Grundlagen der Informatik für Ingenieure I

19.1 Rechnerarchitektur

- Rechnerarchitektur
  - Funktionseinheiten der “von Neumann-Architektur”
    - *Rechenwerk*
    - *Leitwerk*
    - *Speicher*
    - *E/A-Werk*
  - Wichtige CPU-Register und ihre Funktion
  - Welche “Attribute” der Rechnerarchitektur spielen eine Rolle im Zusammenhang mit
    - *Wertebereichen*
    - *Genauigkeit*
    - *Adressierbarkeit*
    - *Typzugehörigkeit primitiver Datentypen*
  - Leistungsmaße für Prozessoren
  - Speicherhierachien
    - *“Geschwindigkeitsproblem”*
    - *“Adressraumproblem”*

19.2 Betriebssysteme

- Betriebssystemklassen
  - Batchsysteme, Realtime Systeme, Time Sharing Systeme
- Grobe Gliederung
  - Systemschnittstelle (API), System-Call-Handler
  - Dateisystem
  - Prozessverwaltungssystem
  - Speicherverwaltungssystem
  - E/A-System
- Ereignisse
  - asynchrone
  - synchrone
  - Interrupt
  - Trap

19.2.1 Dateisysteme

- Darstellung
  - Kataloge
  - Dateien
  - Links
  - Zugriffsrechte
  - Dateihierachie
- Organisation (Datenstrukturen) und dynamischer Ablauf
  - E/A-Systemcalls
  - Dateikontrollstrukturen (inodes, filetables)
  - Puffersysteme
- Plattenorganisation

19.2.2 Prozessverwaltungssystem

- Prozesse, Threads
  - Schutzumgebungen
  - Zustände, Scheduler
  - Interupts; Traps; SVC
  - Erzeugung/Ausführung eines Prozesses unter Unix (Prinzipskizze)
  - Prozesshierachie
  - Wechselwirkungen
    - *Speicherverwaltung*
    - *Dateiverwaltung/ E-A-Verwaltung*
  - Prozesshierachie

19.2.3 Speicherverwaltungssystem

- Strukturierung von Speichern
  - physikalischer Adressraum/ logischer Adressraum
  - Relocation
  - Bit, Bytes, Worte, Segmente, Kacheln
  - Codesegment/Datensegment/Stacksegment
- Verfahren der Speicherverwaltung/Hardwareunterstützung
  - Vergabestrategien
  - Segmentierung/Paging
  - Virtueller Speicher
  - Ein/Auslagerungsstrategien

19.2.4 Rechnernetze

- Netzwerk-Protokolle
  - was ist grundsätzlich ein Netzwerkprotokoll?
  - sichere und unsichere Datenübertragung; Flusskontrolle
  - Client, Server
  - Adressierung und Routing auf IP-Ebene; Routingtabellen, DNS, ARP
  - Dienste, Services, Ports
  - NORMA-Architektur

19.3 OO Programmentwicklung

19.3.1 Klassen

- Was ist eine Klasse?
- Klassenhierachien
- Vererbung *(Inheritance)*
  - *Single ...*
  - *Multiple ...*
  - *(Wie und) was erbt man implizit?*
  - *Wie erbt man explizit?*
- Klassenvariable
- Klassenmethoden
  - *Modifier*
  - *Zugriff auf Variable*

19.3.1 Klassen

- Klassendefinition
  - *extends*
  - *implements*
  - *Modifier*
    - *public*
    - *abstract*
    - *final*
  - *Konstruktoren*
  - *default Constructor*
  - *default finalizer*

19.3.2 Objekte/Instanzen

- Was ist ein Objekt, eine Instanz?
- Kreieren eines Objekts, Instantiieren
  - *Template --> Objekt*
  - *Referenz auf das Objekt*
  - *Constructor*
  - *Variable*
  - *Code vs. Daten*
  - *Speichermanagement*
- Objektzustand
- Instanzvariable
  - *Gültigkeitsbereich*
  - *Modifier*
  - *Access Controll*

19.3.3 Methoden und Konstruktoren

- Methoden
  - Definition
    - *Signatur*
    - *Returntyp*
    - *Access Control*
  - Parameterübergabe
    - *Call by Reference*
    - *Call by Value*
  - *Modifier*
    - *abstract*
    - *static*
    - *(native)*
    - *final*
    - *synchronized*

19.3.3 Methoden und Konstruktoren

  - *Overriding*
  - *Overloading*
- Konstruktoren
  - *Overriding*
  - *Overloading*
  - *super()*
  - *super(p1, p2,...)*
- Referenzen
  - *this*
  - *this()*
  - *this.name*
  - *return[value]*
  - *super.methodename()*

19.3.3 Methoden und Konstruktoren

- Ausgezeichnete Methoden
  - *main()*
  - *init()*
  - *run()*
- Referenzen auf Objekte
- Zugehörigkeit eines Objekts zu einer Klasse
- Relationale Operationen auf Objekte

