

# Imperative Programmierung (IPR)

## Kapitel 0: Motivation und Organisatorisches

**Univ.-Prof. Dr.-Ing. habil. Gero Mühl**

Lehrstuhl für Architektur von Anwendungssystemen (AVA)  
Fakultät für Informatik und Elektrotechnik (IEF)  
Universität Rostock

Universität  
Rostock



Traditio et Innovatio



# Inhalte

1. Motivation

2. Organisatorisches

# Kapitel 0.1

# Motivation

# Motivation

- **Programmierung** ist das Erstellen eines Programms, mit dem Ziel dieses auf einem Computer auszuführen.
- Ein **imperatives Programm** ist eine Folge elementarer Befehle, die der Computer sequentiell abarbeitet.
- Alle Computer werden letztendlich in einer sehr einfachen imperativen Sprache – der jeweiligen **Maschinensprache** – programmiert.
- Die imperative Programmierung bildet daher die Basis für alle anderen Programmierparadigma, wie z. B. der funktionalen Programmierung.
- Programmieren ist wesentlich für den Berufsalltag eines Informatikers.
- „Informatiker“, die nicht gut programmieren können, haben nicht verstanden, was den Beruf eines Informatikers ausmacht.

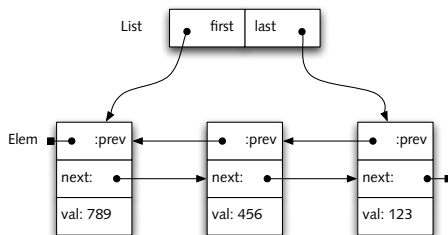
# Was heißt „Programmieren lernen“?

## Versuch einer Analogie

- Programmiersprache = Auto
- Programmieren = Auto fahren
- Programmieren lernen = Auto fahren lernen
- Ihre erste Programmiersprache = Fahrschulauto
- Fahrschule
  - Der theoretische Unterricht erklärt die Zusammenhänge.
  - Die Fahrstunden trainieren die Fähigkeit, tatsächlich Auto zu fahren.
- So wenig, wie man Autofahren aus einem Buch lernen kann, so wenig geht das beim Programmieren.
- Ziel der Fahrschule ist es nicht, Ihnen das Fahren eines bestimmten Auto beizubringen, sondern das Autofahren – mit dem Ziel, nachher verschiedene Autos fahren zu können.

# Motivation

- Für die imperative Programmierung haben sich über die Jahrzehnte eine Reihe grundlegender **Datentypen** etabliert.
  - Listen, Keller, Warteschlangen, Tabellen, Bäume, Graphen etc.
- Für jeden Datentyp gibt es diverse Realisierungsmöglichkeiten in Form konkreter **Datenstrukturen**.
  - z. B. die Realisierung einer Liste als Array oder als verkettete Liste
- Die Realisierungen unterscheiden sich teils wesentlich in ihren Eigenschaften (z. B. Laufzeit der Operationen, Speicherverbrauch).
- Auch aufgrund der vielfachen Verwendung in der Praxis sollte jeder Informatiker diese Datentypen und ihre Realisierungen kennen.



# Inhalt der Veranstaltung

- Diese Vorlesung
  - führt in die imperative Programmierung am Beispiel der Programmiersprache C ein und
  - gibt einen Überblick über die wichtigsten Datentypen der Informatik.
- Hierbei wird
  - die **Spezifikation** der Datentypen als abstrakter Datentyp und
  - die **Implementierung** ihrer Realisierungsvarianten in Cbesprochen.
- Die Vorlesung *Algorithmen und Datenstrukturen* behandelt darauf aufbauend nächstes Semester die wichtigsten Algorithmen der Informatik und ihre Eigenschaften
  - Sortieren, Suchen, ...
  - Asymptotische Komplexitätsanalyse

# Lernziele der Veranstaltung

- Prinzipien der imperativen Programmierung kennenlernen.
  - Anweisungen, Ausdrücke, Kontrollstrukturen, Prozeduren etc.
- Grundlegende Begriffe der Programmierungstechnik verstehen.
  - Algorithmus, Sprache, Semantik, Spezifikation, Verifikation etc.
- Spezifikationsverfahren erlernen und diese auf einfache Problemstellungen anwenden können.
- Die grundlegenden Datentypen der Informatik und ihre Realisierungsmöglichkeiten kennenlernen und implementieren können.



# Inhalt der Vorlesung

## ① Einführung

- Was ist Informatik?
- Was ist ein Algorithmus?
- Was ist eine Programmiersprache?
- Was ist ein Programm?
- ...

