

---

# Informatik-Praktikum im Grundstudium

## Organisation und Thema

Prof. Dr. W. Hesse / J. Berthold  
Sommersemester 2004  
Philipps-Universität Marburg



---

# VORAUSSETZUNGEN

Vorausgesetzt werden:

- Programmierkenntnisse in Java
- Vorlesungen zur Praktische Informatik 1 und 2  
(einer der beiden Scheine muss **benotet** sein, um die Programmierkenntnisse nachzuweisen)
- Bereitschaft, sich selbstständig in neue Themen einzuarbeiten
- Zeit für das Projekt<sup>a</sup>
- Interesse am Thema

---

<sup>a</sup>Knapper Zeitrahmen; bitte vorher überlegen!

---

# ZIELE DES PRAKTIKUMS

## Team- und Projektarbeit

- selbstständige Organisation
- termingerechte und koordinierte Teamarbeit
- Dokumentation und Präsentation der Ergebnisse
- Einsatz von Werkzeugen für Koordination und Dokumentation

## Vertiefung der Programmierkenntnisse

- Grundlagen der Softwaretechnik
- Entwurf und Programmierung einer größeren Software
- Auswahl und Benutzung verfügbarer Bibliotheken

---

## VORSTELLUNG DES THEMAS

Verwaltungs- und Abspielsoftware für  
Musikstücke in verschiedenen Formaten

---

## VORSTELLUNG DES THEMAS

### Verwaltungs- und Abspielsoftware für Musikstücke in verschiedenen Formaten

- Abspielen von Daten in verschiedenen Audioformaten (mp3, ogg, pcm etc.)
- Verwalten von Informationen zu den Daten (Komponist, Interpret, Titel, Jahr etc.)
- Suche, Sortierung, Speichern der Informationen
- Zusammenstellung und Verwaltung von Abspielisten

---

## VORSTELLUNG DES THEMAS

### Verwaltungs- und Abspielsoftware für Musikstücke in verschiedenen Formaten

- Abspielen von Daten in verschiedenen Audioformaten (mp3, ogg, pcm etc.)
- Verwalten von Informationen zu den Daten (Komponist, Interpret, Titel, Jahr etc.)
- Suche, Sortierung, Speichern der Informationen
- Zusammenstellung und Verwaltung von Abspielisten

Plattformunabhängige Software in Java

Verwendung von vorhandenen Bibliotheken

---

# Zweistufige Entwicklung

## Basisversion:

- Importieren und Abspielen bestimmter Formate
- einfache Suche, Speichern der benötigten Informationen
- Verwaltung von Abspiellisten

---

# Zweistufige Entwicklung

## Basisversion:

- Importieren und Abspielen bestimmter Formate
- einfache Suche, Speichern der benötigten Informationen
- Verwaltung von Abspiellisten

## Erweiterte Version (Increment):

- Unterstützung weiterer Dateiformate
- Verwaltung von Webradios und Audio-CDs  
(erweitertes Datenmodell)
- zusätzliche Informationen, Kategorien und Suchfunktionen, Erweiterung zum *Informationssystem*
- Raum für eigene Ideen

---

# Existierende Produkte

→ WinAmp

⊕ ab Version 5 Verwaltungsfunktionen (Media Library)

⊖ plattformabhängig

---

# Existierende Produkte

## → WinAmp

- ⊕ ab Version 5 Verwaltungsfunktionen (Media Library)

- ⊖ plattformabhängig

## → Windows Media Player

- ⊖ stark plattformabhängig

---

# Existierende Produkte

## → WinAmp

- ⊕ ab Version 5 Verwaltungsfunktionen (Media Library)

- ⊖ plattformabhängig

## → Windows Media Player

- ⊖ stark plattformabhängig

## → Sonique

- ⊕ ansprechendes GUI

- ⊖ keine Verwaltungsfunktionen

## → ...verschiedene andere Programme...

---

## Existierende Produkte

- WinAmp
  - ⊕ ab Version 5 Verwaltungsfunktionen (Media Library)
  - ⊖ plattformabhängig
- Windows Media Player
  - ⊖ stark plattformabhängig
- Sonique
  - ⊕ ansprechendes GUI
  - ⊖ keine Verwaltungsfunktionen
- ...verschiedene andere Programme...

