

# **3. + 4. Übungsblatt**

## **Abstraktion, Modultypen und OO**

### **Erweiterung des Entwurfs**

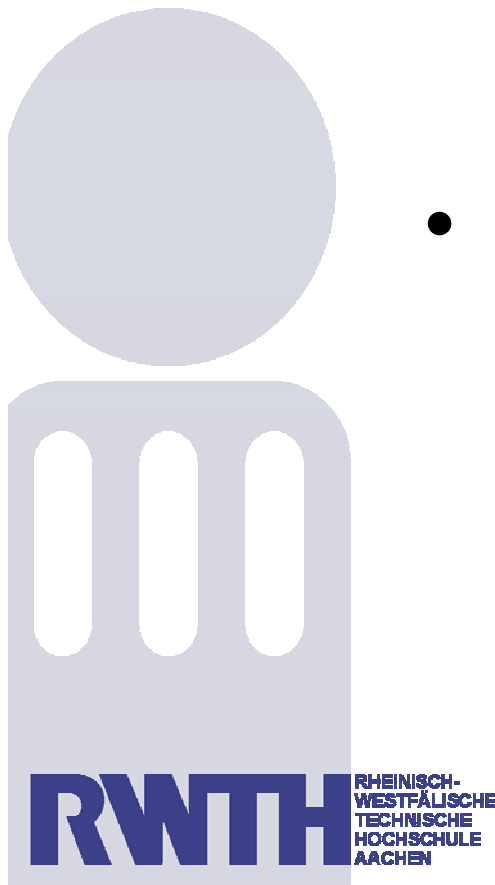
### **Benutzbarkeitsbeziehungen**

11. Mai 2006

Dipl.-Inform. Christian Fuß

# Funktionale Abstraktion vs. Datenabstraktion

- Funktionale Abstraktion
  - Datentransformation, Steuerung
  - keine Datenhaltung
  - Funktionales Modul, z.B. Parser
  - Funktionales Typmodul, z.B. Task zur parallelen Bearbeitung von Aufgaben
- Datenabstraktion
  - Information Hiding, Datenabstraktion
  - Datenhaltung + Zugriffsmethoden
  - Abstraktes Datenobjekt: Einzelne Instanz, z.B. Laufzeitkeller, Datenbank-Layer
  - Abstraktes Datentypmodul: Schablone zur Erstellung mehrerer Instanzen, z.B. Eintrag in einer Kollektion

