

# Imperative Programmierung für Informatik

## Übung 1: Die Arbeitsumgebung

**Andreas Ruscheinski**

Bereich Informatik, Universität Rostock

## Plan für heute

1. Vorstellung
2. Hinweise ala „Wir haben es euch doch gesagt“
3. Vorstellung der Entwicklungsumgebung
4. Ausblick

## Wer ich bin

### M.Sc. Andreas Ruscheinski (alias „Du“)

2017 - 2023: Mitarbeiter am Lehrstuhl für Modellierung und Simulation

2023 - heute: Lehrkraft für besondere Aufgaben

@ [andreas.ruscheinski@uni-rostock.de](mailto:andreas.ruscheinski@uni-rostock.de)

✉ [Konrad-Zuse-Haus](#), Raum 241

☎ 0381/498-7455

## Strukturierung - Vorlesung/Übung/Praktikum

- **Vorlesung:** Vermittlung von neuen Lernstoff auf theoretischer Ebene
  - Inhalt: Theorien, Ideen, Probleme, Lösungen
  - Ziel: Bekanntmachung von Lernstoff
- **Übung:** Wiederholung und praktische Anwendung des Lernstoffs
- **Praktikum:** Bearbeitung des Lernstoffs rein aus programmieretechnischer Sicht

## Strukturierung - Vorlesung/Übung/Praktikum

- **Vorlesung:** Vermittlung von neuen Lernstoff auf theoretischer Ebene
- **Übung:** Wiederholung und praktische Anwendung des Lernstoffs
  - Inhalt: Lösung von konkreten Problemen
  - Ziel: Erste Festigung des Lernstoffs
- **Praktikum:** Bearbeitung des Lernstoffs rein aus programmiertechnischer Sicht

## Strukturierung - Vorlesung/Übung/Praktikum

- **Vorlesung:** Vermittlung von neuen Lernstoff auf theoretischer Ebene
- **Übung:** Wiederholung und praktische Anwendung des Lernstoffs
- **Praktikum:** Bearbeitung des Lernstoffs rein aus programmiertechnischer Sicht
  - Inhalt: Implementierung von komplexeren Algorithmen + Weitergehende Anwendung und Verknüpfung des Lernstoffs
  - Ziel: Vertiefende Festigung des Lernstoffs

## Strukturierung - Vorlesung/Übung/Praktikum

- **Vorlesung:** Vermittlung von neuen Lernstoff auf theoretischer Ebene
- **Übung:** Wiederholung und praktische Anwendung des Lernstoffs
- **Praktikum:** Bearbeitung des Lernstoffs rein aus programmiertechnischer Sicht

→ Nicht ausreichend um zu lernen!

## Wie „lerne“ ich jetzt was?

### **Durch selber machen und ausprobieren!**

- Ziel: Verknüpfung von Wissen!
- Ansätze:
  - Selbständigen lösen von Aufgaben
  - Diskussion von Lösungen
  - Ausprobieren von unterschiedlichen Ansätzen
  - Beheben von Fehlern in eigener Lösung
  - Alles löschen und neu anfangen



## Wie „lerne“ ich jetzt was?

### Durch selber machen und ausprobieren!

- Ziel: Verknüpfung von Wissen!
- Ansätze:
  - Selbständigen lösen von Aufgaben
  - Diskussion von Lösungen
  - Ausprobieren von unterschiedlichen Ansätzen
  - Beheben von Fehlern in eigener Lösung
  - Alles löschen und neu anfangen

→ Übung und Praktikum **hilft** dabei

## Hinweise - Allgemein

- Veranstaltungen haben idR. eine steile Lernkurve
- Besucht ALLE Veranstaltungen
- Bei Fragen: Immer Mail (oder Stud.IP)
  - ... zur Vorlesung: Prof. Mühl
  - ... zur Übung: Übungsleiter = mich
  - ... zum Praktikum: Tutor oder mich
- Bei Problemen: Persönlichen Gespräch oder Mail → Rechtzeitig! Jedes Problem ist lösbar!
- Schaut regelmäßig in die E-Mails (mind. 1-2 täglich)
- Arbeitet in Gruppen... findet „Leidensgefährten“

