sie Ihre Designentscheidungen. Erläutern Sie auch die grundlegende Strategie Ihres Agenten. Gehen Sie an dieser Stelle nicht auf Implementierungsdetails ein. Ihre Einführung sollte beispielsweise auch von einem Leser nachvollziehbar sein, der zwar grundlegende Programmiererfahrungen besitzt, aber die Programmiersprache Java nicht beherrscht.

Ihre Programmdokumentation legen Sie bitte als PDF-Datei in einem Unterordner des Java-Projekts ab. Dieser Ordner soll den Namen **documentation** tragen.

Die gesamte PDF-Dokumentation kann wahlweise in Deutsch oder Englisch geschrieben werden und sollte mindestens 3 und maximal 15 DIN-A4-Seiten umfassen.

#### 3.2 Javadoc-Dokumentation

Sämtliche nichtprivate Elemente (Klassen, Methoden, Attribute, ...) sind über Javadoc zu dokumentieren (private Elemente benötigen nicht zwingend einen Kommentar; manchmal kann dies jedoch praktisch sein). Erläutern Sie in je zwei bis drei prägnanten Sätzen, welche Aufgabe von der betreffenden Klasse bzw. Methode erfüllt wird, sowie die Einordnung der Klasse/Methode in den Gesamtkontext des Programms. Denken Sie daran, bei Methoden auch die Parameter und Rückgabewerte zu beschreiben. Die Kommentare können wahlweise in Deutsch oder in Englisch verfasst werden.

Zur Erstellung der Javadoc-Dokumentation benutzen Sie das Javadoc-Werkzeug und erzeugen Sie Javadoc für alle Elemente (Sichtbarkeit „package“) in einem Unterordner des Java-Projekts mit dem Namen `doc`. Diese generierte HTML-Dokumentation ist ebenfalls abzugeben.

In Eclipse können Sie Javadoc mittels des Befehls `Project → Generate Javadoc...` für Ihr Projekt generieren lassen. Falls in den erzeugten HTML-Dateien Umlaute falsch dargestellt werden, sollten Sie folgende Optionen bei der Javadoc Generierung mit angeben (im letzten Schritt in das Feld *Extra Java Options*):

`-encoding UTF-8 -charset UTF-8 -docencoding UTF-8`

### **3.3 Kommentierung des Programmcodes**

Kommentieren Sie zudem auch Ihren Programmcode (wahlweise in Deutsch oder Englisch). Das soll nicht heißen, dass Sie zu jeder Anweisung einen Kommentar schreiben müssen, aber Ihr Programm muss mit Hilfe der Kommentare soweit verständlich sein, dass Leser Ihre Lösung ohne ein langwieriges Hineindenken in Ihre Java-Konstrukte nachvollziehen können.