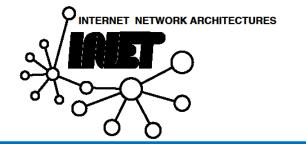


## Einführung in die Programmierung

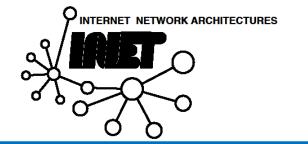
**Organisation und Inhalt** 



#### Kontakt

#### ■Anja Feldmann

- ➤ Professorin für Internet Network Architectures an der TU (ab 1.1.2018 am Max Planck Institut für Informatik in SB, hat keinen Einfluss auf den Lehrbetrieb im WS17/18.)
- Siehe auch http://www.inet.tu-berlin.de
- > Sprechstunde: Di. 12.00 12.30
- Bitte mit Voranmeldung per E-mail oder über das Sekretariat
- Sekretariat MAR 4-4
  - secretary@inet.tu-berlin.de

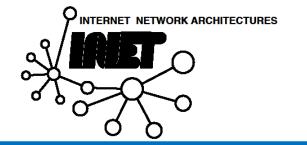


#### Wichtige Kontakte

- Damien Foucard (Zuständig für die Organisation)
- ☐ Florian Streibelt (Zuständig für die Osiris/SVN Infrastruktur)
- Franziska Lichtblau
- Theresa Enghardt
- Thorben Krüger
- Mirko Palmer
- Infos: via ISIS

#### Einführung in die Programmierung

- Kontakt: introprog-team@inet.tu-berlin.de
- □ Kontakt nur über obige E-mail-Adresse, <u>nicht</u> individuell!

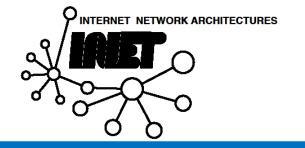


# Kontakt für ausländische Studierende

□ Sahar Ben Hassine Maria Kornienko Nesrine Ouanes May Said Ernest Valverde Laks

http://www.betreuung-int-stud.tu-berlin.de/menue/home/

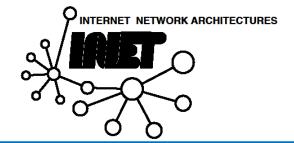
☐ Hilfe bei sprachlichen und kulturellen Problemen



## Zielgruppe

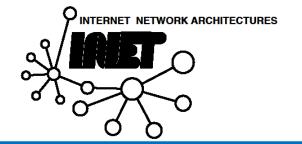
- Studierende der Studiengänge
  - > Informatik
  - > Technische Informatik
  - Medieninformatik
  - Lehramt

Anja Feldmann, TU Berlin, 2017



## Lernziele

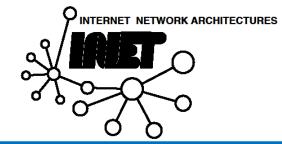
Anja Feldmann, TU Berlin, 2017



### Grundlegende Datenstrukturen

- Kenntnis
  - > Elementarer Datenstrukturen
  - Such- und Sortierverfahren

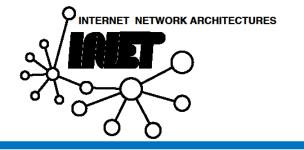
- □ Fähigkeit
  - > Probleme und Strukturen (wieder)zuerkennen
  - Für ein gegebenes Anwendungsproblem die geeignete Datenstruktur zu wählen



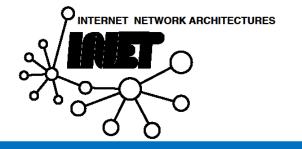
### Einführung in die Programmierung

- ☐ Beherrschung der Sprachelemente von C
- Verständnis des Paradigmas: Iterativen Programmierung
- □Fähigkeit
  - > Einfache Programme in C zu schreiben
  - Lesbare und verständliche Programme zu schreiben
  - ➤ Den Aufwand eines Algorithmus bzw. eines C-Programms abzuschätzen





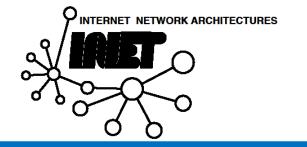
- ☐ Einführung in eine Programmiersprache
  - > Elementare Datentypen und Operatoren
  - > Kontrollstrukturen: Verzweigungen, Schleifen
  - > Funktionen
  - Dynamische Datenstrukturen
- Datenstrukturen
  - Listen
  - Queue, Stack und Heap
  - Bäume



