Responsibility-Driven Design

Authors of slides:
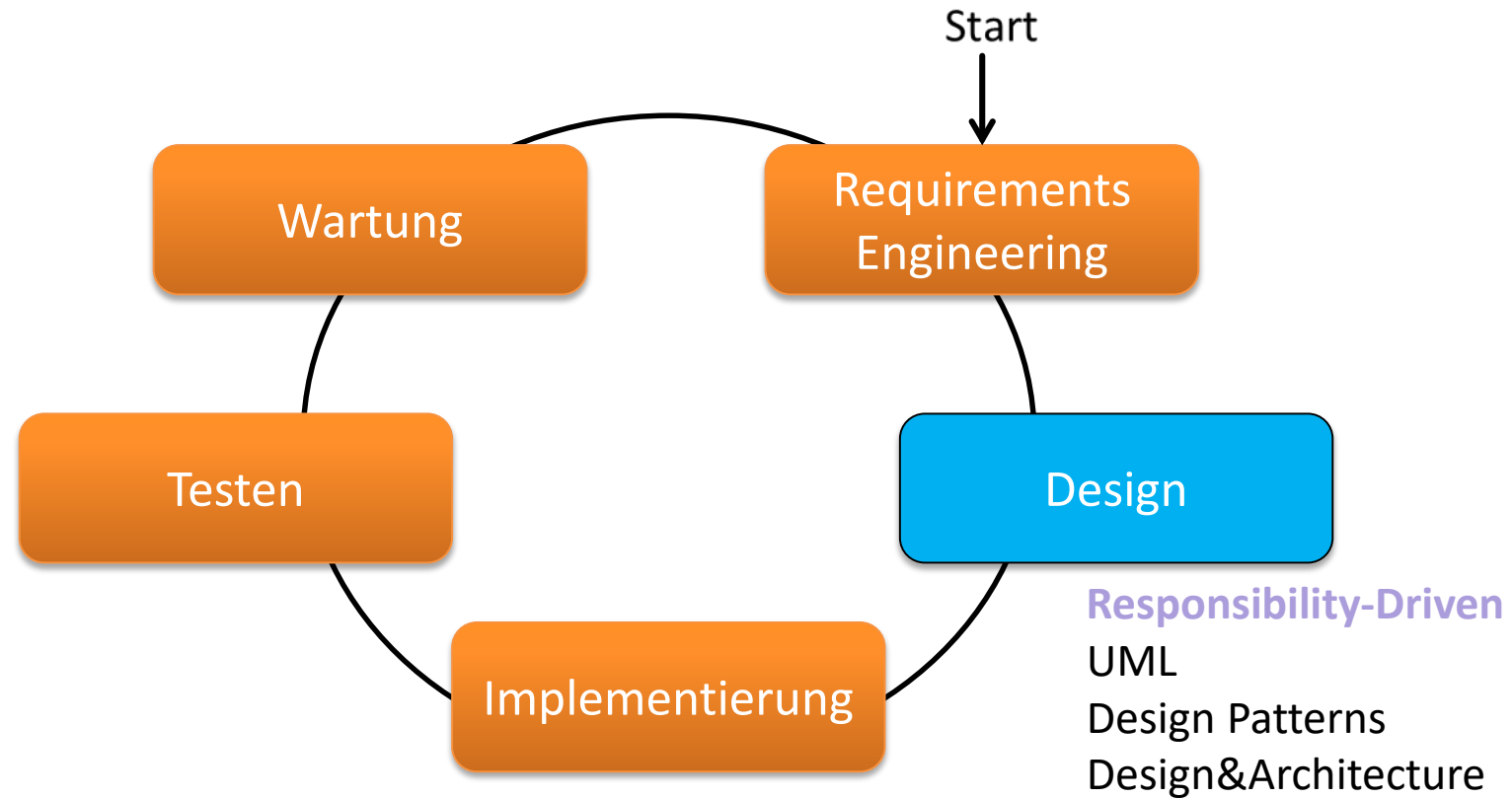**Norbert Siegmund**
Janet Siegmund
Oscar Nierstrasz
Sven Apel

Studie

- Es geht um Code-Verstehen
- Dabei werden Augenbewegungen aufgezeichnet
- 5 Euro für Teilnahme

Einordnung

Start

Requirements Engineering

Wartung

Design

Testen

Implementierung

Responsibility-Driven
UML
Design Patterns
Design&Architecture

Lernziele

- Notwendigkeit von Responsibility-Driven Design kennen lernen

- Prinzip von Responsibility-Driven Design kennen lernen und anwenden können

Warum
Responsibility-Driven Design?

