

Programmieren und Modellieren

Jens Kosiol

Wintersemester 2023/24

Organisatorisches

- 6 CP (Vorlesung und Übung)
- Vorlesung
 - Jens Kosiol (meistens)
 - Dienstag, 16:15 – 17:45 Uhr, Raum -1319
- Übung
 - Clemens Emme und Natascha Nolte
 - Freitag, 10:00 – 11:30 Uhr, Raum -1606
- Prüfungsleistung
 - Bearbeitung der Übungsaufgaben ist verpflichtend
 - Benotete Prüfungsleistung ist ein Abschlussprojekt (2 Wochen)
- Materialien
 - Links zu Folien, Übungsblättern, Videos (ohne Garantie!!!) über den SE-Blog (<https://seblog.cs.uni-kassel.de/ws2324/programming-and-modelling/>)
 - Fragen und Diskussion über Discord: <https://discord.gg/WbUAxaPdMC>

Inhalt

- Objektorientierte Modellierung und Programmierung
 - Story Driven Modeling als Vorgehensmodell
 - Unified Modeling Language (UML)
- Testen von Software
 - Test-first Prinzip
 - Testprinzipien
 - JUnit als Testframework
- Entwicklung graphischer Nutzeroberflächen (GUIs)
 - Wireframes und MockUps als Entwurfsmuster
 - JavaFX als Technologie
 - Model–view–controller als Design Pattern
- Programmieren von Client–Server–Anwendungen (Einführung)
 - HTTP, REST, WebSockets

Hintergrund

- Konzept der Vorlesung (und manche Folien) gehen auf Albert Zündorf zurück
- Buch zur grundlegenden Methodik: Norbistrath, Zündorf und Jubeh: *Story Driven Modeling* (CreateSpace Independent Publishing Platform 2013)
- Einige Folien gehen auch auf Gabriele Taentzer (Universität Marburg) zurück

Datentransparenz

- Organisation und Abgabe der Hausaufgaben geschieht über *GitHub* (<https://github.com/>), das zu Microsoft gehört. Die Repositories sind aber privat (nur Besitzer, Betreuer und Korrektoren haben Zugriff).
- Zur Unterstützung der Hausaufgabenkorrektur benutzen wir ein selbstentwickeltes Tool, das ChatGPT (OpenAI) verwendet. Die Prompts enthalten **Dateipfade** und **Codeschnipsel**. **Verwenden Sie dort keine Namen!**

Datennutzung für Forschung

Wir wollen in diesem Semester den Nutzen unseres Tools zur Unterstützung der Hausaufgabenkorrektur evaluieren. Dazu veranstalten wir wöchentlich kurze Umfragen (online) zu Ihrer Zufriedenheit mit der Korrektur der Übungsaufgaben.

- Die Teilnahme ist freiwillig.
- Regelmäßige Teilnahme an den Umfragen wird mit Bonuspunkten für das Abschlussprojekt belohnt.
- Die erhobenen Daten werden anonymisiert ausgewertet.
- Die erhobenen Daten sollen die Grundlage für eine wissenschaftliche Veröffentlichung bilden.

Einstieg in die Modellierung: Beispiele und Szenarien/Stories

Jens Kosiol

Wintersemester 23/24

Warum modellieren?

Im Allgemeinen dient Modellierung der **Reduktion von Komplexität**.

- **Vereinfachte** Sicht auf diejenigen **Aspekte** eines komplexen Phänomens, die für eine Aufgabe von Bedeutung sind
- **Eingeschränkte** Sicht auf einen **Ausschnitt** der Realität, der untersucht werden soll

Anhand von Modellen kann man Informationen und Erfahrungen sammeln, Schlüsse ziehen, Vergleiche anstellen, Alternativen bewerten, Entscheidungen treffen, Strategien entwickeln, etc., **die für die Bearbeitung des Originals von Nutzen sind** und sich an diesem selbst nicht (so leicht) vollziehen lassen.

Warum in der Informatik modellieren?