#### Inhalte

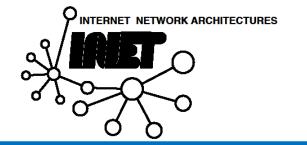
- ☐ Elementare Algorithmen
  - > Suchen
  - Sortieren

- Algorithmen
  - Aufwandsabschätzung
  - > Korrektheit



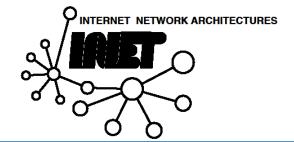
## Schwerpunkte

- Zwei Schwerpunkte entsprechend der Werkzeugklassen
  - > Erlernen der Sprache C
  - > Umgang mit Datenstrukturen und algorithmischen Aspekten
- ☐ Entsprechend zwei Vorlesungsteile
  - C-Kurs (erste zwei Vorlesungswochen)
  - > Einführung in die imperative Programmierung (IntroProg)



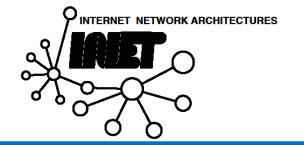
#### Vorbemerkungen - C

- Beispielsprache C
  - > Weit verbreitet, etabliert
  - Auf allen Plattformen verfügbar
  - Grundlage für viele weitere Vorlesungen, u.a. Rechnerorganisation
- Hier: Programmierung "im Kleinen", algorithmisches Handwerkszeug
- Programmbeispiele auf Deutsch und/oder Englisch



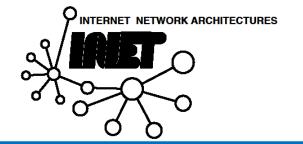
## Ablauf

Anja Feldmann, TU Berlin, 2017



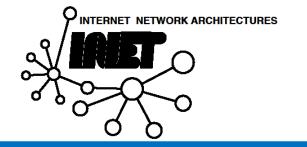
#### Veranstaltungskomponenten

- Vorlesung:
  - Vorstellung der Konzepte
  - Beispiel Programme
- Tutorien:
  - Q&A (Fragen und Antworten)
  - Besprechung der Hausaufgaben
  - Codebeispiele inklusive Fehlersuche!
- Betreute Rechnerzeiten:
  - ➤ Hilfestellung beim Programmieren vor Ort
- Hausaufgaben:
  - Eigenständiges Auseinandersetzen mit den Konzepten
  - Sowohl als Einzel- als auch als Gruppenabgaben (wie auf Blatt angegeben)



### Veranstaltungsgliederung

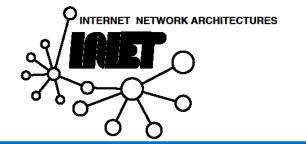
- C-Kurs:
  - Vorstellung der Konzepte
  - Beispiel Programme
  - Dauer: Zwei Wochen
  - > Folgende Vorlesungen finden in diesen zwei Wochen nicht statt:
    - Rechnerorganisation
    - Formale Sprachen und Algorithmen
    - Informatik Propädeutikum
- ☐ Einführung in die imperative Programmierung (IntroProg):
  - Grundlegende Datenstrukturen
  - Algorithmen Am Beispiel von Listen, Bäumen, und Sortieren
  - Dauer: Rest des Semesters



## **C-Kurs Vorlesungen**

- Dienstag
- Mittwoch
- Donnerstag
- □ Freitag
- Montag
- Dienstag
- Mittwoch
- Donnerstag

- 17.10.2017 12 Uhr ct in HE 101
- 18.10.2017 10 Uhr ct in HE 101
- 19.10.2017 12 Uhr ct in H 0105
- 20.10.2017 10 Uhr ct in HE 101
- 23.10.2017 08 Uhr ct in H 0105
- 24.10.2017 12 Uhr ct in HE 101
- 25.10.2017 10 Uhr ct in HE 101
- 26.10.2017 12 Uhr ct in H 0105

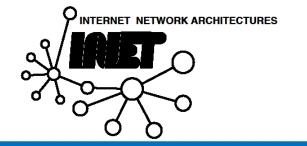


## IntroProg Vorlesung

Ab dem 3.11.2017

☐ Freitag: 10.15 – 11.45 Uhr in HE 101

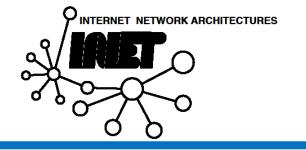
- Diese Vorlesungen starten in der Woche vom 30.10!
  - > Rechnerorganisation
  - ➤ Formale Sprachen und Algorithmen
  - ➤ Informatik Propädeutikum