## “Unsere” Software:

- allein in Java implementiert, plattformunabhängig
- mit erweiterten Verwaltungsfunktionen

---

# ORGANISATION

80 Teilnehmer, 3 Monate  $\implies$

Zwei **Projektgruppen** arbeiten konkurrierend

---

# ORGANISATION

80 Teilnehmer, 3 Monate  $\implies$

Zwei **Projektgruppen** arbeiten konkurrierend

→ **Selbstständige Organisation**

Der gesamte Entwicklungszyklus wird von den Teilnehmern organisiert. Bestimmte Gruppenmitglieder übernehmen die Leitung.

---

# ORGANISATION

80 Teilnehmer, 3 Monate  $\implies$

Zwei **Projektgruppen** arbeiten konkurrierend

→ **Selbstständige Organisation**

Der gesamte Entwicklungszyklus wird von den Teilnehmern organisiert. Bestimmte Gruppenmitglieder übernehmen die Leitung.

→ **Tutor als Ansprechpartner**

Der Tutor berät und begleitet die Projektgruppen

---

# ORGANISATION

80 Teilnehmer, 3 Monate  $\implies$

Zwei **Projektgruppen** arbeiten konkurrierend

→ **Selbstständige Organisation**

Der gesamte Entwicklungszyklus wird von den Teilnehmern organisiert. Bestimmte Gruppenmitglieder übernehmen die Leitung.