Warum Responsibility-Driven Design?

Ansatz: Funktionale Dekomposition

> *Decompose according to the functions a system is supposed to perform.*

– Gut, bei stabilen Requirements und einer monolithischen Funktion

Aber:

– Naiv: Moderne Systems verfügen über mehr als eine Funktion

– Wartbarkeit: Systemfunktionen entwickeln sich $\Rightarrow$ Redesign betrifft gesamtes System

– Interoperabilität: Zusammenarbeit mit anderen Systemen ist kompliziert

Warum Responsibility-Driven Design?

Ansatz: Objekt-Orientierte Dekomposition

> *Decompose according to the objects a system is supposed to manipulate.*

– Besser bei komplexen, sich ändernden Systemen

Aber:

– **Wie findet man die Objekte?**

Iteration in Objekt-Orientierten Designs

- Ergebnis des Design-Prozesses ist **kein finales Produkt***:*
  - Design-Entscheidung werden evtl. **überdacht**, selbst nach deren Implementierung
  - Design ist nicht linear, sondern **iterativ**

- Der Design-Prozess ist **nicht algorithmisch**:
  - Eine Design-Methode bietet daher nur **Richtlinien** und keine festen Regeln
  - "a good **sense of style** often helps produce clean, elegant designs — designs that make a lot of sense from the engineering standpoint"

> Responsibility-Driven Design ist eine (Analyse- und) Design-Technik, die gut in Kombination mit verschiedenen Methoden und Notationen arbeitet.

Wie komme ich zu Klassen?
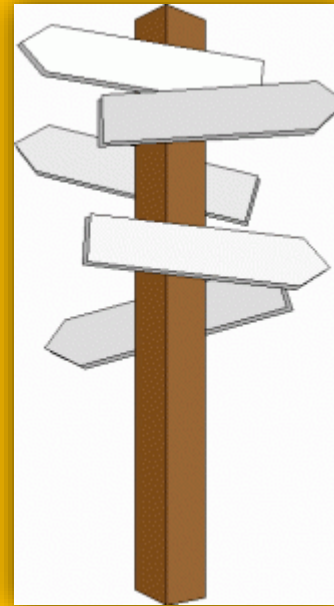
Initiale Exploration

1. Finde die **Klassen** in deinem System

2. Bestimme die **Verantwortlichkeiten** jeder Klasse

3. Bestimme wie Objekte **zusammenarbeiten**, um ihre Verantwortlichkeiten zu erfüllen

Initiale Exploration: Finden der Klassen

Start mit der Requirements Spezifikation*:*

Was sind die Ziele, der erwartete Input und Output des Systems, welches entworfen wird?

1. Suche nach **Nomen Phrasen**:
   – Separiere in offensichtliche Klassen, Kandidaten für Klassen und keine Klassen

Beispiel: Drawing Editor Requirements Specification

The drawing editor is an interactive graphics editor. With it, users can create and edit drawings composed of lines, rectangles, ellipses and text.

Tools control the mode of operation of the editor. Exactly one tool is active at any given time.

Two kinds of tools exist: the selection tool and creation tools. When the selection tool is active, existing drawing elements can be selected with the cursor. One or more drawing elements can be selected and manipulated; if several drawing elements are selected, they can be manipulated as if they were a single element. Elements that have been selected in this way are referred to as the current selection. The current selection is indicated visually by displaying the control points for the element. Clicking on and dragging a control point modifies the element with which the control point is associated.

When a creation tool is active, the current selection is empty. The cursor changes in different ways according to the specific creation tool, and the user can create an element of the selected kind. After the element is created, the selection tool is made active and the newly created element becomes the current selection.

The text creation tool changes the shape of the cursor to that of an I-beam. The position of the first character of text is determined by where the user clicks the mouse button.

The creation tool is no longer active when the user clicks the mouse button outside the text element. The control points for a text element are the four corners of the region within which the text is formatted. Dragging the control points changes this region. The other creation tools allow the creation of lines, rectangles and ellipses. They change the shape of the cursor to that of a crosshair. The appropriate element starts to be created when the mouse button is pressed, and is completed when the mouse button is released. These two events create the start point and the stop point.

The line creation tool creates a line from the start point to the stop point. These are the control points of a line. Dragging a control point changes the end point.