- Dinge aus der Realität in Datenstrukturen abbilden, die ein Computer verarbeiten kann
- Kompakter(er) Überblick über komplexes Gesamtsystem
- Modularisierung: Zerlegen eines komplexen Gesamtsystems in handhabbare(re) Einzelteile
- Kommunikation
 - Mit Stakeholdern
 - Zwischen Entwicklern
- ...

Modellierung: Kernkompetenz der Informatik

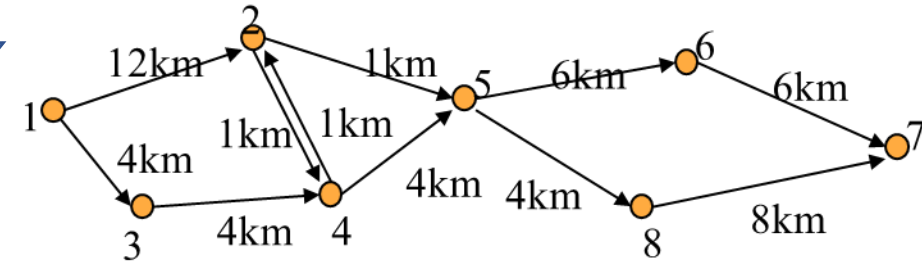
Realität



Modellierung

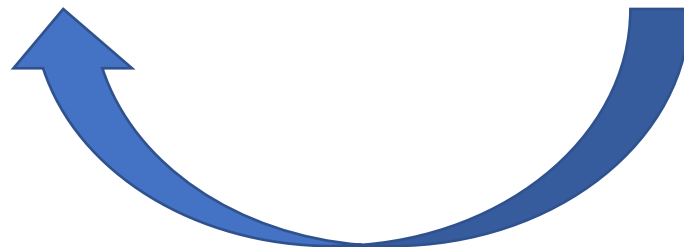


Modell



dist[1,2] = 12;
 dist[1,3] = 4;
 dist[2,5] = 1;
 dist[2,4] = 1;
 dist[3,4] = 4;
 ...
 dist[8,7] = 8;

result = findDist(1,7);



Interpretation

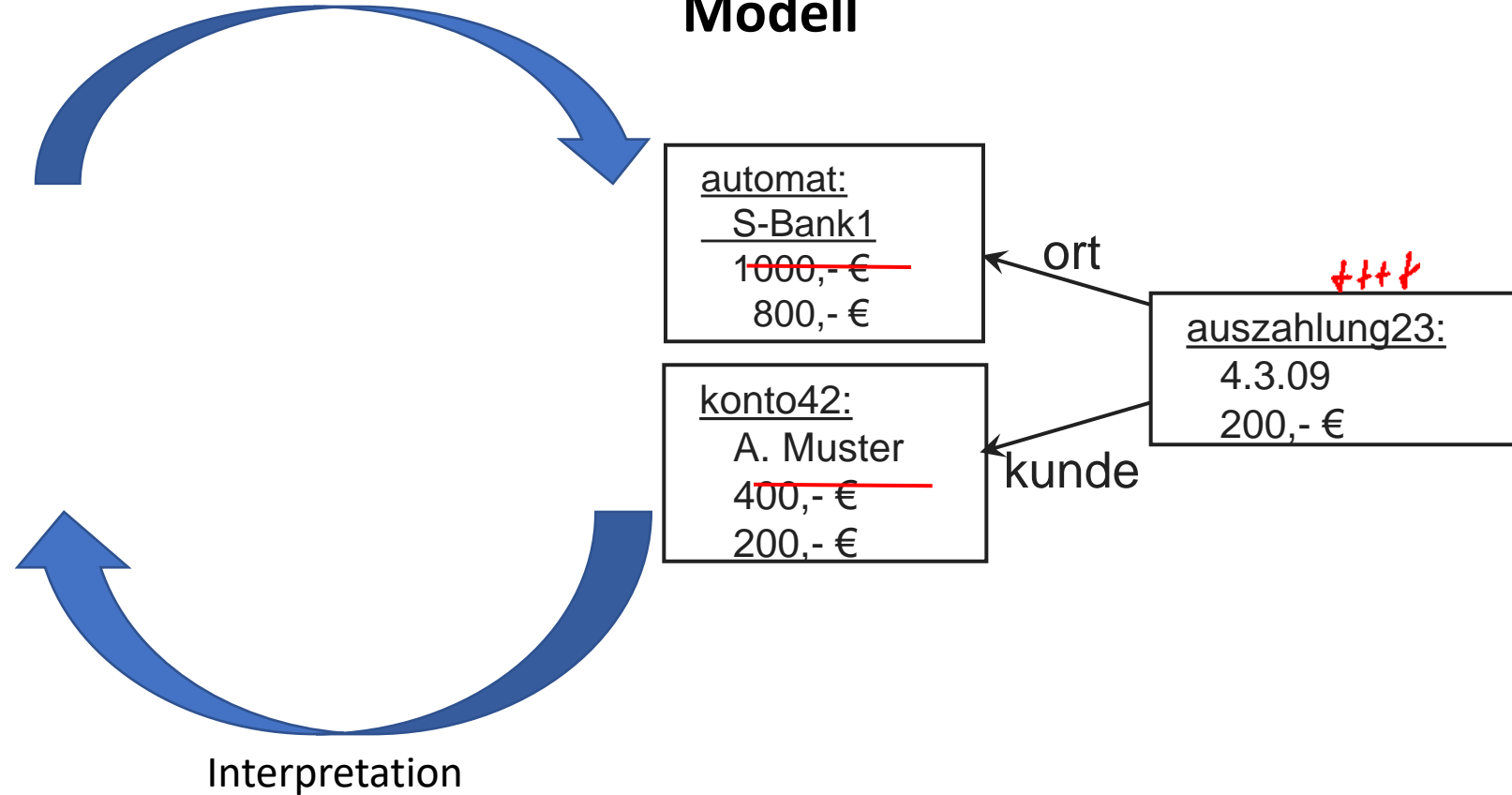
Modellierung: Kernkompetenz der Informatik