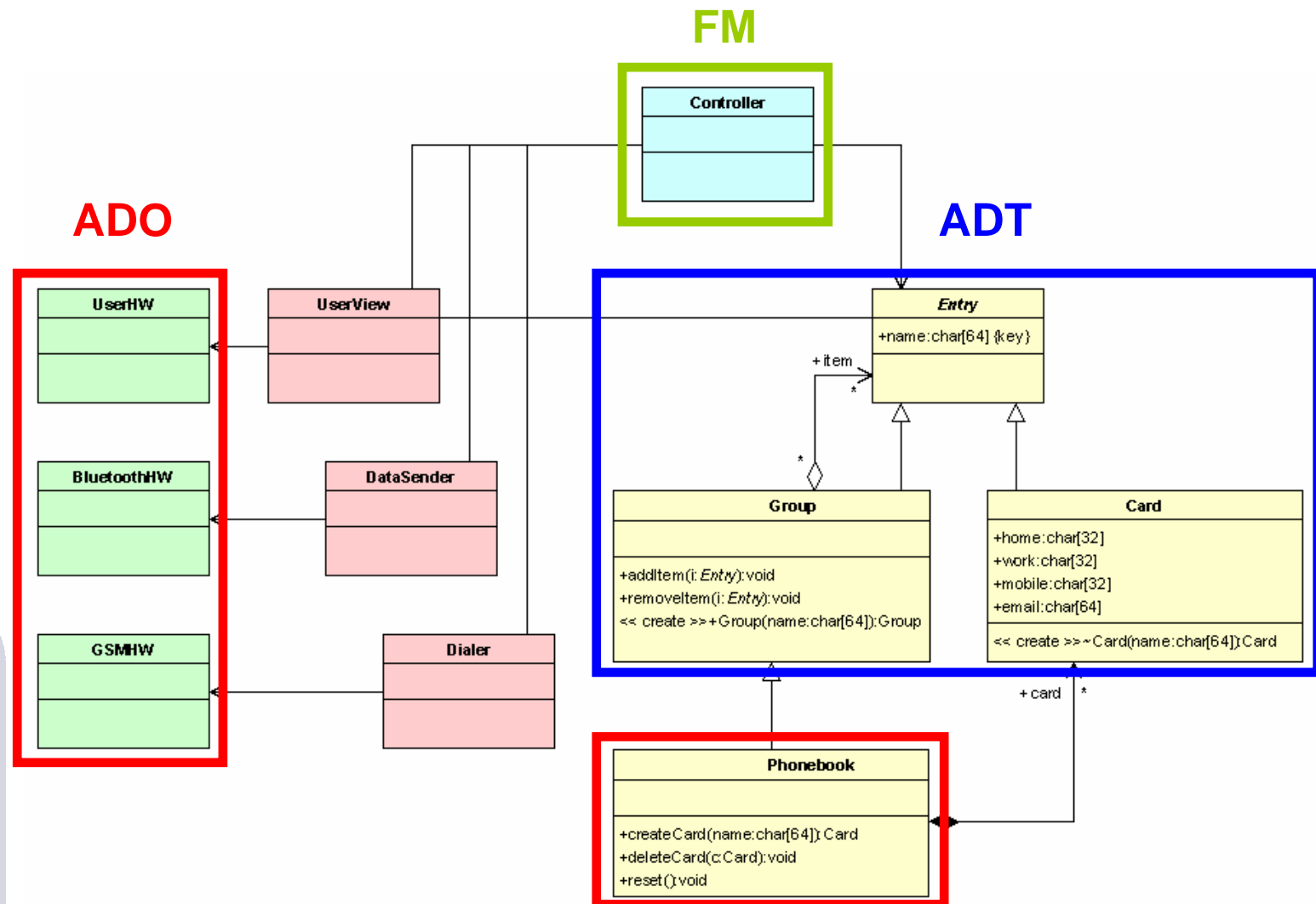


# Modultypen und OO

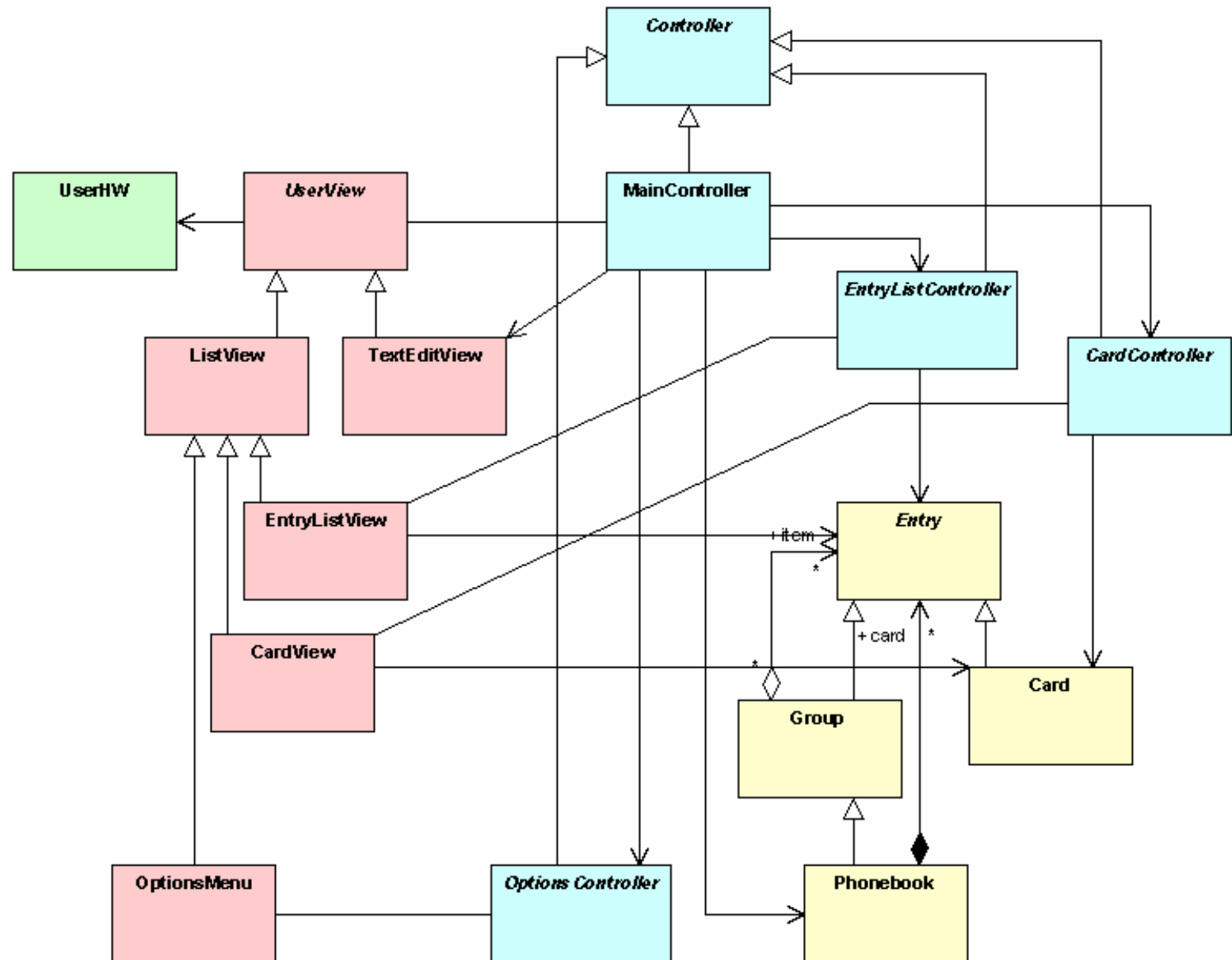
- OO kennt oft nur Klassen, d.h. ADTs
- Möglichkeiten bestehen in informeller Kennzeichnung und entsprechender Disziplin
  - ADO als *Singleton-Klasse*
  - FM als abstrakte Klasse ohne Attribute lediglich mit statischen Methoden
  - FTM benötigt zusätzliche Sprachkonstrukte, z.B. Task
- C-Kompatibilität von C++ erlaubt auch logische Konstrukte wie ADO oder FM direkt abzubilden



