

# Organisatorisches

Stefan Riezler, Michael Staniek

Computerlinguistik  
Universität Heidelberg  
Sommersemester 2023



# Überblick

## Herzlich Willkommen zur Begleitveranstaltung zum Softwareprojekt

- Voraussetzungen:

Erfolgreiche Teilnahme an ICL, PII, FF-SM  
Bestehen der Orientierungsprüfung

- Dauer: 1 Semester
- 6 LP → ca. 12h/Woche + 4 LP ÜK

# Überblick

Herzlich Willkommen zur Begleitveranstaltung zum Softwareprojekt

- Voraussetzungen:
  - Erfolgreiche Teilnahme an ICL, PII, FF-SM
  - Bestehen der Orientierungsprüfung
- Dauer: 1 Semester
- 6 LP → ca. 12h/Woche + 4 LP ÜK

# Überblick

Herzlich Willkommen zur Begleitveranstaltung zum Softwareprojekt

- Voraussetzungen:
  - Erfolgreiche Teilnahme an ICL, PII, FF-SM
  - Bestehen der Orientierungsprüfung
- Dauer: 1 Semester
- 6 LP → ca. 12h/Woche + 4 LP ÜK

# Überblick

Herzlich Willkommen zur Begleitveranstaltung zum Softwareprojekt

- Voraussetzungen:
  - Erfolgreiche Teilnahme an ICL, PII, FF-SM
  - Bestehen der Orientierungsprüfung
- Dauer: 1 Semester
- 6 LP → ca. 12h/Woche + 4 LP ÜK

# Heute

- ① Inhalte & Ziele
- ② Semesterplan
- ③ Benotung
- ④ Themen

# Inhalte

Im Softwareprojekt soll eine computerlinguistische Aufgabenstellung

- ... weitgehend eigenverantwortlich und in Teamarbeit
- ... geplant,
- ... programmiert,
- ... getestet,
- ... dokumentiert und
- ... präsentiert werden.

# Ziele

Am Ende des Softwareprojekts könnt ihr...

- ① aus einem theoretischen Ansatz einen konkreten Projektplan erarbeiten.
- ② den Plan in Teamarbeit umsetzen.
- ③ eure Projektergebnisse präsentieren und analysieren



# Ziele

Am Ende des Softwareprojekts könnt ihr...

- ① aus einem theoretischen Ansatz einen konkreten Projektplan erarbeiten.
- ② den Plan in Teamarbeit umsetzen.
- ③ eure Projektergebnisse präsentieren und analysieren

# Ziele

Am Ende des Softwareprojekts könnt ihr...

- ① aus einem theoretischen Ansatz einen konkreten Projektplan erarbeiten.
  - den Ansatz in eigenen Worten wiedergeben.
  - zu implementierende Module festlegen und Abhängigkeiten zwischen den Modulen definieren.
  - Aus dem Modulplan einen Zeitplan und eine Aufgabenverteilung entwickeln.
  - Programmarchitektur, Datenstrukturen und Schnittstellen festlegen.
- ② den Plan in Teamarbeit umsetzen.
- ③ eure Projektergebnisse präsentieren und analysieren

# Ziele

Am Ende des Softwareprojekts könnt ihr...

- ① aus einem theoretischen Ansatz einen konkreten Projektplan erarbeiten.
- ② den Plan in Teamarbeit umsetzen.
- ③ eure Projektergebnisse präsentieren und analysieren

# Ziele

Am Ende des Softwareprojekts könnt ihr...

- ① aus einem theoretischen Ansatz einen konkreten Projektplan erarbeiten.
- ② den Plan in Teamarbeit umsetzen.
  - die euch zugewiesenen Module plangemäß implementieren,
  - eure Module so testen und dokumentieren, dass andere Teammitglieder sie verwenden können.
  - **gemeinsame** Instrumente entwickeln und einsetzen, die sicherstellen, dass ihr den Plan einhaltet, und dass ihr Fragen und Probleme rechtzeitig klärt.
- ③ eure Projektergebnisse präsentieren und analysieren

# Ziele

Am Ende des Softwareprojekts könnt ihr...

