

Einführung in die Informatik I
Klausurvorbereitung

Zentralübung am
5. Februar 2001

Dr. Andreas Harrer
Technische Universität München

Wintersemester 2000/2001

Übersicht

- ❖ **Organisatorisches**
- ❖ **Was wird in der Klausur gefordert?**
- ❖ **Wie komme ich mit der Klausur klar?**
- ❖ **Programmierstile**
- ❖ **Fragen?**

Organisatorisches:

- ❖ **Vorlesung einschließlich letzten Dienstag bei Fachschaft**
- ❖ **Übungsblätter und Lösungen als Kopiervorlage hier nach der Zentralübung**
- ❖ **Glossar (noch nicht vollständig) im Web**
- ❖ **Wiederholungsklausur am 20. April 2001**
- ❖ **Betreute Lerngruppen als Vorbereitung (von Studenten für Studenten)**
- ❖ **Sprechstunden mit der Übungsleitung nach Vereinbarung**

Anforderungen der Klausur:

- ❖ **Verständnis der Konzepte und Verfahren**
 - Auswendiglernen ist nicht gefragt, sondern Verstehen!
 - Kein Multiple Choice
- ❖ **Eigenständiges Problemlösen**
- ❖ **Implementieren bzw. Programmieren der Ideen**
 - Syntax ist nicht das Wichtigste
- ❖ **Unsere Empfehlungen zur Vorbereitung:**
 - Durcharbeiten der Übungsblätter
 - Lauffähige Programme schreiben

Wie gehe ich mit Unterlagen um?

❖ **Unterlagen**

- **Vorlesungsfolien (Fachschaft)**
- **Übungsblätter und Lösungen (Web oder Kopiervorlage)**
- **eigene Notizen**
- **evtl. Java-Buch**
- **evtl. Glossar zur Vorlesung**
- **Wörterbuch**

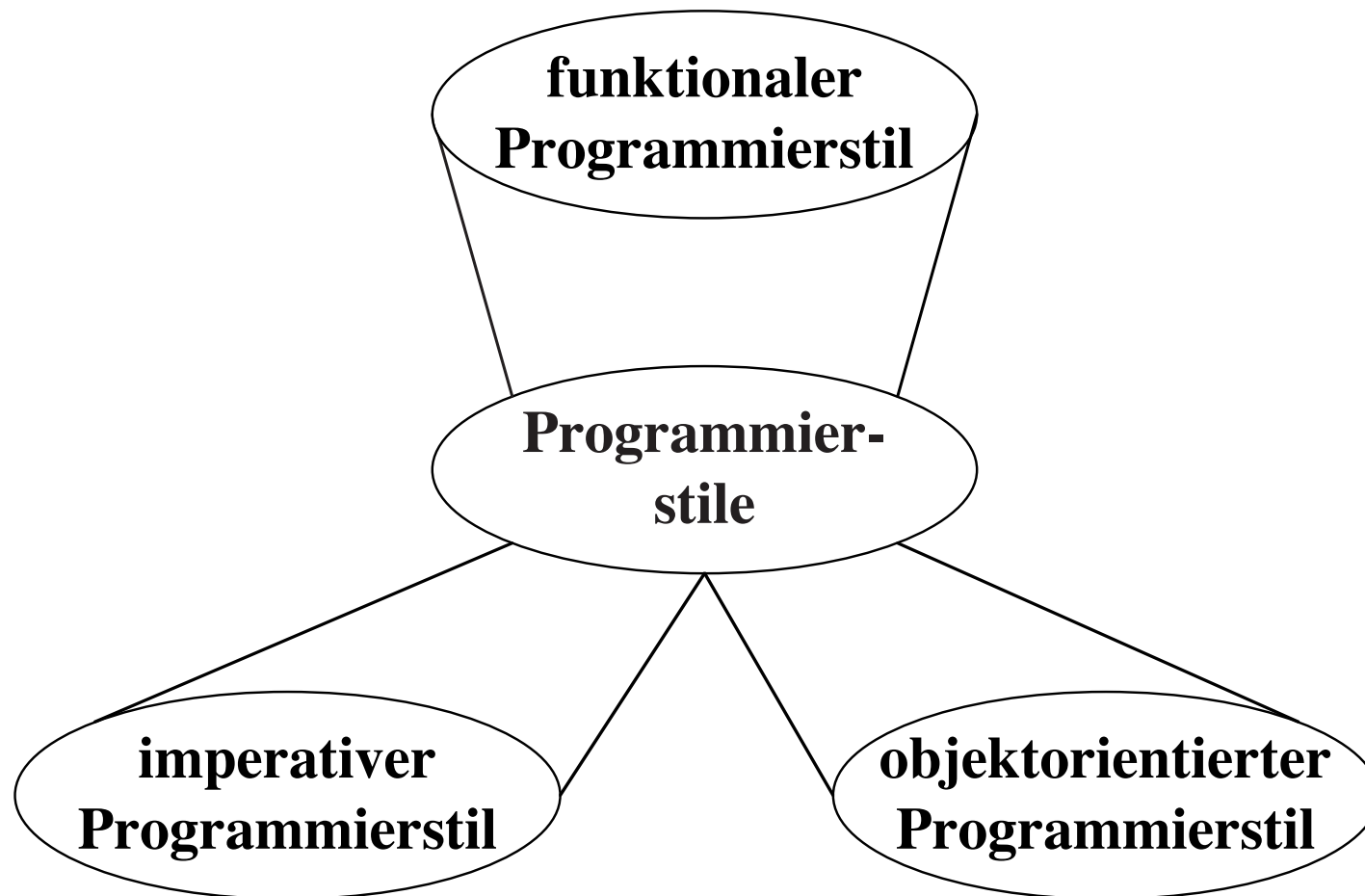
❖ **Organisation der Unterlagen ist wichtig**