The rectangle creation tool creates a rectangle such that these points are diagonally opposite corners. These points and the other corners are the control points. Dragging a control point changes the associated corner.

The ellipse creation tool creates an ellipse fitting within the rectangle defined by the two points described above. The major radius is one half the width of the rectangle, and the minor radius is one half the height of the rectangle. The control points are at the corners of the bounding rectangle. Dragging control points changes the associated corner.

CRC-Karten

- C: Candidates
- R: Responsibility
- C: Collaboration

Klassenname	
Verantwortlichkeit 1	Klasse 1, Klasse 2
Verantwortlichkeit 2	Klasse 3

CRC-Karten
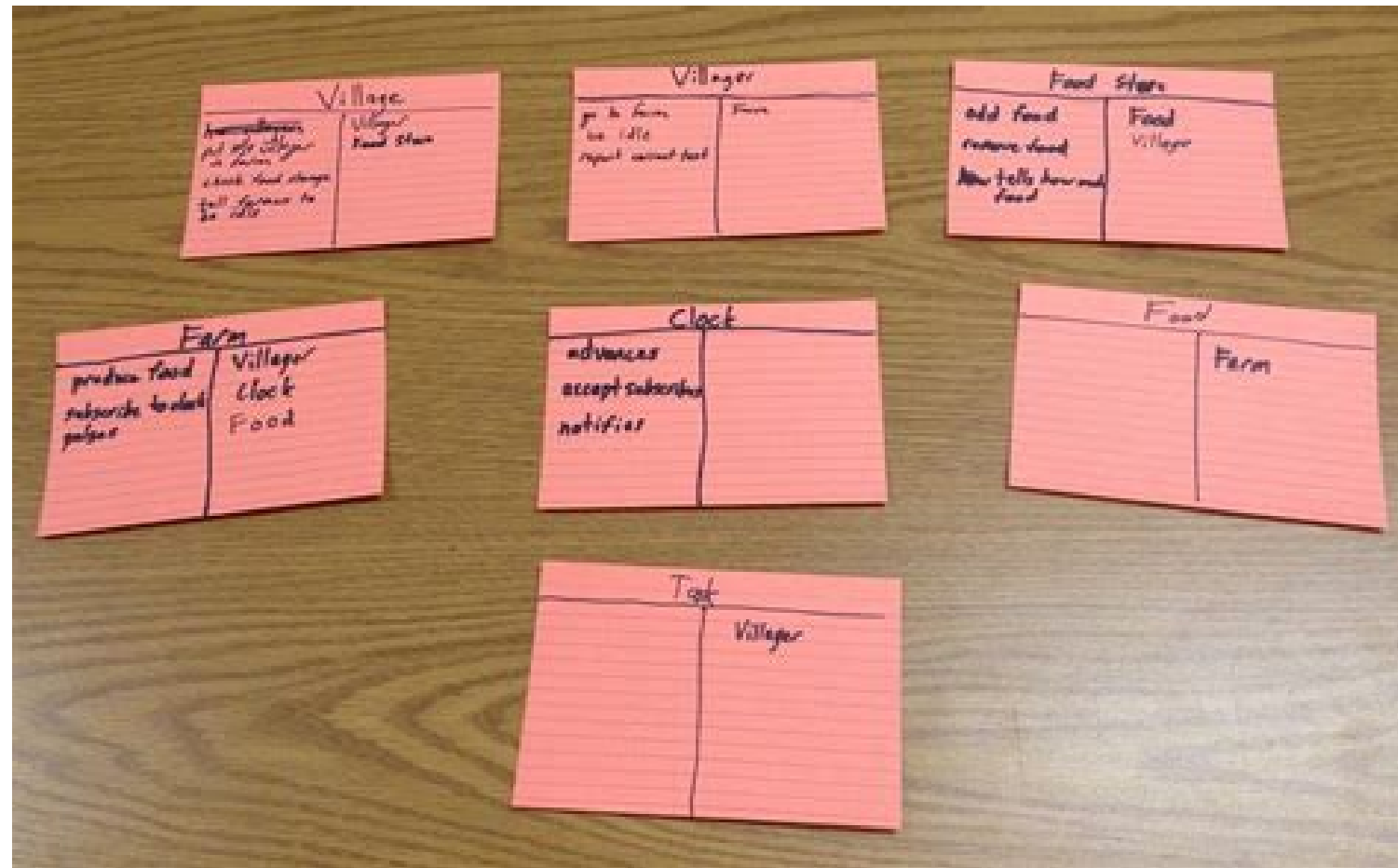
Wie wähle ich Klassen aus?