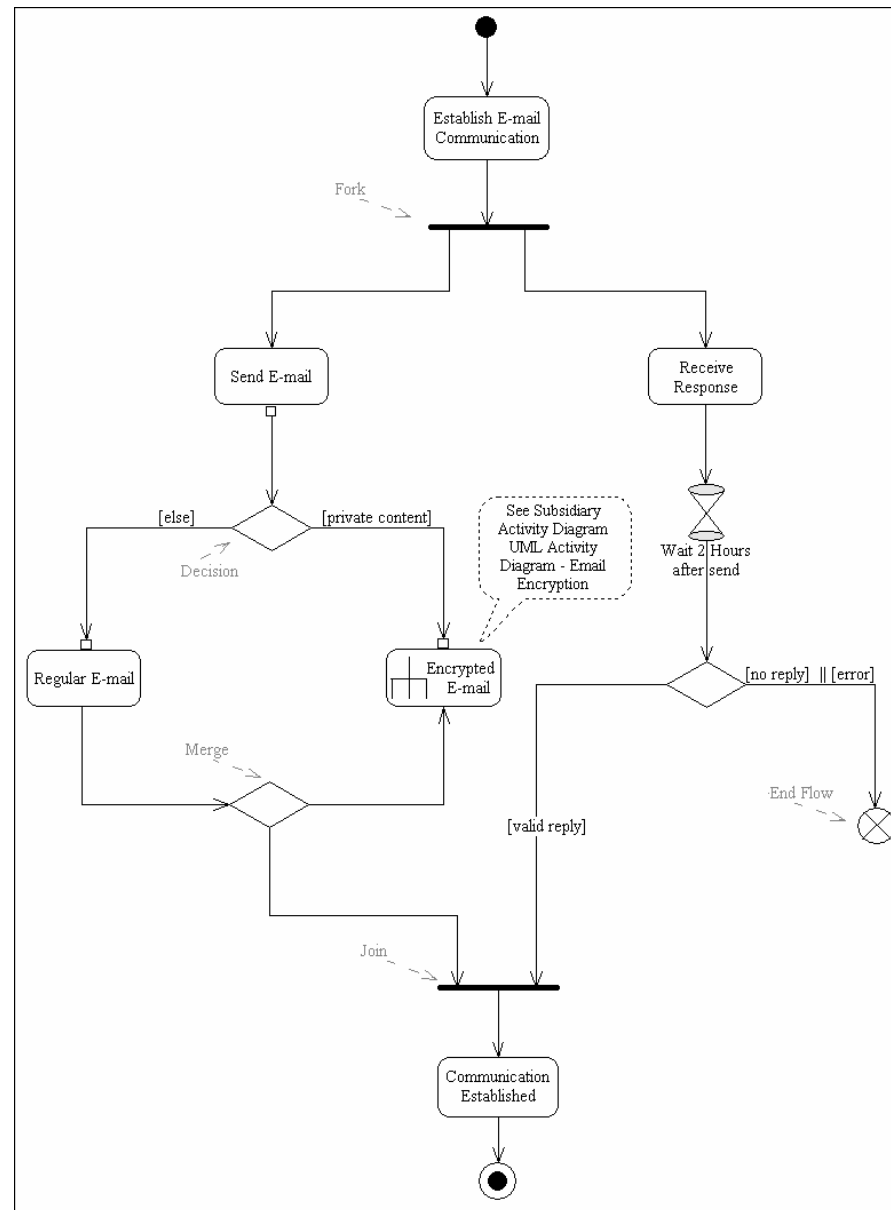
# Modultypen im UML-Entwurf



# Verfeinerung des MVC-Entwurfs



# Anwendungsablauf



# Daten- und Kontrollfluss in einer Architektur

- Daten- und Kontrollfluss nur entlang der Benutzbarkeitskanten **möglich**
- Aussage über Richtung des Flusses ist nicht möglich
- Unterscheidung zwischen Kontroll- und Datenfluss nur eingeschränkt möglich (anhand der Module)
- Benutzung nur entlang von Benutzbarkeitskanten, die wiederum Beschränkungen folgen, welche in der *potentiellen* Benutzbarkeit ausgedrückt sind
- weder Benutzung noch potentielle Benutzbarkeit finden sich explizit im Architekturdiagramm