## Organisatorische Hinweise - Übungsablauf

- Generelle Aufteilung:
  - Kurze Wiederholung der Vorlesung ( $\approx 20\text{min}$   $\rightarrow$  Nach Bedarf)
  - Bearbeitung der Übungsaufgaben ( $\approx 50\text{min}$   $\rightarrow 70\text{min}$ )
  - Gemeinsame Erarbeitung von einiger Lösungen ( $\approx 20\text{min}$ )
- Übungsaufgaben werden vor der Übung (auf Stud.IP und) im SVN hochgeladen
- Nicht alle Aufgaben können/müssen zwangsläufig in der Übung/Praktikum bearbeitet werden  $\rightarrow$  Selbststudium

## Organisatorische Hinweise - Hausaufgaben

- Prüfungszulassung: 50% der möglichen Punkte
- Anfangs: wöchentlich, Später:  $\approx$  14-tägig
- Erste Hausaufgabe: 20.10.2024 – 27.10.2024
- Abgabe via SVN
- Bearbeitung in Gruppen von 3-4 Studierenden; Kleinere Gruppen werden zusammengelegt
- Eintragung in Gruppen via **Stud.IP** in der Vorlesung
- Eintragung nur bis 23.10 23:59 Uhr möglich → danach Verzögerung + Abgabe der HA 1 nicht möglich

## Die Entwicklungsumgebung

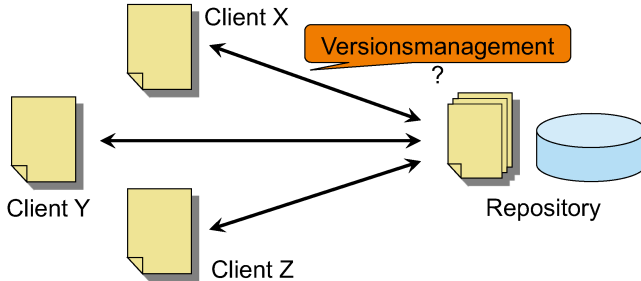
- Programmiersprache: C
- Editor: Visual Studio Code + Plugins
- Versionskontrolle: SVN
- Für Zuhause: Eigene Umgebung einrichten ← „Hausaufgabe 00“
  - Diverse Anleitung zur Einrichtung im SVN und auf Stud.IP
  - Fragen/Probleme per Mail an mich
- In der Uni: Rechnerpool (Unicomp)
  - Laufwerk R: = DEIN ORDNER!
  - Login mit dem Nutzerkürzel!

**Jetzt:** Bisschen Umgebung kennenlernen

## Versionskontrolle mit SVN

Folien von Dr. Helge Parzyjega

# Verteilte Entwicklung



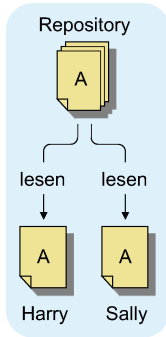
## Arbeitskopie

- > **Lokale** Kopie zur Bearbeitung beim Client
- > Kann Teile oder gesamtes Repository umfassen

## Repository

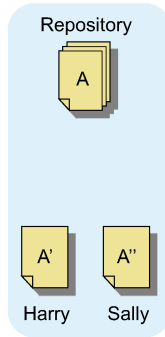
- > Lager für alle Ressourcen eines Projektes (inklusive Historie)
- > Oft an **zentralem** Ort realisiert → Repository Server