- **Ordner anlegen**
- **Index, um Begriffe schneller zu finden**
- **wildes Blättern kostet viel Zeit!**

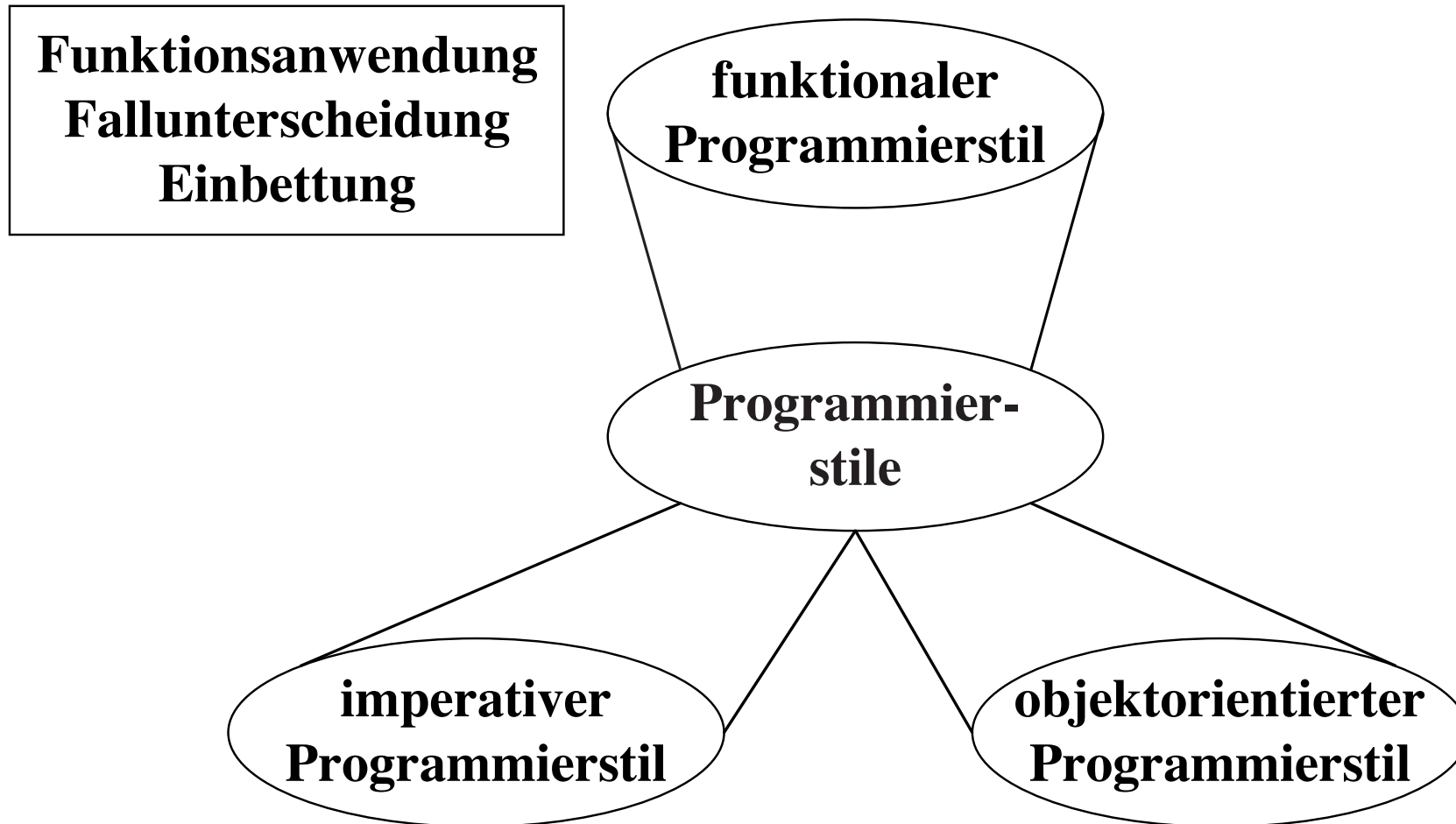
Wie sieht die Klausur aus?