Realität



Modellierung

Modell



Generelles Vorgehen

Stories/Scenarios/Beispiele

- Stammen vom Kunden
- So konkret wie möglich
- Grundlage für Objektdiagramme und für Tests



Objektdiagramme

- Dienen der Kommunikation mit Kunden und Entwicklern
- Snapshot des Heap zur Programmlaufzeit
- Grundlage für Klassendiagramm

<u>konto42: Konto</u>
kunde = "A. Muster"
stand = 400



Klassendiagramme

- Dienen der Kommunikation zwischen Entwicklern
- Muster für Datenstruktur
- Grundlage für Codegenerierung

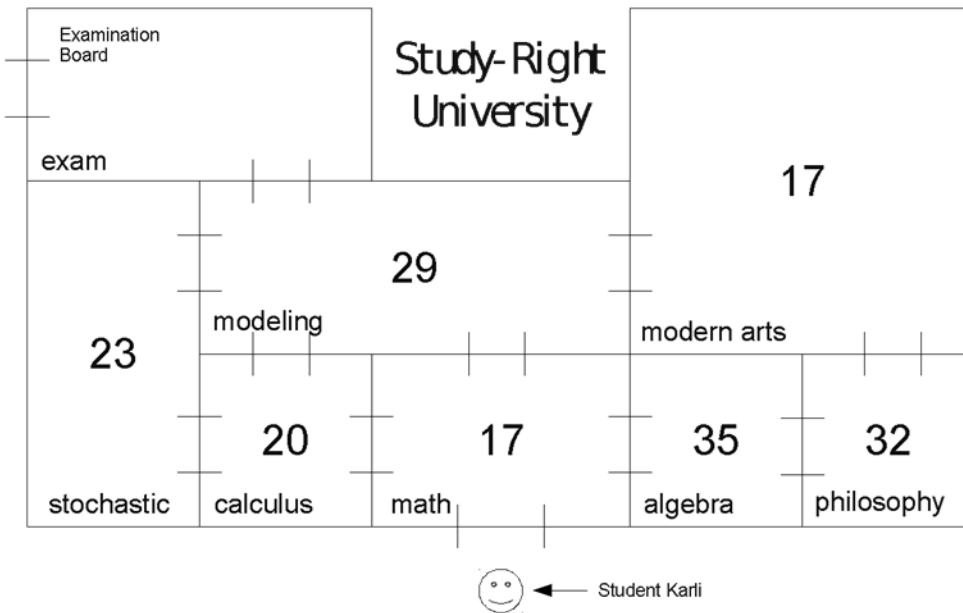
<u>Konto</u>
kunde: String
stand: float



Code

```
public class Konto {
    public String kunde;
    public float stand;
}
```