## ② Einführung in C

## ③ Abstrakte Datentypen

## ④ Fallbeispiel: Listen

## ⑤ Fallbeispiel: Keller

## ⑥ Fallbeispiel: Warteschlangen

## ⑦ Fallbeispiel: Tabellen

## ⑧ Fallbeispiel: Bäume

## Kapitel 0.2

# Organisatorisches

# Verantwortlicher Dozent

Name:

Univ.-Prof. Dr.-Ing. habil. Gero Mühl

Email:

[gero.muehl@uni-rostock.de](mailto:gero.muehl@uni-rostock.de)

Telefon:

0381 / 498-7621 (Sekretariat)

Sprechstunde:

nach Vereinbarung

Lehrstuhl:

Architektur von Anwendungssystemen (AVA)

URL:

<https://www.ava.uni-rostock.de>

Adresse:

Albert-Einstein-Str. 22, Raum 267 (Sekretariat)

# Das IPR-Team

## ■ Beteiligte Übungsleiter

Name	E-Mail
Willi Brekenfelder	<code>willi.brekenfelder@uni-rostock.de</code>
Andreas Ruscheinski	<code>andreas.ruscheinski@uni-rostock.de</code>
Justin Kreikemeyer	<code>justin.kreikemeyer@uni-rostock.de</code>

- Bitte erfragen Sie die Sprechzeiten Ihres Übungsleiters in der ersten Übung oder schlagen Sie sie auf seiner persönlichen Webseite nach.

# Kommunikationswege

- Vorlesung im Stud.IP
  - Folien
  - Übungsblätter
  - Ankündigungen
  - Diskussionsforum
- E-Mail
  - Terminabsprachen für Sprechzeiten
- Persönliche Kommunikation in der Übung
  - Verständnisprobleme und Fragen von allgemeinem Interesse
- Persönliche Kommunikation in der Sprechstunde
  - Individuelle Fragen und Probleme, die nicht für eine Erörterung in der Übung geeignet sind

# Veranstaltungen

## ■ Vorlesung (3 SWS)

- Mittwochs, 13-15 Uhr
- Donnerstags, 15-17 Uhr (in ungeraden Wochen)
- Führt in das Gebiet ein und ermöglicht so **weiteres Selbststudium** und **selbstständige Nutzung der Literatur**.
- Gibt Hinweise auf Gesamtzusammenhänge und etwaige Probleme.
- Bereitstellung als Folienskriptum im Stud.IP.
- Kleinere Abweichungen nachträglich möglich (z. B. Ergänzungen und Korrekturen).

## ■ Übung (2 SWS) und Praktikum (1 SWS)

- Ergänzung der Vorlesung durch umfangreichere Beispiele.
- Dialog von Übungsleiter und Studierenden, z. B. zur Beseitigung von Unklarheiten.
- Gemeinsames Programmieren im Praktikum (nicht für alle Studiengänge vorgesehen).
- Eine aktive Mitarbeit ist zur erfolgreichen Auseinandersetzung mit dem Stoff unabdingbar.

# Zu den Übungen

- Erfolg in der Übung korreliert mit Erfolg in der Klausur.
- Ohne aktive Teilnahme an den Übungen werden Sie die Prüfung höchstwahrscheinlich nicht bestehen.
- Bilden Sie Arbeitsgemeinschaften und bearbeiten Sie die Aufgaben *nicht* alleine, sondern zusammen im Team!
- Denken Sie aber auch daran: in der Klausur sind Sie allein!
- Plagiiere führt zum Nichtbestehen des Übungsblattes.
- Bei Mangel an Interesse/Vorbereitung spielt der Übungsleiter nicht den Alleinunterhalter, sondern beendet die Übung.
- Weitere Details zur Durchführung erfahren Sie von Ihrem Übungsleiter in der ersten Übung.

# Prüfungsmodalitäten

- Zulassungsvoraussetzung für die Modulprüfung
  - Erreichen von 50% der Hausaufgabenpunkte
- Prüfung „Imperative Programmierung für Informatik“, 9 CP, 120-minütige Klausur.
- Prüfung „Imperative Programmierung“, 6 CP, 120-minütige Klausur.
- Prüfung „Imperative und Funktionale Programmierung“, 9 CP
  - Zwei getrennte Prüfungsleistungen
  - 80-minütige Klausur „Imperative Programmierung“
  - Prüfung „Funktionale Programmierung“
- **Melden Sie sich rechtzeitig für die richtige Prüfung an!**
- **Eine verpasste Anmeldung kann nicht nachgeholt werden!**



# Vielen Dank für Ihre Aufmerksamkeit!

Univ.-Prof. Dr.-Ing. Gero Mühl

`gero.muehl@uni-rostock.de`  
`https://www.ava.uni-rostock.de`