- **Physikalische Objekte**
- **Konzeptuelle Entitäten**
- **Ein Wort für ein Konzept**
- **Vorsicht bei Adjektiven**
- **Kategorien von Klassen**
- **Interface zum System**
- **Modelliere Attributwerte**

Initiale Exploration: Finden der Klassen

2. Verfeinere die Liste von **Kandidaten**

Mögliche Hinweise:
- Modelliere **physikalische Objekte** — z.B. Festplatte, Drucker
- Modelliere **konzeptuelle Entitäten** — z.B. Windows, Dateien
- Wähle **ein Wort für ein Konzept** — Was bedeutet das Konzept innerhalb des Systems?
- Vorsicht bei **Adjektiven** — Ist es wirkliche eine eigenständige Klasse?
- Modelliere **Kategorien von Klassen** — Verschiebe noch Modellierung von Vererbungen
- Modelliere **Interfaces** zum System — z.B., Nutzerinterface, Programminterface
- Modelliere Attribut**werte**, nicht Attribute — z.B., Punkt vs. Center

Physikalische Objekte

The drawing editor is an interactive graphics editor. With it, users can create and edit drawings composed of lines, rectangles, ellipses and text.

Tools control the mode of operation of the editor. Exactly one tool is active at any given time.

Two kinds of tools exist: the selection tool and creation tools. When the selection tool is active, existing drawing elements can be selected with the cursor. One or more drawing elements can be selected and manipulated; if several drawing elements are selected, they can be manipulated as if they were a single element. Elements that have been selected in this way are referred to as the current selection. The current selection is indicated visually by displaying the control points for the element. Clicking on and dragging a control point modifies the element with which the control point is associated.

When a creation tool is active, the current selection is empty. The cursor changes in different ways according to the specific creation tool, and the user can create an element of the selected kind. After the element is created, the selection tool is made active and the newly created element becomes the current selection.

The text creation tool changes the shape of the cursor to that of an I-beam. The position of the first character of text is determined by where the user clicks the mouse button. The creation tool is no longer active when the user clicks the mouse button outside the text element. The control points for a text element are the four corners of the region within which the text is formatted. Dragging the control points changes this region. The other creation tools allow the creation of lines, rectangles and ellipses. They change the shape of the cursor to that of a crosshair. The appropriate element starts to be created when the mouse button is pressed, and is completed when the mouse button is released. These two events create the start point and the stop point.

The line creation tool creates a line from the start point to the stop point. These are the control points of a line. Dragging a control point changes the end point.

The rectangle creation tool creates a rectangle such that these points are diagonally opposite corners. These points and the other corners are the control points. Dragging a control point changes the associated corner.

The ellipse creation tool creates an ellipse fitting within the rectangle defined by the two points described above. The major radius is one half the width of the rectangle, and the minor radius is one half the height of the rectangle. The control points are at the corners of the bounding rectangle. Dragging control points changes the associated corner.

Konzeptuelle Entitäten

The drawing editor is an interactive graphics editor. With it, users can create and edit drawings composed of lines, rectangles, ellipses and text.

Tools control the mode of operation of the editor. Exactly one tool is active at any given time.

Two kinds of tools exist: the selection tool and creation tools. When the selection tool is active, existing drawing elements can be selected with the cursor. One or more drawing elements can be selected and manipulated; if several drawing elements are selected, they can be manipulated as if they were a single element. Elements that have been selected in this way are referred to as the current selection. The current selection is indicated visually by displaying the control points for the element. Clicking on and dragging a control point modifies the element with which the control point is associated.

When a creation tool is active, the current selection is empty. The cursor changes in different ways according to the specific creation tool, and the user can create an element of the selected kind. After the element is created, the selection tool is made active and the newly created element becomes the current selection.

The text creation tool changes the shape of the cursor to that of an I-beam. The position of the first character of text is determined by where the user clicks the mouse button. The creation tool is no longer active when the user clicks the mouse button outside the text element. The control points for a text element are the four corners of the region within which the text is formatted. Dragging the control points changes this region. The other creation tools allow the creation of lines, rectangles and ellipses. They change the shape of the cursor to that of a crosshair. The appropriate element starts to be created when the mouse button is pressed, and is completed when the mouse button is released. These two events create the start point and the stop point.