19.3.4 Variable und Konstanten

- Namenskonventionen
- Variable, Konstante
  - *Definition*
  - *Datentypen*
  - *Namenskonstante, Literale (Konstante)*
  - *Modifier*
  - *Initialisierung*
  - *Gültigkeitsbereiche*
  - *Wertebereiche (Wortbreite)*
  - *Genauigkeit (Wortbreite)*
  - *Lebensdauer*
  - *Casting*
  - *Converting*

19.3.5 Datentypen, Operatoren

- Primitive Datentypen
  - *Typgruppen*
  - *Typen*
  - *Wertebereiche*
  - *Operatoren*
    - arithmetische
    - bool'sche
    - relationale
    - Vergleichsoperator als spez. Statement
    - *Casting*
    - *Converting*
    - Objekte prim. Datentypen
    - *arrays*
    - *strings*

19.3.6 Java-Sprachkonstrukte

- Quellcodelayout
- Strukturierung
- Statement
- Block of Statements
- Flusskontrolle
  - *if ..*
  - *if...then...*
  - *if...then...else*
  - *while...*
  - *do ... while*
  - *for....*
  - *break*
  - *continue*
  - *switch*
  - *label*

19.3.6 Java-Sprachkonstrukte

- Regeln
  - *Precedenceregeln*
  - *Klammerungen*
  - *Aussprünge aus Schleifenkonstrukten*
  - *Schreibweisen von*
    - Klassennamen
    - Variablennamen
    - Methodennamen
    - Konstantennamen
    - Literalen

19.3.7 Modifier

- *default, public, private, final, static, protected, abstract, synchronized, (volatile), (native)* angewandt auf:
  - *Variable*
  - *Instanzvarialble*
  - *Klassenvariable*
  - *Methoden*
  - *Klassen*
- Schutz-(Protection) Konzept
- Kapselung
- *“Need to Know” - principle*
- *Access Methods*

19.3.8 Applets und Applikationen

- Applets
  - *Eigenschaften*
  - *Voraussetzungen*
  - *Anwendungsumfeld*
  - *Restriktionen (Security)*
- Applikationen
  - *Eigenschaften*
  - *Voraussetzungen*
  - *Anwendungsumfeld*

19.3.9 Exceptionhandling

- Worum geht es beim *Exceptionhandling?*
- Gruppierung von *Exceptions*
  - *synchron, asynchron*
  - *Error, Exceptions*
  - *Runtime-Exceptions*
  - *implizit, explizit*
  - *I/O-Exceptions*
- Sprachkonstrukte und ihre Bedeutung
  - *try {...}*
  - *catch {...}*
  - *throw*
  - *(...throws)*
  - *(“rethrowing”)*
  - *finally*

19.3.10 Interfaces

- Was ist ein Interface?(vs. abstracte Klassen)
  - *Hierachiebeziehungen*
  - *nur abstrakte Methoden*
  - *Namenskonstante sind möglich*
  - *implements*
    - vollständige Implementierung
  - *mehrfache Interfaces*
  - *Interfacedefinition*
  - *Beispiele*
    - *Runnable (threads)*
    - *xxxxListener*
    - *LayoutManager*
    - *DataInput*
    - *DataOutput*

19.3.11 Java-E/A-System

- Streamskonzept
  - *Inputstreams*
  - *typische Methoden*
    - Reader
    - InputStream
  - *Outputstreams*
  - *typische Methode*
    - Writer
    - OutputStream
  - *Ein/Ausgabe auf Dateien*
  - *Ein-/Ausgabe auf Sockets*
- Standard-E/A
- Formatierte E/A
  - Random Access Dateien

19.3.12 Threads

- Was sind *Threads*?
- Threadkontrolle
  - *run()*
  - *start()*
  - *yielt()*
  - *sleep()*
  - ...
- Threadanwendungen

19.3.13 Abstract Window Toolkit

- Was ist das AWT?
  - *Basiskonstrukte*
  - *Layoutkonstrukte*
  - *Windows*
- ...in Applets
- ...in Applikationen
- *Eventhandling*
  - *Listener*
  - *Events*

19.4 Datenstrukturen

- Datenstrukturen
  - Pointer (Zeiger) in C und Java
  - Felder(arrays)
  - Verkettete Listen
  - Warteschlangen, Kellerspeicher (wo einschlägig?)
  - Suchen und Sortieren
  - Gestreute Speicherung (Hashing)

19.5 Parallele und Verteilte Systeme

- Kassifikation Paralleler und Verteilter Systeme
- Synchronisation und Koordinierung
- Multithreading
- Sockets

19.6 Allgemeines

- Was versteht man unter dem Begriff *JavaVM*?
- Was versteht man unter einer interpretativen Abarbeitung?
  - *Welche Vorteile*
  - *Welche Nachteile*
- Was ist eine Klassenbibliothek?
- Was ist ein Package?
- Was verspricht man sich vom Einsatz des OO-Programmiermodells?
- Wie erreicht man die Kompatibilität von Java-Programm-Systemen?
- *Byte Code*
- Was ist ein *“Just in Time Compiler”*?
- Was versteht man unter “Garbage Collection”?
- Java und Security

19.7 Notizen