# Potentielle lokale Benutzbarkeit

Möglichkeiten wie in blockstrukturierten Sprachen

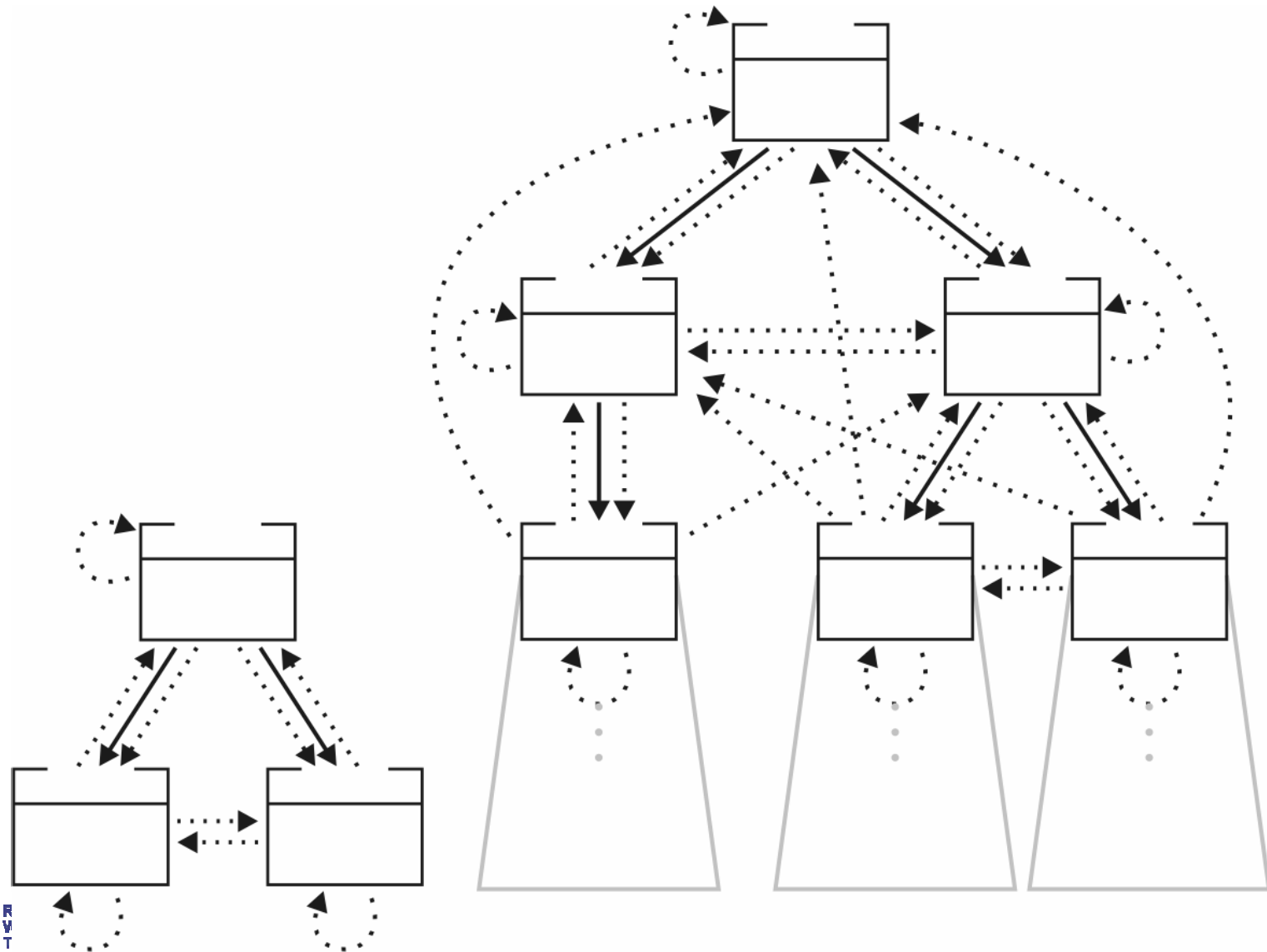
Modul M darf folgendes benutzen:

- sich selbst
- seine Kinder
- beliebigen Ahnen
- direkte Nachfolger eines beliebigen Ahnen