# Wozu eine Versionskontrolle?



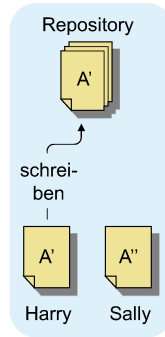
## Lesen

- > Harry und Sally erzeugen eigene Kopien



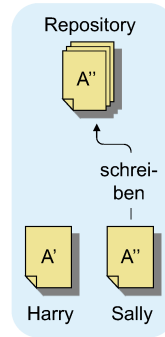
## Editieren

- > Beide arbeiten gleichzeitig auf ihren Kopien



## Schreiben

- > Zurückschreiben von Harrys Änderungen

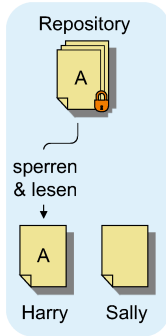


## Überschreiben

- n
- > Verlust von Harrys Änderungen

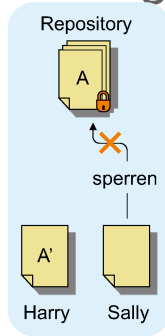


# Lösung: Sperren → Ändern → Freigeben



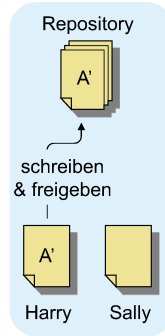
## Sperren

- > Harry sperrt Dokument zur Bearbeitung



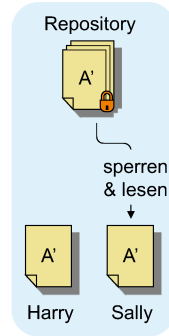
## Ändern/ Warten

- > Sallys Sperrversuch scheitert
- > Warten auf Freigabe



## Freigeben

- > Harrys Freigabe erfolgt nach Änderung

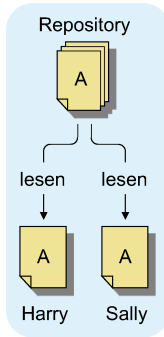


## Sperren

- > Sally sperrt Dokument zur Bearbeitung

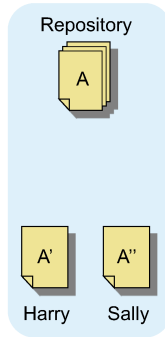
# Lösung: Kopieren → Ändern →

...



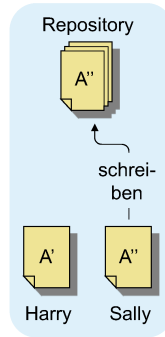
## Kopieren

> Harry und Sally erzeugen eigene Kopien



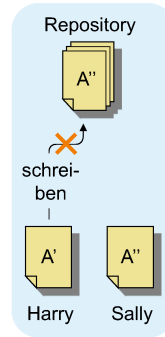
## Editieren

> Beide arbeiten gleichzeitig auf ihren Kopien



## Schreiben

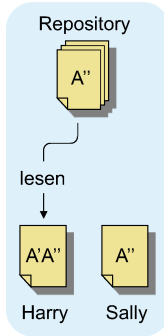
> Sally schreibt Änderungen zuerst zurück



## Schreiben

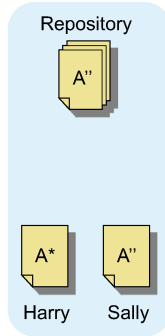
> Harrys Schreibversuch schlägt fehl (da nicht mehr aktuell)