The line creation tool creates a line from the start point to the stop point. These are the control points of a line. Dragging a control point changes the end point.

The rectangle creation tool creates a rectangle such that these points are diagonally opposite corners. These points and the other corners are the control points. Dragging a control point changes the associated corner.

The ellipse creation tool creates an ellipse fitting within the rectangle defined by the two points described above. The major radius is one half the width of the rectangle, and the minor radius is one half the height of the rectangle. The control points are at the corners of the bounding rectangle. Dragging control points changes the associated corner.

Ein Wort für ein Konzept

The drawing editor is an interactive graphics editor. With it, users can create and edit drawings composed of lines, rectangles, ellipses and text.

Tools control the mode of operation of the editor. Exactly one tool is active at any given time.

Two kinds of tools exist: the selection tool and creation tools. When the selection tool is active, existing drawing elements can be selected with the cursor. One or more drawing elements can be selected and manipulated; if several drawing elements are selected, they can be manipulated as if they were a single element. Elements that have been selected in this way are referred to as the current selection. The current selection is indicated visually by displaying the control points for the element. Clicking on and dragging a control point modifies the element with which the control point is associated.

When a creation tool is active, the current selection is empty. The cursor changes in different ways according to the specific creation tool, and the user can create an element of the selected kind. After the element is created, the selection tool is made active and the newly created element becomes the current selection.

The text creation tool changes the shape of the cursor to that of an I-beam. The position of the first character of text is determined by where the user clicks the mouse button. The creation tool is no longer active when the user clicks the mouse button outside the text element. The control points for a text element are the four corners of the region within which the text is formatted. Dragging the control points changes this region. The other creation tools allow the creation of lines, rectangles and ellipses. They change the shape of the cursor to that of a crosshair. The appropriate element starts to be created when the mouse button is pressed, and is completed when the mouse button is released. These two events create the start point and the stop point.

The line creation tool creates a line from the start point to the stop point. These are the control points of a line. Dragging a control point changes the end point.

The rectangle creation tool creates a rectangle such that these points are diagonally opposite corners. These points and the other corners are the control points. Dragging a control point changes the associated corner.

The ellipse creation tool creates an ellipse fitting within the rectangle defined by the two points described above. The major radius is one half the width of the rectangle, and the minor radius is one half the height of the rectangle. The control points are at the corners of the bounding rectangle. Dragging control points changes the associated corner.

Adjektive und zusammengesetzte Nomen

The drawing editor is an interactive graphics editor. With it, users can create and edit drawings composed of lines, rectangles, ellipses and text.

Tools control the mode of operation of the editor. Exactly one tool is active at any given time.

Two kinds of tools exist: the selection tool and creation tools. When the selection tool is active, existing drawing elements can be selected with the cursor. One or more drawing elements can be selected and manipulated; if several drawing elements are selected, they can be manipulated as if they were a single element. Elements that have been selected in this way are referred to as the current selection. The current selection is indicated visually by displaying the control points for the element. Clicking on and dragging a control point modifies the element with which the control point is associated.

When a creation tool is active, the current selection is empty. The cursor changes in different ways according to the specific creation tool, and the user can create an element of the selected kind. After the element is created, the selection tool is made active and the newly created element becomes the current selection.

The text creation tool changes the shape of the cursor to that of an I-beam. The position of the first character of text is determined by where the user clicks the mouse button. The creation tool is no longer active when the user clicks the mouse button outside the text element. The control points for a text element are the four corners of the region within which the text is formatted. Dragging the control points changes this region. The other creation tools allow the creation of lines, rectangles and ellipses. They change the shape of the cursor to that of a crosshair. The appropriate element starts to be created when the mouse button is pressed, and is completed when the mouse button is released. These two events create the start point and the stop point.

The line creation tool creates a line from the start point to the stop point. These are the control points of a line. Dragging a control point changes the end point.

The rectangle creation tool creates a rectangle such that these points are diagonally opposite corners. These points and the other corners are the control points. Dragging a control point changes the associated corner.

The ellipse creation tool creates an ellipse fitting within the rectangle defined by the two points described above. The major radius is one half the width of the rectangle, and the minor radius is one half the height of the rectangle. The control points are at the corners of the bounding rectangle. Dragging control points changes the associated corner.

Kategorien von Klassen; Interfaces

The drawing editor is an interactive graphics editor. With it, users can create and edit drawings composed of lines, rectangles, ellipses and text.