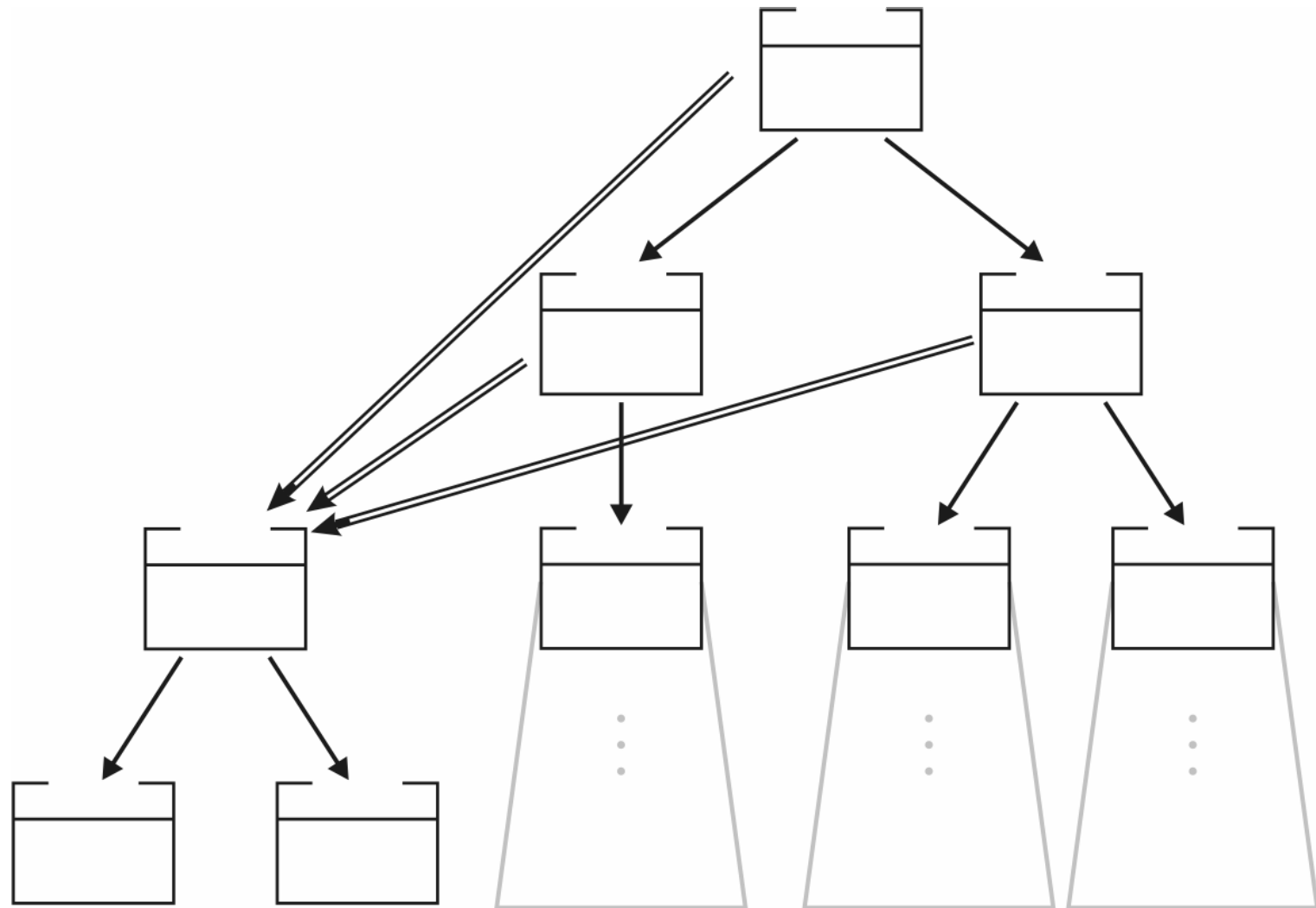




# Potentielle lokale Benutzbarkeit



## Potentielle allgemeine Benutzbarkeit



# Benutzbarkeitsbeziehung

|     | Schnittstelle          | Rumpf  |
|-----|------------------------|--|
| adt | • Typen von Parametern | <ul style="list-style-type: none"> <li>• Erzeugung von Instanzen</li> <li>• Zugriffsoperationen</li> </ul> |
| ado | ∅                      | • Zugriffsoperationen  |
| f   | ∅                      | • Operationen  |
| tk  | • Typen von Parametern | • Typen in der Implementierung   |