- ❖ **Dauer 180 Minuten**
- ❖ **40 Punkte, 17 Punkte nötig für das Bestehen**
- ❖ **DVP-, Bachelor- und Scheinklausur gleich**
- ❖ **Angabe:**
 - Aufgabe 1 (2+2=4 Punkte) Titel**
- ❖ **Reihenfolge der Aufgaben selbst wählen**
- ❖ **Aufgaben unabhängig voneinander lösbar
(auch Teilaufgaben)**

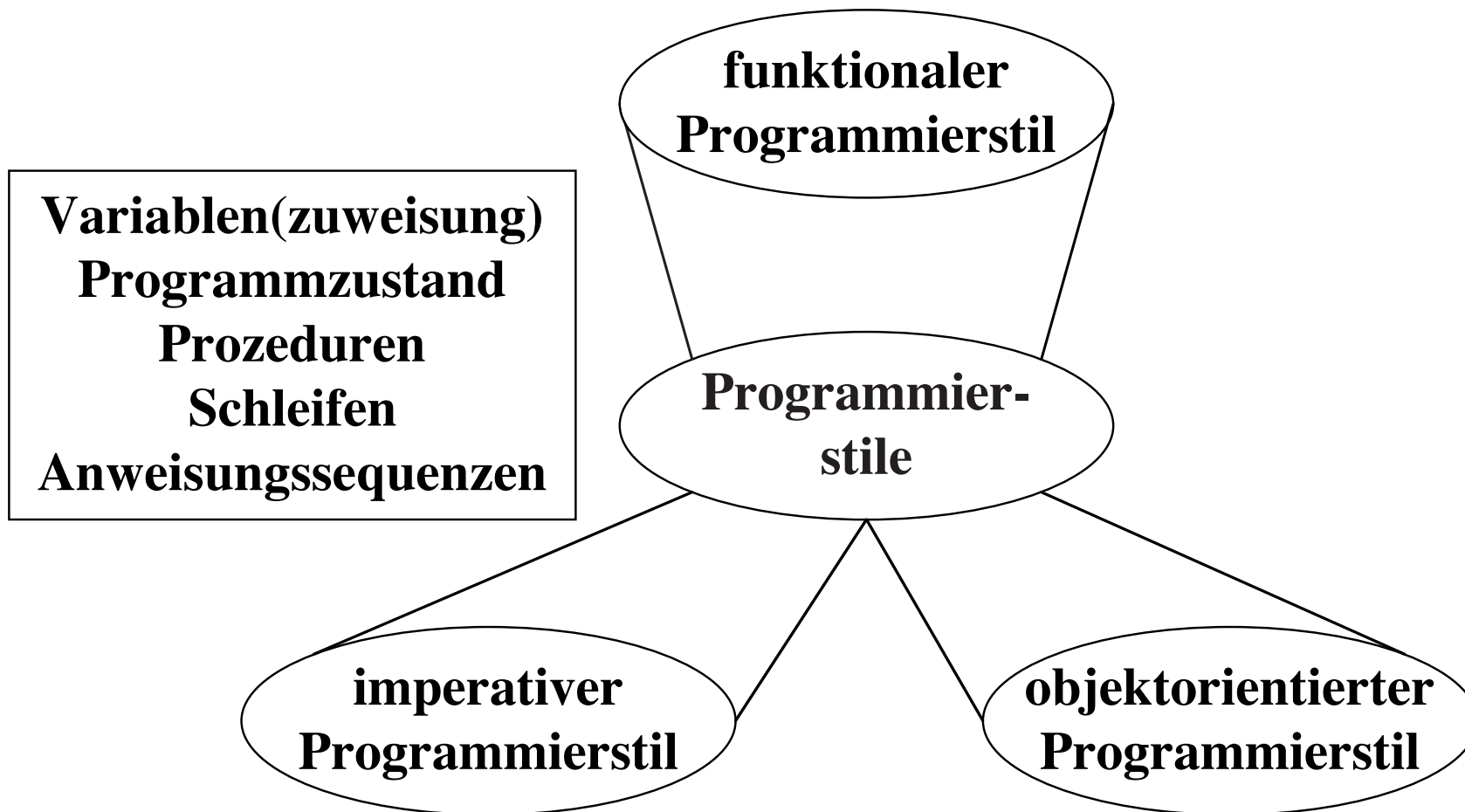
Programmierstile:



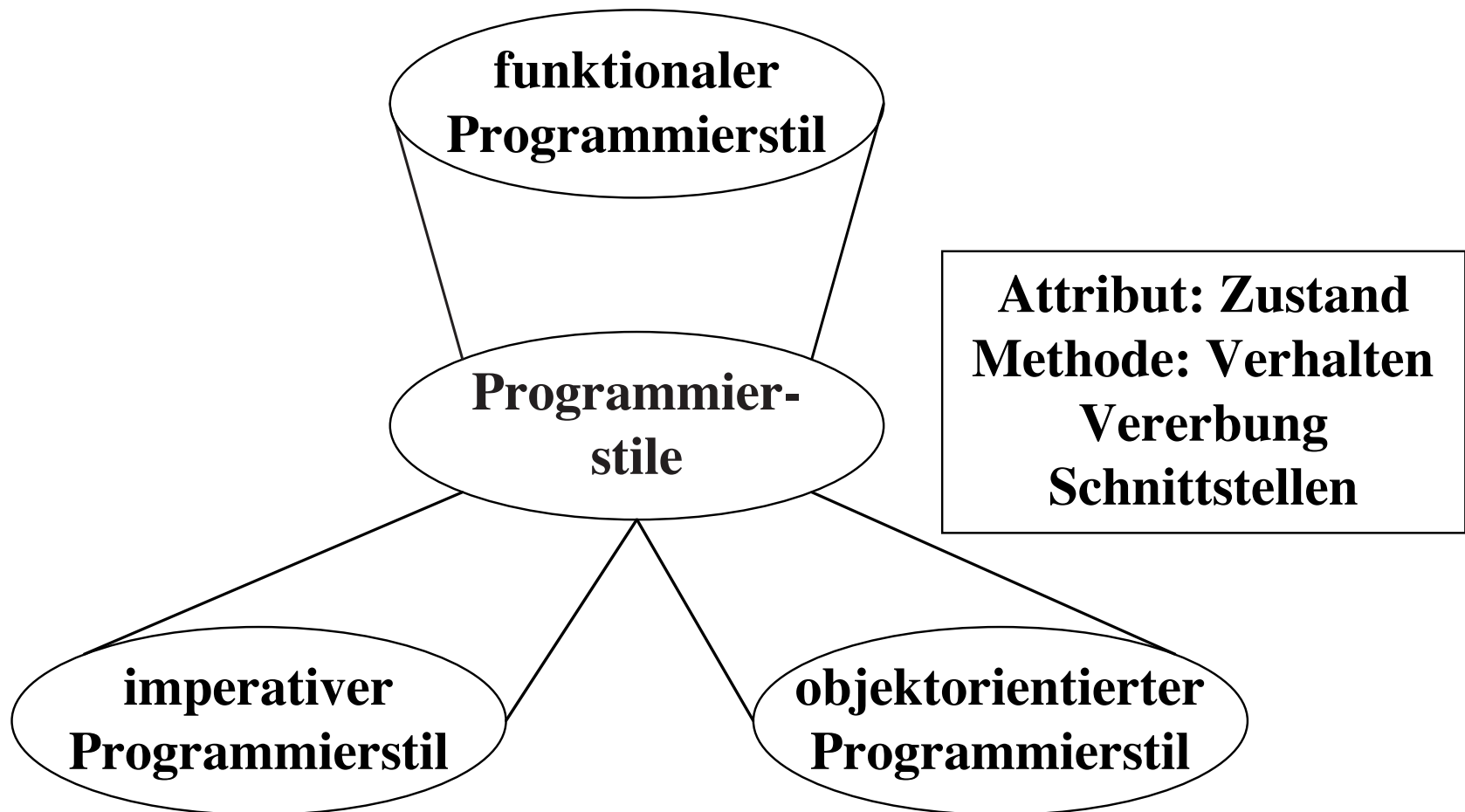
Programmierstile:



Programmierstile:



Programmierstile:



Die last-Operation: funktionaler Programmierstil

```
int last(IntSequenz s) {  
    return (isEmpty(rest(s)) ?  
           first(s) :  
           last(rest(s)));  
}
```

Die last-Operation: imperativer Programmierstil

```
int last(IntSequenz s) {  
  
    IntSequenz temp=s;  
    while (! isEmpty(rest(temp))) {  
        temp = rest(temp);  
    }  
    return first(temp);  
  
}
```

Die last-Operation: objekt-orientierter Programmierstil

```
class IntListe {  
  
    int wert;  
    IntListe restliste;  
  
    int last() {  
  
        if (restliste == null)  
            return wert;  
        else return restliste.last();  
    }  
}
```

Das war's von meiner Seite. Noch Fragen?

**Viel Erfolg und
alles Gute für
die Klausur!**