Vorlesungsbeispiel: Study-Right University



Aufgabe:

- Wegesuche (vom Mathe- zum Examensraum)

Bedingungen:

- Jeder Credit Point kostet einen Motivationspunkt
- 214 Motivationspunkte gegeben und 214 Credit Points zu erreichen
- Modulabhängigkeiten (Türen zwischen Räumen)
- Mehrfachbelegung möglich (bei Mehrfachbetreten eines Raumes wird jeweils ein anderes Modul unterrichtet)

Regeln – abstrakt

Würfel.



- Die Rohstoffkarten werden in 5 Stapel sortiert und offen in die Fächer der Kartenhalter gelegt. Die Kartenhalter werden neben dem Spielfeld bereitgestellt.



- Die Entwicklungskarten (→) werden gemischt und verdeckt in das freie Fach des Kartenhalters gelegt.



- Zuletzt erhält jeder Spieler für die mit einem Stern markierte Siedlung (siehe S. 1) die ersten Rohstofferträge: Für jedes Landfeld, das an diese Siedlung angrenzt, nimmt er sich eine entsprechende Rohstoffkarte vom Stapel. *Beispiel: Blau erhält für seine Siedlung ganz oben je 1 Karze Holz, Erz und Wolle.*
- Seine Rohstoffkarten hält jeder Spieler verdeckt in der Hand.

Zusätzlich: Er kann zu einem beliebigen Zeitpunkt seines Zuges (auch vor dem Würfeln) eine seiner Entwicklungskarten (→) ausspielen. Danach ist sein linker Nachbar an der Reihe; dieser setzt das Spiel mit Aktion 1 fort.

DER SPIELABLAUF IM EINZELNEN

1. Rohstofferträge

Der Spieler am Zug wirft beide Würfel. Die addierte Gesamt-Augenzahl bestimmt die Landfelder, die Erträge erbringen!

- Jeder Spieler, der eine Siedlung an einem Feld mit der gewürfelten Zahl stehen hat, nimmt sich für die Siedlung 1 Rohstoffkarte dieses Feldes. Hat ein Spieler 2 oder 3 Siedlungen um ein ausgewürfeltes Landfeld stehen, erhält er für jede Siedlung 1 Rohstoffkarte. Hat ein Spieler eine Stadt an einem Feld mit der gewürfelten Zahl stehen, erhält er für die Stadt 2 Rohstoffkarten.

Beispiele: Wird eine „8“ gewürfelt, erhält Spieler Rot 2 x Erz für seine beiden Siedlungen. Spieler Weiß erhält 1 Erz. Beim Wurf einer „10“ erhält Spieler Weiß 1 Wolle. Wäre die weiße Siedlung eine Stadt, so würde Spieler Weiß beim Wurf einer „10“ 2 Wolle erhalten.



Spielanleitung *Siedler von Catan*, www.kosmos.de

Regeln – konkret

Würfel.



- Die Rohstoffkarten werden in 5 Stapel sortiert und offen in die Fächer der Kartenhalter gelegt. Die Kartenhalter werden neben dem Spielfeld bereitgestellt.



- Die Entwicklungskarten (→) werden gemischt und verdeckt in das freie Fach des Kartenhalters gelegt.



- Zuletzt erhält jeder Spieler für die mit einem Stern markierte Siedlung (siehe S. 1) die ersten Rohstoffstränge: Für jedes Landfeld, das an diese Siedlung angrenzt, nimmt er sich eine entsprechende Rohstoffkarte vom Stapel. *Beispiel: Blau erhält für seine Siedlung ganz oben je 1 Karte Holz, Erz und Wolle.*
- Seine Rohstoffkarten hält jeder Spieler verdeckt in der Hand.