Tools control the mode of operation of the editor. Exactly one tool is active at any given time.

Two kinds of tools exist: the selection tool and creation tools. When the selection tool is active, existing drawing elements can be selected with the cursor. One or more drawing elements can be selected and manipulated; if several drawing elements are selected, they can be manipulated as if they were a single element. Elements that have been selected in this way are referred to as the current selection. The current selection is indicated visually by displaying the control points for the element. Clicking on and dragging a control point modifies the element with which the control point is associated.

When a creation tool is active, the current selection is empty. The cursor changes in different ways according to the specific creation tool, and the user can create an element of the selected kind. After the element is created, the selection tool is made active and the newly created element becomes the current selection.

The text creation tool changes the shape of the cursor to that of an I-beam. The position of the first character of text is determined by where the user clicks the mouse button. The creation tool is no longer active when the user clicks the mouse button outside the text element. The control points for a text element are the four corners of the region within which the text is formatted. Dragging the control points changes this region. The other creation tools allow the creation of lines, rectangles and ellipses. They change the shape of the cursor to that of a crosshair. The appropriate element starts to be created when the mouse button is pressed, and is completed when the mouse button is released. These two events create the start point and the stop point.

The line creation tool creates a line from the start point to the stop point. These are the control points of a line. Dragging a control point changes the end point.

The rectangle creation tool creates a rectangle such that these points are diagonally opposite corners. These points and the other corners are the control points. Dragging a control point changes the associated corner.

The ellipse creation tool creates an ellipse fitting within the rectangle defined by the two points described above. The major radius is one half the width of the rectangle, and the minor radius is one half the height of the rectangle. The control points are at the corners of the bounding rectangle. Dragging control points changes the associated corner.

Attributwerte

The drawing editor is an interactive graphics editor. With it, users can create and edit drawings composed of lines, rectangles, ellipses and text.

Tools control the mode of operation of the editor. Exactly one tool is active at any given time.

Two kinds of tools exist: the selection tool and creation tools. When the selection tool is active, existing drawing elements can be selected with the cursor. One or more drawing elements can be selected and manipulated; if several drawing elements are selected, they can be manipulated as if they were a single element. Elements that have been selected in this way are referred to as the current selection. The current selection is indicated visually by displaying the control points for the element. Clicking on and dragging a control point modifies the element with which the control point is associated.

When a creation tool is active, the current selection is empty. The cursor changes in different ways according to the specific creation tool, and the user can create an element of the selected kind. After the element is created, the selection tool is made active and the newly created element becomes the current selection.

The text creation tool changes the shape of the cursor to that of an I-beam. The position of the first character of text is determined by where the user clicks the mouse button. The creation tool is no longer active when the user clicks the mouse button outside the text element. The control points for a text element are the four corners of the region within which the text is formatted. Dragging the control points changes this region. The other creation tools allow the creation of lines, rectangles and ellipses. They change the shape of the cursor to that of a crosshair. The appropriate element starts to be created when the mouse button is pressed, and is completed when the mouse button is released. These two events create the start point and the stop point.

The line creation tool creates a line from the start point to the stop point. These are the control points of a line. Dragging a control point changes the end point.

The rectangle creation tool creates a rectangle such that these points are diagonally opposite corners. These points and the other corners are the control points. Dragging a control point changes the associated corner.

The ellipse creation tool creates an ellipse fitting within the rectangle defined by the two points described above. The major radius is one half the width of the rectangle, and the minor radius is one half the height of the rectangle. The control points are at the corners of the bounding rectangle. Dragging control points changes the associated corner.

Kandidaten für Klassen

Initiale Analyse ergibt die folgenden Kandidaten:

Character	Line Element
Control Point	Point
Creation Tool	Rectangle
Current Selection	Rectangle Creation Tool
Drawing	Rectangle Element
Drawing Editor	Selection Tool
Drawing Element	Text Creation Tool
Ellipse Creation Tool	Text Element
Ellipse Element	Tool
Line Creation Tool	

Erwartet, dass die Liste sich weiterentwickelt während Ihr beim Design voranschreitet

Wie identifiziere ich die Verantwortlichkeiten?

Initiale Exploration: Verantwortlichkeiten (Responsibilities)

Was sind Verantwortlichkeiten?