- ① aus einem theoretischen Ansatz einen konkreten Projektplan erarbeiten.
- ② den Plan in Teamarbeit umsetzen.
- ③ eure Projektergebnisse präsentieren und analysieren

# Ziele

Am Ende des Softwareprojekts könnt ihr...

- ① aus einem theoretischen Ansatz einen konkreten Projektplan erarbeiten.
- ② den Plan in Teamarbeit umsetzen.
- ③ eure Projektergebnisse präsentieren und analysieren
  - Ergebnisse und Methoden so präsentieren und demonstrieren, dass andere Teams verstehen, was ihr gemacht habt.
  - Schwächen eurer Implementierung und Probleme beim Projektablauf identifizieren und Verbesserungsvorschläge/Gründe angeben

## Teamarbeit

*“gemeinsame Instrumente entwickeln und einsetzen, die sicherstellen, dass ihr den Plan einhaltet, und dass ihr Fragen und Probleme rechtzeitig klärt.”*

- regelmäßige Treffen - in der Gruppe und mit dem Betreuer
- gemeinsames “Logbuch”
- gemeinsamer Terminkalender
- wöchentliche Statusreports aller Teilnehmer
- Codereviews
- Versionskontrolle

## Teamarbeit

*“gemeinsame Instrumente entwickeln und einsetzen, die sicherstellen, dass ihr den Plan einhaltet, und dass ihr Fragen und Probleme rechtzeitig klärt.”*

- regelmäßige Treffen - in der Gruppe und mit dem Betreuer
- gemeinsames “Logbuch”
- gemeinsamer Terminkalender
- wöchentliche Statusreports aller Teilnehmer
- Codereviews
- Versionskontrolle



# Semesterplan

`www.cl.uni-heidelberg.de/courses/ws24/  
softwareprojekt/material/`

# Projektanmeldung

Bitte schickt bis Montag, **21.10.**, eine Mail an `{riezler|staniek}@cl.uni-heidelberg.de` mit

- 1 Betreff: "SWP Projektanmeldung"
- 2 Projektpräferenz
- 3 Falls ihr schon im Vorfeld eine Gruppe bilden wollt, die Namen der anderen Gruppenmitglieder.

## Spezifikationsvortrag: 26.11.

- **Inhaltliche Spezifikation:**
  - Problemstellung, Lösungsansatz, Evaluation (→ *Forschungsplan*)
  - Auswahl/Sichtung benötigter Ressourcen und Verfahren/Algorithmen
- **Modularisierung und Aufgabenverteilung:**
  - Definition von Modulen/Aufgaben und Abhängigkeiten
  - Zuordnung von Namen zu Aufgaben
  - Zeitplan
- **konkrete Planung** von Programmarchitektur, Datenstrukturen, Schnittstellen
- *Vortrag im Plenum, maximal 25 Minuten*

## Spezifikationsvortrag: 26.11.

- **Inhaltliche Spezifikation:**
  - Problemstellung, Lösungsansatz, Evaluation (→ *Forschungsplan*)
  - Auswahl/Sichtung benötigter Ressourcen und Verfahren/Algorithmen
- **Modularisierung und Aufgabenverteilung:**
  - Definition von Modulen/Aufgaben und Abhängigkeiten
  - Zuordnung von Namen zu Aufgaben
  - Zeitplan
- **konkrete Planung** von Programmarchitektur, Datenstrukturen, Schnittstellen
- *Vortrag im Plenum, maximal 25 Minuten*