Zusätzlich: Er kann zu einem beliebigen Zeitpunkt seines Zuges (auch vor dem Würfeln) eine seiner Entwicklungskarten (→) ausspielen. Danach ist sein linker Nachbar an der Reihe; dieser setzt das Spiel mit Aktion 1 fort.

DER SPIELABLAUF IM EINZELNEN

1. Rohstofferträge

Der Spieler am Zug wirft beide Würfel: Die addierte Gesamt-Augenzahl bestimmt die Landfelder, die Erträge erbringen!

- Jeder Spieler, der eine Siedlung an einem Feld mit der gewürfelten Zahl stehen hat, nimmt sich für die Siedlung 1 Rohstoffkarte dieses Feldes. Hat ein Spieler 2 oder 3 Siedlungen um ein ausgewürfeltes Landfeld stehen, erhält er für jede Siedlung 1 Rohstoffkarte. Hat ein Spieler eine Stadt an einem Feld mit der gewürfelten Zahl stehen, erhält er für die Stadt 2 Rohstoffkarten.

Beispiel: Wird eine „8“ gewürfelt, erhält Spieler Rot 2 x Erz für seine beiden Siedlungen. Spieler Weiß erhält 1 Erz. Beim Wurf einer „10“ erhält Spieler Weiß 1 Wolle. Wäre die weiße Siedlung eine Stadt, so würde Spieler Weiß beim Wurf einer „10“ 2 Wolle erhalten.



Spielanleitung *Siedler von Catan*, www.kosmos.de

Beispiel, das

- a) beliebig herausgegriffener, typischer Einzelfall (als Erklärung für eine bestimmte Erscheinung oder einen bestimmten Vorgang); Exempel
- b) Vorbild, [einmaliges] Muster

[nach: *Duden*]

Ziel von Beispielen

- Verständigung mit Kunden und anderen Stakeholdern
 - Aufdecken von Randfällen
 - Vermeiden von Missverständnissen
- Kommunikation zwischen Entwicklern
- Grundlage für Tests

Abstrakt vs. konkret

Abstrakt	Konkret
Student	Karli
Statusgruppe an der Universität	Student
Auto	Das blaue Auto an der Kreuzung mit dem Nummernschild ...
Farbe	Grün

Faustregel: Wenn es einen individuellen Namen hat, dann ist es konkret.

Abstrakt oder konkret?

- Es gibt eine Universität, in der man sich von Raum zu Raum bewegen kann.
- Karli wechselt den Raum.
- Um an der Abschlussprüfung teilzunehmen, muss jeder Student eine festgelegte Anzahl an Credit Points erreichen.
- Karli befindet sich im *calculus*-Raum, hat 37 Credit Points, noch 177 Punkte Motivation und wechselt in den Raum *modeling*.



Szenario, das

1. szenisch gegliederter Entwurf eines Films [als Entwicklungsstufe zwischen Exposé und Drehbuch] ...
2. ...
3. (in der öffentlichen und industriellen Planung) hypothetische Aufeinanderfolge von Ereignissen, die zur Beachtung kausaler Zusammenhänge konstruiert wird ...
4. Beschreibung, Entwurf, Modell der Abfolge von möglichen Ereignissen oder der hypothetischen Durchführung einer Sache

[nach: *Duden*]