#### Prüfungsmodalitäten

- ☐ Portfolio Prüfung Komponenten
  - ➤ Einzelaufgaben: 15 Portfoliopunkte
    - C-Kurs Aufgaben
    - Größere Programmieraufgabe über Weihnachten (Wiederholungsmöglichkeit)
  - ➤ Gruppenhausaufgaben: IntroPROG: 35 Portfoliopunkte
  - ➤ Schriftlicher Test: **50 Portfoliopunkte** 
    - Am Ende des Semesters: Mittwoch, der 21.2.2018, von 9 bis 11 Uhr
    - Am Anfang des Sommersemesters (Wiederholungsmöglichkeit):
      Dienstag, der 3.4.2018, von 9 bis 11 Uhr





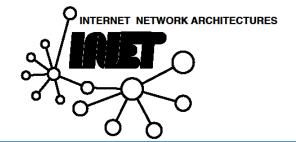
- ☐ ISIS für Vorlesungsmaterial am besten sofort
- OSIRIS für Übungsbetrieb am besten sofort

■Anmeldefrist IntroProg Tutorien via Moses:

18.10.2017 18:00

■ Modulanmeldefrist via QISPOS oder Prüfungsamt

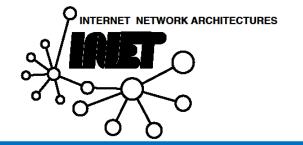
10.11.2017 18:00



## **C-Kurs - Blockveranstaltung**

**Organisation** 

Anja Feldmann, TU Berlin, 2017



## C-Kurs – Tagesablauf

#### 1. Vorlesung:

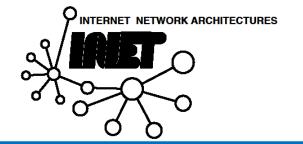
Vorstellung der Konzepte

#### 2. Tutorien:

Q&A (Fragen und Antworten) zu Vorlesungsinhalten

#### 3. Abgaben:

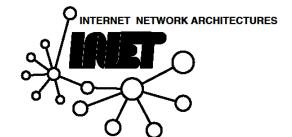
- Selbständig zu bearbeitende Programmieraufgaben
- Einzelabgaben



## C-Kurs – Tagesablauf

#### ☐ Zusätzlich:

- Betreute Rechnerzeiten in TEL 1. und 2. Stock
  - Hilfestellung beim Programmieren vor Ort
  - Genaue Zeiten und Orte: siehe ISIS
  - Generell von 10-18 Uhr jeden Werktag außer während der Vorlesung
- Arbeitsräume
  - Genaue Zeiten und Orte: siehe ISIS
  - Unterstützung per ISIS Forum
- > Helpdesk im TEL 109
  - Nur für organisatorische Probleme!



## Tagesablauf - Dienstag/Mittwoch

1. Vorlesung Di: 12:15 – ca. 13:45

2. Freischaltung der Aufgabe Di: ca. 14 Uhr

3. Betreute Rechnerzeit Di: 14 – 18 Uhr

4. Tutorien

1. Möglichkeit Mi: 12:00 – ca. 12:55

2. Möglichkeit Mi: 13:00 – ca. 13:55

3. Möglichkeit Mi: 14:00 – ca. 14:55

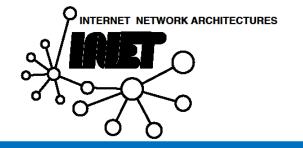
. . .

Details: siehe ISIS

5. Vorlesung Mi: 10:15 – ca. 11:45

6. Betreute Rechnerzeit Mi: 12 – 18 Uhr

7. Abgabe der 1. Aufgabe Do: 21:59



#### **Abgaben**

- Ausgabe:
  - ➤ Ab Dienstag 17.10 nach der jeweiligen Vorlesung (ISIS)
- ■Abgaben:

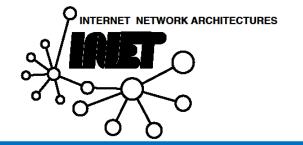
➤ 1. Blatt: Abgabefrist 19.10.2017 21:59 CET

> 2. Blatt: Abgabefrist 20.10.2017 21:59 CET

**>** ...

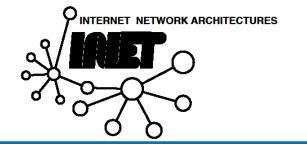
➤ 6. Blatt: Abgabefrist 26.10.2017 21:59 CET