– Das Wissen, das ein Objekt verwaltet und anbietet

– Die Aktionen, die es ausführen kann

Verantwortlichkeiten repräsentieren die **öffentlichen Leistungen,** die ein Objekt seinen Klienten anbietet (aber nicht die Art, wie diese Leistungen realisiert werden können)

– Spezifiziere, **was** ein Objekt tut, nicht **wie** es dies tut

– Beschreibe noch nicht das Interface, sondern nur die **konzeptuellen Verantwortlichkeiten**

Initiale Exploration: Identifizieren der Verantwortlichkeiten

- Studiere die Requirements-Spezifikationen:
  - Hebe **Verben** hervor und bestimme, welche Verantwortlichkeiten diese repräsentieren
  - Mache einen **walk-through** vom System
    - Exploriere so viele Szenarien wie möglich
    - Identifiziere Aktionen, welche durch Eingaben an das System resultieren

- Studiere die Kandidatenklassen:
  - Klassennamen $\Rightarrow$ Rollen $\Rightarrow$ Verantwortlichkeiten
  - Aufgenommene „Sinnhaftigkeiten" von Klassen $\Rightarrow$ Verantwortlichkeiten

Drawing Editor: Verben

The drawing editor is an interactive graphics editor. With it, users can create and edit drawings composed of lines, rectangles, ellipses and text.

Tools control the mode of operation of the editor. Exactly one tool is active at any given time.

Two kinds of tools exist: the selection tool and creation tools. When the selection tool is active, existing drawing elements can be selected with the cursor. One or more drawing elements can be selected and manipulated; if several drawing elements are selected, they can be manipulated as if they were a single element. Elements that have been selected in this way are referred to as the current selection. The current selection is indicated visually by displaying the control points for the element. Clicking on and dragging a control point modifies the element with which the control point is associated.

When a creation tool is active, the current selection is empty. The cursor changes in different ways according to the specific creation tool, and the user can create an element of the selected kind. After the element is created, the selection tool is made active and the newly created element becomes the current selection.

…

Wie weise ich Verantwortlichkeiten zu?

Pelrine's Laws:

1. "Don't do anything you can push off to someone else."

2. "Don't let anyone else play with you."

- Beispiel: Klasse Buch und Klasse Bibliothek
  – Szenario: Such nach Text in einem Buch
  – Law 1: Bibliothek sollte nicht nach Text in einem bestimmten Buch suchen, obwohl die Klasse eine Liste von Büchern hat.
  – Law 2: Der Text eines Buches gehört zum Buch, also laß niemanden mit deinem Text arbeiten, sondern nur die Klasse Buch bestimmt alle erlaubten Aktionen dafür

Zuweisen von Verantwortlichkeiten: Zentral

- Behalte Informationen über eine Sache an **einem Ort**

- Ein Objekt wird zum Kontrollzentrum

  - Falls mehrere Objekte Zugriff auf die gleiche Information benötigen:

    1. Ein neues Objekte kann eingeführt werden, welches die Information verwaltet, oder
    2. Eines der vorhandenen Objekte kann die Information verwalten, oder
    3. die mehrfachen Objekte können in ein einzelnes Objekt überführt werden

- Sinnvoll bei kleinen Systemen

Zuweisen von Verantwortlichkeiten: Delegiert/Gebündelt

- Breche komplexe Verantwortlichkeiten auf
  - **Teile** Verantwortlichkeiten zwischen ähnlichen Objekten

- **Verteile gleichmäßig** die System-Intelligenz
  - Verhindere prozedural zentralisierte Verantwortlichkeiten
  - Halte Verantwortlichkeiten nah an den Objekten und nicht an deren Nutzern

- Definiere Verantwortlichkeiten so **generell** wie möglich
  - "draw yourself" vs. "draw a line/rectangle etc."
  - Führt zum Teilen

- Halte **Verhalten** zusammen mit jedweder **relevanten Information**
  - Prinzip der Kapselung

Beziehungen zwischen Klassen I

Weitere Verantwortlichkeiten können entdeckt werden, indem wir die Beziehungen zwischen Klassen untersuchen:

- Die "Is-Kind-Of" Beziehung:
  - Klassen, die **gemeinsame Attribute** teilen, teilen oft eine **gemeinsame Oberklasse**
  - Gemeinsame Oberklassen weisen auf **gemeinsame Verantwortlichkeiten** hin
  - z.B., um ein neues Drawing Element zu kreieren, muss das Creation Tool Folgendes tun:
  1. accept user input   **implemented in subclass**
  2. determine location to place it   **dynamisch**
  3. instantiate the element   **implemented in subclass**