Von Beispielen zu Szenarien/Stories

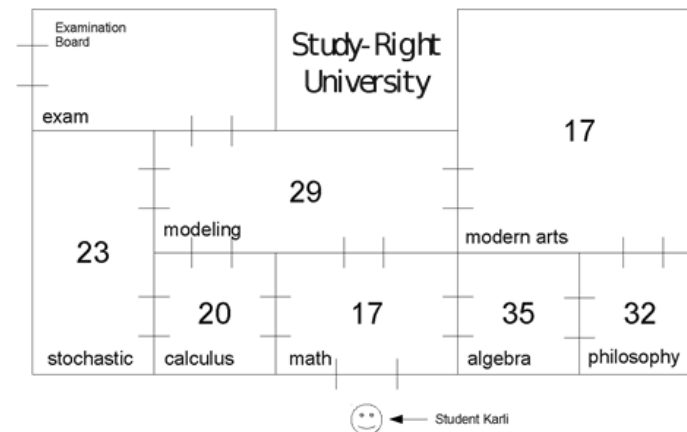
- In Vorlesung: *Szenario* und *Story* äquivalent verwendet
- Bedeutung:
 - Bezieht sich auf die *Ausführung einer bestimmten Funktionalität* des zu entwickelnden Systems
 - Besteht aus einer Reihe von Schritten, die ein konkretes Beispiel für das *Verhalten* des fertig entwickelten Systems in der Praxis und in der Interaktion mit potenziellen Benutzern beschreiben
- Besteht aus
 - Konkretem **Titel**
 - Abfolge von konkreten Schritten (**Aktionen**)
 - **Start** (Ausgangssituation) und **Ende** sind klar markiert
- **User story**: Die ursprünglichen Szenarien, die der Kunde liefert
 - Werden im Laufe der Entwicklung intern angepasst, um weiterhin zum Zustand des Systems zu passen und der Kommunikation zwischen Entwicklern zu dienen

Beispielszenario

Titel: Karli versucht zu graduieren und scheitert

Beschreibung:

- Ausgangssituation:** Karli möchte einen Abschluss an der Study-Right University erwerben. Zur Zeit braucht er dafür 214 Credit Points (CP). Karli startet mit 0 CP und einer Motivation von 214 Punkten. Er befindet sich außerhalb der Universität.

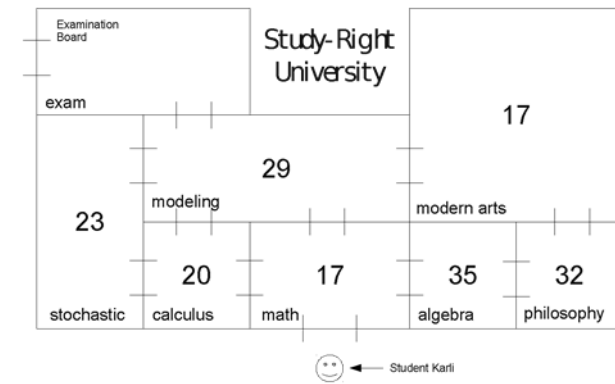


Beispielszenario (Fortsetzung)

Titel: Karli versucht zu graduieren und scheitert

Beschreibung:

2. Karli betritt den Raum *math*, nimmt an der Vorlesung teil, erhält automatisch 17 CP und verliert 17 Motivationspunkte. Er hat nun 17 CP und 197 Motivationspunkte.
3. Karli betritt den Raum *modeling*, nimmt an der Vorlesung teil, erhält automatisch 29 CP und verliert 29 Motivationspunkte. Er hat nun 46 CP, 168 Motivationspunkte und an den Vorlesungen *Mathematik* und *Modellierung* teilgenommen.
4. Karli betritt den Raum *examination*. Er wird aufgefordert nachzuweisen, welche Kurse er belegt hat und wie viele CP erworben. Da Karli über weniger als 214 CP verfügt, darf er nicht graduieren, muss die Universität verlassen und neu anfangen.



Zusammenfassung

- Modellierung ist eine zentrale Tätigkeit in der Informatik.
 - Wir benötigen Modelle der Wirklichkeit, um dem Computer Daten überhaupt zugänglich zu machen.
 - Wir benötigen Modelle, um den Überblick über komplexe Phänomene zu behalten.
- Grundvoraussetzung für gute Modelle ist ein gutes Verständnis des Phänomens, das modelliert werden soll.
- Konkrete Beispiele helfen, ein solches gutes Verständnis zu entwickeln.
 - Sie decken Randfälle auf und helfen zu überprüfen, ob die allgemeinen Regeln korrekt und vollständig sind.
 - Sie helfen sicherzustellen, dass alle Stakeholder die allgemeinen Regeln gleich interpretieren.
- Szenarien (user stories) stellen das gewünschte Verhalten eines zu entwickelnden Systems an konkreten Beispielen exemplarisch dar.