## Spezifikationsvortrag: 26.11.

- **Inhaltliche Spezifikation:**
  - Problemstellung, Lösungsansatz, Evaluation (→ *Forschungsplan*)
  - Auswahl/Sichtung benötigter Ressourcen und Verfahren/Algorithmen
- **Modularisierung und Aufgabenverteilung:**
  - Definition von Modulen/Aufgaben und Abhängigkeiten
  - Zuordnung von Namen zu Aufgaben
  - Zeitplan
- konkrete Planung von Programmarchitektur, Datenstrukturen, Schnittstellen
- *Vortrag im Plenum, maximal 25 Minuten*

## Spezifikationsvortrag: 26.11.

- **Inhaltliche Spezifikation:**
  - Problemstellung, Lösungsansatz, Evaluation (→ *Forschungsplan*)
  - Auswahl/Sichtung benötigter Ressourcen und Verfahren/Algorithmen
- **Modularisierung und Aufgabenverteilung:**
  - Definition von Modulen/Aufgaben und Abhängigkeiten
  - Zuordnung von Namen zu Aufgaben
  - Zeitplan
- **konkrete Planung** von Programmarchitektur, Datenstrukturen, Schnittstellen
- *Vortrag im Plenum, maximal 25 Minuten*

## Spezifikationsvortrag: 26.11.

- **Inhaltliche Spezifikation:**
  - Problemstellung, Lösungsansatz, Evaluation (→ *Forschungsplan*)
  - Auswahl/Sichtung benötigter Ressourcen und Verfahren/Algorithmen
- **Modularisierung und Aufgabenverteilung:**
  - Definition von Modulen/Aufgaben und Abhängigkeiten
  - Zuordnung von Namen zu Aufgaben
  - Zeitplan
- **konkrete Planung** von Programmarchitektur, Datenstrukturen, Schnittstellen
- *Vortrag im Plenum, maximal 25 Minuten*

# Statusmeetings

- Individuelle Termine für jede Gruppe mit Betreuer.
- Bei Bedarf wöchentlich zur Seminarzeit
- Keine Meetings außerhalb der Seminarzeit - *come prepared!*



## Abschlussvortrag: 28.1.

- Wie wurde der Lösungsansatz umgesetzt?
- Präsentation der Evaluationsergebnisse
- Demo
- *Lessons Learned*: Identifikation von Schwächen eurer Implementierung und Problemen beim Projektablauf → Verbesserungsvorschläge/ Gründe angeben.
- *Vortrag im Plenum, maximal 25 Minuten*

# Leistungsnachweis

## Abgabe: 4.2.

### Voraussetzung für einen Schein:

- Teilnahme an allen gemeinsamen Veranstaltungen (Plenum)
- Spezifikationsvortrag
- Abschlussvortrag + Demo, sofern sinnvoll
- Dokumentation und Archivierung des Projekts
  - Dokumentation des Quelltexts
  - README
  - Je nach Projekt, ggf. schriftlicher Ergebnisbericht (ungefähr 4 Seiten), alternativ zusätzliche Anleitungen/Userguide

# Leistungsnachweis

Abgabe: 4.2.

Voraussetzung für einen Schein:

- Teilnahme an allen gemeinsamen Veranstaltungen (Plenum)
- Spezifikationsvortrag
- Abschlussvortrag + Demo, sofern sinnvoll
- Dokumentation und Archivierung des Projekts
  - Dokumentation des Quelltexts
  - README
  - Je nach Projekt, ggf. schriftlicher Ergebnisbericht (ungefähr 4 Seiten), alternativ zusätzliche Anleitungen/Userguide

# Benotung

## Benotungskriterien:

- ① Sorgfältige Herangehensweise bei Planung, Implementierung, Dokumentation  
→ Evaluation durch Betreuer
- ② Präsentation des fertigen Projekts und Lessons Learned  
→ Evaluation u.a. durch andere Projektgruppen
- ③ Zusammenarbeit im Team  
→ Evaluation u.a. durch Teammitglieder

In Sonderfällen können Einzelnoten vergeben werden!

## Vorstellung der Themen