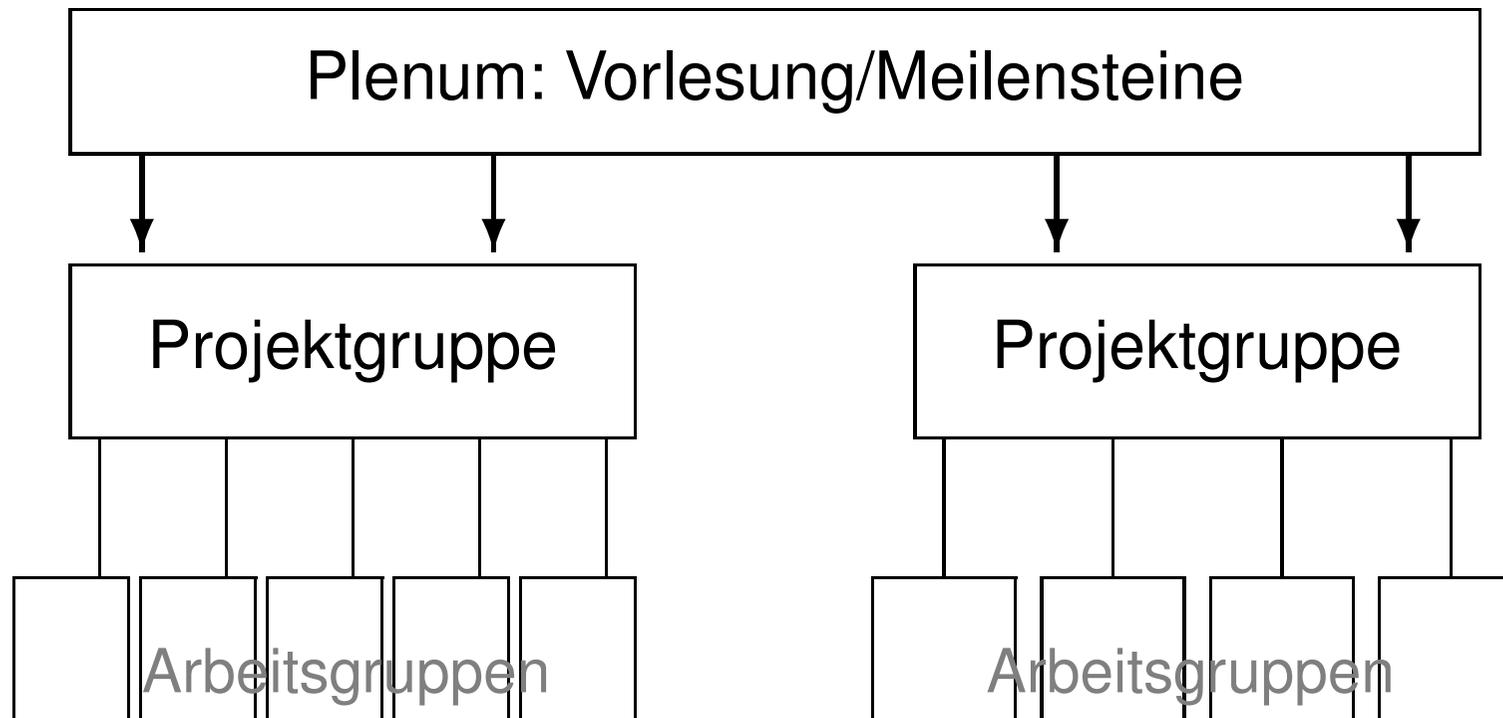
→ **Tutor als Ansprechpartner**

Der Tutor berät und begleitet die Projektgruppen

→ Verteilung von **Teil-Aufgaben** an feste Arbeitsgruppen

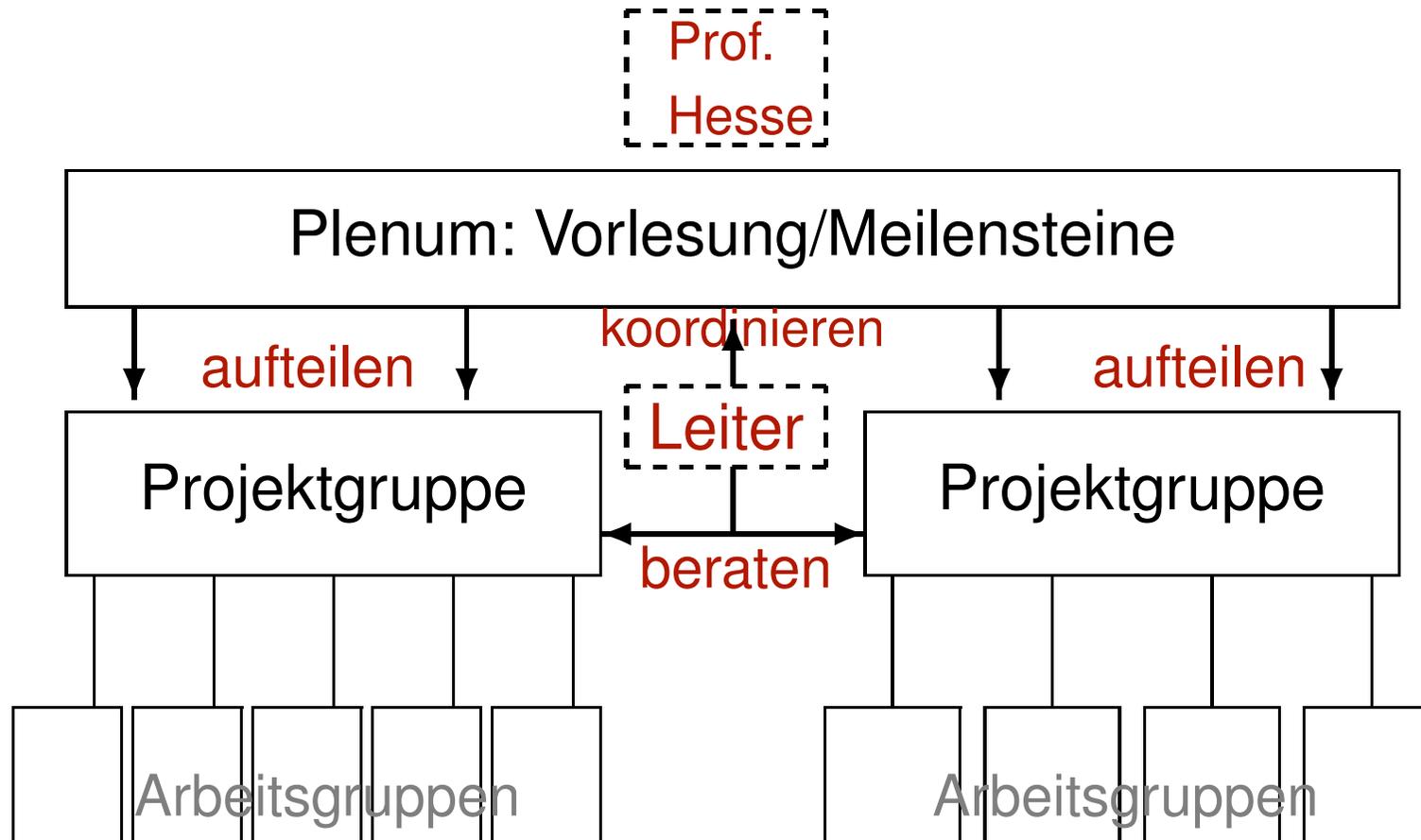
---

# ORGANISATION

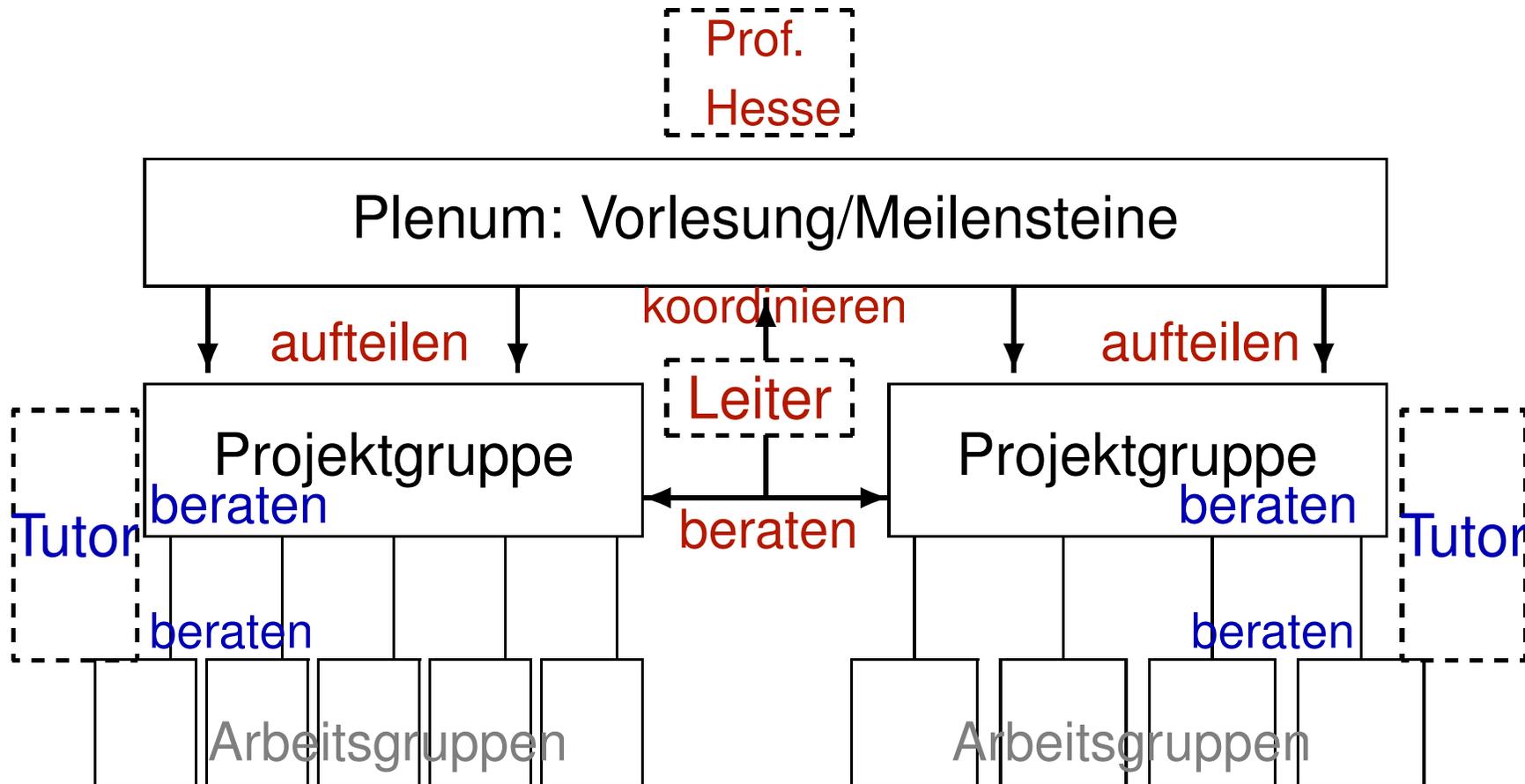


---

# ORGANISATION



# ORGANISATION



---

## Plenum: Chemie HS B

- **Einführungsvorlesung** Softwaretechnik (3 Termine)  
Die Vorlesung vermittelt Kenntnisse über Vorgehensweisen und Techniken für die Softwareentwicklung in größeren Projekten.
- an festgelegten Terminen: **Meilensteine**  
Hier nehmen alle gemeinsam teil, man erfährt den Stand der anderen Projektgruppe. Es werden Entwürfe und Implementierung vorgestellt und diskutiert.
  - Analyse-Review
  - Entwurfs-Review
  - Präsentation der Software (Basis / Erweiterung)
- Abschlusstreffen

---

## Projektgruppe: **Chemie HS B / HS III**

- ab der zweiten Woche: **wöchentliche Gruppentreffen**  
Hier werden Aufgaben koordiniert, Absprachen getroffen und Informationen ausgetauscht. Arbeitsgruppen berichten über ihre Arbeit.
- Außerdem werden **kurze Vorträge** gehalten.  
Vorträge regeln die Zusammenarbeit der Gruppe oder vermitteln Wissen über die eingesetzten Werkzeuge.
- Alle Treffen werden **protokolliert**.