> 7. Blatt: Abgabefrist 30.10.2017 09:59 CET



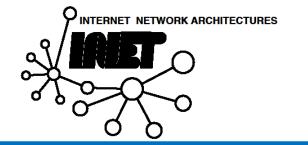
## **Tutorien Einteilung**

- Tutorien:
  - > Zum Teil mehr als 10 Tutorien parallel
  - > Thema: Aktuelle Vorlesung und Aufgabenblatt
    - Pro Thema gibt es mehrere Zeitwahlmöglichkeiten
    - Teilnahme an jedem Thema ist sinnvoll
  - > Ziel: ca. 30 Teilnehmer pro Tutorium



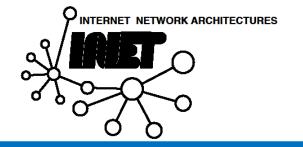
### **Tutorien Einteilung**

- ☐ Verteilungsalgorithmus für die Teilnehmer auf die Tutorien:
  - Verfahren:
    - Ausgabe der Tutorienplätze nach Zeit
    - Es gibt begrenzte Plätze und Zeitfenster!
    - Nur belegte Tutorien finden statt!
    - Überblick über die Angebote gibt es in ISIS
    - First come, first served!!! Wie im Kino....
  - > Jeweils im Anschluss an die Vorlesung! (ca. 30 Minuten)
  - ➤ Danach bis 18 Uhr bzw. von 10 11 Uhr im Tel 109 beim Helpdesk
- ☐ Bei Problemen am Helpdesk im TEL 109 melden



#### Abgaben – Wie?

- Mittels Versionsverwaltungssystem
  - > System zur Erfassung von Änderungen an Dokumenten oder Dateien
  - > Alle Versionen werden mit Zeitstempel und Benutzerkennung gesichert
  - Versionen können später wiederhergestellt werden
  - Versionsverwaltungssysteme werden u.a. in der Softwareentwicklung zur Quelltextverwaltung eingesetzt
- Wir verwenden:
  - Subversion (SVN) (u.a. weil GIT keine detaillierte Rechteverwaltung unterstützt)
- Details zur Benutzung von SVN in den Tutorien und auf ISIS



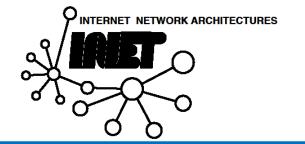
#### Einzelabgabe – Hinweise

#### Einzelabgabe

- Jede/r Studierende erarbeitet eine eigene Lösung und gibt diese ab!
- Diskussionen von Lösungswegen, Herangehensweisen, Hilfestellung sind erlaubt und sogar erwünscht!
- ➤ Aber Weitergabe von Lösungsteilen ist keine Hilfestellung, da das nicht dazu führt, ein eigenes Verständnis der Herangehensweise zu entwickeln!

#### Regeln

- Zwei identische Abgabeteile
  - ⇒ Eine ist ein Plagiat!
  - ⇒ Das ist ein Täuschungsversuch
  - ⇒ Beide Abgaben gelten als nicht bearbeitet, da generell der/die Originalautor/in nicht ermittelbar ist.
- Wiederholungsfall => Nicht bestehen wegen Täuschung

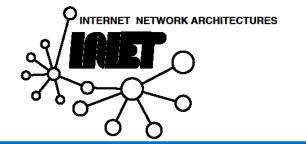


#### Identische Abgabeteile

- Abgaben werden als identisch betrachtet, wenn sie sich, u.a. nur in den
  - Variablennamen
  - Kommentaren
  - Einrückungen

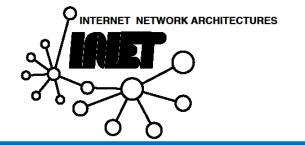
unterscheiden.

Hinweis: Wir benutzen Plagiatcheckertools! Zusammen mit manueller Überprüfung



### Acknowledgement

- □ Dank an:
  - > Randal E. Bryant und David R. Hallaron, CMU
  - > Christian Sohler, TU Dortmund



#### Literatur – Beispiele

- - > Kernighan, Programmieren in C, 1990
- Algorithmen und Datenstrukturen
  - ➤ Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C.: Introduction to Algorithms, 3. Aufl. MIT Press Cambridge, 2009
  - Sedgewick, R.: Algorithms in C, Addison-Wesley, 2005
  - Goodrich, M. Tamassia, R.: Data Structures and Algorithms in C++, John Wiley
- Systemsoftware
  - Randal E. Bryant, David R. Hallaron "Computer Systems: A Programmer's Perspective", Prentice Hall