# ... → Zusammenführen



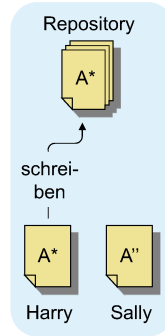
## Update

- > Harry vergleicht seine Version mit aktueller



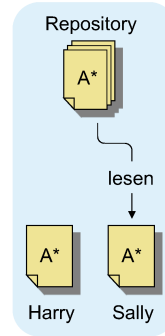
## Zusammenführe

- n
- > Erstellung einer gemeinsamen Version
- > Konfliktpotential



## Schreiben

- > Harry schreibt zusammengeführte Version zurück



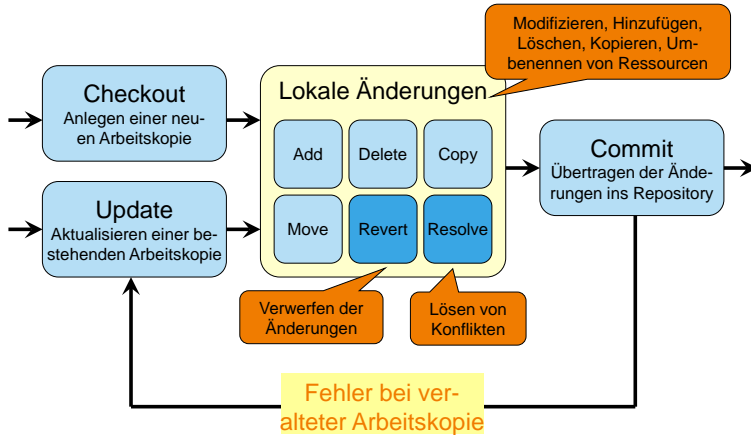
## Lesen

- > Sally erhält neue Version vom Repository

# Subversion

- Versionsmanagementsystem
  - Zentrales Repository für Ordner und Dateien
  - Versionsmanagement → Revisionen des gesamten Repositorys
  - Historie (aller Revisionen) einer Ressource verfügbar  
→ Speicherung des Originals und sämtlicher Änderungen
  - Unterstützung von Kopieren/Ändern/Zusammenführen
  - Explizites Sperren von Ressourcen ebenfalls möglich
- Subversion im Web
  - Webseite → <http://subversion.tigris.org/>
  - Download des Subversion Servers und Kommandozeilen-Clients
  - Dokumentation → <http://svnbook.red-bean.com/>

# Subversion Workflow



# Konflikte

- > Scheitern der automatischen Zusammenführung von Revisionen beim Aktualisieren → **Konflikt**
- > Zusätzliche Kopien bei Dateien mit Konfliktstatus
  - > DATEI : Datei mit enthaltenen **Konfliktmarkierungen**
  - > DATEI .mine: Eigene Arbeitskopie der Datei vor Aktualisierung
  - > DATEI .r XX: aktuelle Revision XX aus dem Repository
  - > DATEI .r YY: Ausgangsrevision YY (<XX) der Arbeitskopie
- > Konflikte werden abschnittsweise innerhalb der Datei markiert

```
... Textzeilen der Datei ohne Konflikt
<<<<<<< .mine
Zeilen der Arbeitskopie
=====
Zeilen der aktuellen Repository-Revision
>>>>>>> .r42
Textzeilen der Datei ohne Konflikt ...
```

# Auflösen von Konflikten

- **Manuelles Lösen** des Konfliktes durch Benutzer
  - Editieren der entsprechenden Datei
  - Verwerfen seiner Änderungen → Revert
  - Ersetzen der Datei durch gewünschte Revision
- Revert
  - **Verwirft lokale Änderungen** der Ressource
  - Kann keine gelöschten Verzeichnisse wiederherstellen
- Resolve
  - Mitteilung über erfolgte Lösung des Konfliktes  
→ Bietet **keinerlei semantische Konfliktbehandlung**
  - Löscht nur Konfliktstatus der Ressource
  - Löscht zusätzlich angelegte Hilfskopien

## Unser SVN-Repository

- Repository-URL: <https://svn.informatik.uni-rostock.de/lehre/ip2024/...>
  - **public**: Öffentliche Ressourcen (z.B. Folien), Lesezugriff für Studierende
  - **playground**: Spielplatz mit Schreibzugriff für alle
  - **groups/xy**: Privater Ordner der Gruppe xy, u.a. für Abgabe der Hausaufgaben



## Regeln im SVN

- Nur Quellcode-Dateien einchecken (Keine .exe)
- Oft commiten
- Auf Dateipfade achten
  - Die Hausaufgabenblätter sagen genau in welchem Pfad die abgegebenen Dateien liegen sollen
  - Mit dem Browser kontrollieren
  - Keine Leerzeichen oder Sonderzeichen in Pfaden oder Dateinamen verwenden

**Jetzt:** Bisschen mit SVN spielen.