---

## Projektgruppe: **Chemie HS B / HS III**

- ab der zweiten Woche: **wöchentliche Gruppentreffen**  
Hier werden Aufgaben koordiniert, Absprachen getroffen und Informationen ausgetauscht. Arbeitsgruppen berichten über ihre Arbeit.
- Außerdem werden **kurze Vorträge** gehalten.  
Vorträge regeln die Zusammenarbeit der Gruppe oder vermitteln Wissen über die eingesetzten Werkzeuge.
- Alle Treffen werden **protokolliert**.

### **Treffen der Arbeitsgruppen** (ohne festen Termin)

- Hier werden die Teilaufgaben bearbeitet.
- Die Gruppen teilen sich nach Bedarf weiter auf.

---

# Tutoren

## Aufgaben des Tutors:

- an Treffen teilnehmen
- Gruppenarbeit beurteilen
- Ansprechpartner für Kleingruppen sein
- Vorschläge machen (organisatorisch und technisch)

Tutoren: Ivo Pacák, Aaron Ruß

---

# Tutoren

## Aufgaben des Tutors:

- an Treffen teilnehmen
- Gruppenarbeit beurteilen
- Ansprechpartner für Kleingruppen sein
- Vorschläge machen (organisatorisch und technisch)

## Nicht Aufgabe des Tutors:

- Gruppe leiten
- Software testen
- Vorschläge umsetzen

Tutoren: Ivo Pacák, Aaron Ruß

---

# SCHEINKRITERIEN

## Projektgruppe:

- Dokumente zu Analyse, Entwurf und Test
- lauffähige Softwaremodule, Quellcode, Dokumentation

---

# SCHEINKRITERIEN

## Projektgruppe:

- Dokumente zu Analyse, Entwurf und Test
- lauffähige Softwaremodule, Quellcode, Dokumentation

## Alle Teilnehmer:

- aktive Teilnahme an Gruppensitzungen
- kontinuierliche Mitarbeit in einer Arbeitsgruppe
- individuelle Leistung: wahlweise
  - Vortrag in der Projektgruppe
  - GUI-Prototyp
  - Protokoll
  - Vortrag im Plenum

---

# VORTRAGSTHEMEN FÜR PROJEKTGRUPPEN

(vorläufige Zusammenstellung)

→ **Infrastruktur und Werkzeuge**

Kommunikationsstrukturen, Versionsverwaltung (CVS),  
IDE und Konfiguration, Modultest mit JUnit

→ **Projektrichtlinien**

Coderichtlinien, Dokumentation (Javadoc), Testplan

→ **Expertise zum Thema**

Java-Bibliotheken für Sound (2 Vorträge)

---

# VORTRAGSTHEMEN FÜR PROJEKTGRUPPEN

(vorläufige Zusammenstellung)

→ **Infrastruktur und Werkzeuge**

Kommunikationsstrukturen, Versionsverwaltung (CVS),  
IDE und Konfiguration, Modultest mit JUnit

→ **Projektrichtlinien**

Coderichtlinien, Dokumentation (Javadoc), Testplan

→ **Expertise zum Thema**

Java-Bibliotheken für Sound (2 Vorträge)

Je Vortrag **zwei Personen** , Dauer **ca. 15 Minuten**.

Zum Vortrag: kurze **schriftliche Zusammenfassung**.

---

# PROTOTYPEN

In jeder Gruppe können **zwei Prototypen** erstellt werden. Es handelt sich um **experimentelle Prototypen**:

- Aussehen der Benutzeroberfläche
- Bedienungsmöglichkeiten
- Konkretisierung der geforderten Funktionen

Evtl. kann man einzelne Teile wiederverwenden, sie müssen aber auf jeden Fall noch angepasst werden.

---

# PROTOTYPEN

In jeder Gruppe können **zwei Prototypen** erstellt werden. Es handelt sich um **experimentelle Prototypen**:

- Aussehen der Benutzeroberfläche
- Bedienungsmöglichkeiten
- Konkretisierung der geforderten Funktionen

Evtl. kann man einzelne Teile wiederverwenden, sie müssen aber auf jeden Fall noch angepasst werden.

- An jedem Prototyp können **zwei Personen** arbeiten.
- Die Programmierung sollte **möglichst früh beginnen**.

---

## VORTRAG IM PLENUM / PROTOKOLL

An Terminen im Plenum: Präsentation von Ergebnissen

- je Projektgruppe **drei Sprecher...**
- ... stellen die Arbeitsergebnisse der Gruppe dar.
- Sie fügen **von der Gruppe gelieferte Dokumente** zu einem Vortrag zusammen

---

# VORTRAG IM PLENUM / PROTOKOLL

An Terminen im Plenum: Präsentation von Ergebnissen

- je Projektgruppe **drei Sprecher...**
- ... stellen die Arbeitsergebnisse der Gruppe dar.
- Sie fügen **von der Gruppe gelieferte Dokumente** zu einem Vortrag zusammen

Protokolle der Gruppentreffen

- enthalten **alle Ergebnisse** und den **Diskussionsverlauf**
- dokumentieren den Fortschritt der Gruppe
- werden der Gruppe zugänglich gemacht
- **alle Treffen sind zu protokollieren**

---

## ABLAUF UND ZEITPLAN

21.4.	Projekt-Initialisierung – Vorlesung
28.4.	Gruppeneinteilung – Vorlesung <b>erstes Gruppentreffen</b> (Analyse)
5.5.	Vorlesung Gruppentreffen (Analyse), erster Vortrag
12.5.	<b>Analyse-Review</b> Gruppentreffen (Arbeitsgruppen), Vorträge
19.5.	Gruppentreffen (Entwurf), Vorträge
26.5.	<b>Entwurfs-Review</b> , Prototypen

⋮

---

⋮

26.5.	<b>Entwurfs-Review</b> , Prototypen
2.6.	Gruppentreffen (Implementierung), Vorträge
9.6.	Gruppentreffen (Implementierung), Vorträge
16.6.	Gruppentreffen (Implementierung)
23.6.	<b>Milestone: Basisversion</b>
30.6.	<i>Sport Dies</i>
7.7.	Gruppentreffen (Implementierung)
14.7.	<b>Milestone: Erweiterte Version</b>
21.7.	Testbericht und Abschlusstreffen <b>strikte Deadline für Nachbesserung</b>

---

# ANMELDUNG / SCRIPT / AUFGABEN

Verbindliche Anmeldung zum Praktikum:

- Bitte Anmeldebögen ausfüllen
- Warum Einschätzung und Interessen ?
  - ⇒ gleichmäßige Einteilung der Projektgruppen
- **Gruppentreffen** nächste Woche nach der Vorlesung

---

# ANMELDUNG / SCRIPT / AUFGABEN

Verbindliche Anmeldung zum Praktikum:

- Bitte Anmeldebögen ausfüllen
- Warum Einschätzung und Interessen ?
  - ⇒ gleichmäßige Einteilung der Projektgruppen
- **Gruppentreffen** nächste Woche nach der Vorlesung

**Script:** Ausgabe vor der jeweiligen Vorlesung

Unkostenbeitrag für Druck: 2 Euro

---

# ANMELDUNG / SCRIPT / AUFGABEN

Verbindliche Anmeldung zum Praktikum:

- Bitte Anmeldebögen ausfüllen
- Warum Einschätzung und Interessen ?
  - ⇒ gleichmäßige Einteilung der Projektgruppen
- **Gruppentreffen** nächste Woche nach der Vorlesung

**Script:** Ausgabe vor der jeweiligen Vorlesung

Unkostenbeitrag für Druck: 2 Euro

**Bis nächste Woche:**

- Information über Audioformate und Vorbilder
- Diskussion der Aufgabenstellung vorbereiten

---

# KOMMUNIKATION

- WWW-Seite des Praktikums:

*<http://www.mathematik.uni-marburg.de/~berthold/APraSS04/>*

- **Newsgroups** und **CVS** der Projektgruppen:

*[pc12392.mathematik.uni-marburg.de](mailto:pc12392.mathematik.uni-marburg.de)*

(wird noch eingerichtet und genauer erklärt)

- Für Fragen und Kommentare:

*[berthold@informatik.uni-marburg.de](mailto:berthold@informatik.uni-marburg.de)*