Beziehungen zwischen Klassen II

- Die "Is-Analogous-To" Beziehung:
  - **Ähnlichkeiten** zwischen Klassen weisen auf eine zur Zeit noch unentdeckte Oberklasse hin

- Die "Is-Part-Of" Beziehung :
  - **Unterscheide** Verantwortlichkeiten zwischen eines **Teils** und des **Ganzen**

Schwierigkeiten bei der Zuweisung:
  - **Fehlende Klassen** im Design (z.B., Group Element), oder
  - **Freie Auswahl** zwischen mehreren Klassen hin

Wie finde ich Kollaborationen?

Kollaborationen

Was sind Kollaborationen?

- Kollaborationen sind Benutzeranfragen (client requests) an Dienste, die benötigt werden, um Verantwortlichkeiten zu erfüllen

- Kollaborationen enthüllen Kontroll- und Informationsflüsse und, ultimativ, Subsysteme

- Kollaborationen können fehlenden Verantwortlichkeiten offenbaren

- Analysen von Kommunikationsmustern können fehlerhaft zugewiesene Verantwortlichkeiten offenbaren

Finden von Kollaborationen

Für jede Verantwortlichkeit:

1. Kann die Klasse die Verantwortlichkeit selbstständig erfüllen?
2. Falls nicht, was benötigt sie, und von welcher anderen Klasse kann sie dies erhalten?

Für jede Klasse:

1. Was weiß diese Klasse?
2. Welche anderen Klassen benötigen ihre Informationen oder Ergebnisse? Prüfe auf Kollaborationen.
3. Klassen, die nicht mit anderen interagieren, sollten aussortiert werden. (Sorgfältig prüfen!)

Auflistung der Kollaborationen

Drawing (das Bild, das entsteht)	
Responsibility	Kollaboration
Kennt die Elemente, aus denen es besteht	DrawingElement
Kennt Reihenfolge seiner Elemente	
Kennt Position seiner Elemente	

Wie finde ich
Vererbungshierarchien?
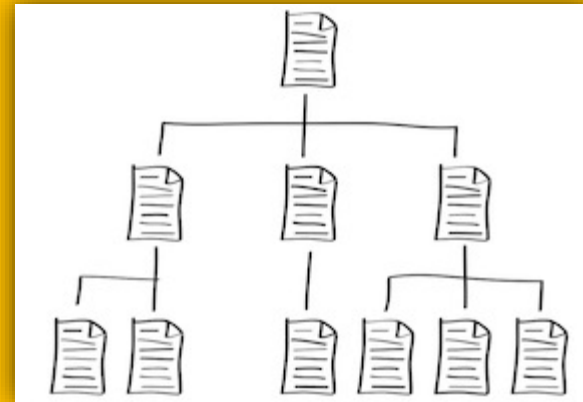
Finden von Abstrakten Klassen

Abstract classes factor out common behavior shared by other classes

- Gruppiere verwandte Klassen mit gemeinsamen Attributen
- Führe abstrakte Oberklassen ein, die diese Gruppe repräsentieren
- "Kategorien" sind gute Kandidaten für abstrakte Klassen

> **Vorsicht vor zu voreiliger Klassifizierung!**
> **Die Hierarchie wird evolvieren!**

Teilen von Verantwortlichkeiten

Konkreten Klassen kann man instanziieren und von ihnen erben.
Von abstrakten Klassen kann man nur erben.

Notiere Abstraktheit in Klassendiagrammen.
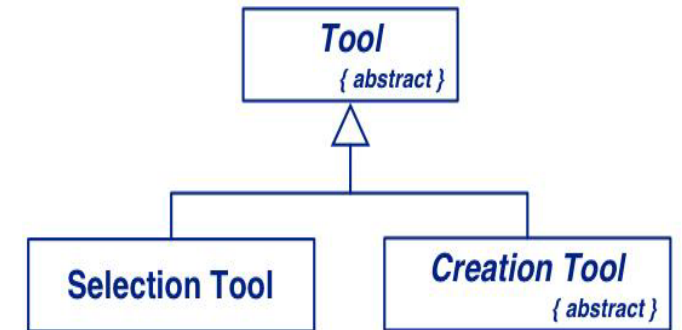
Venn Diagramme können für die Visualisierung von geteilten Verantwortlichkeiten verwendet werden.
(Achtung: nicht Teil von UML!)
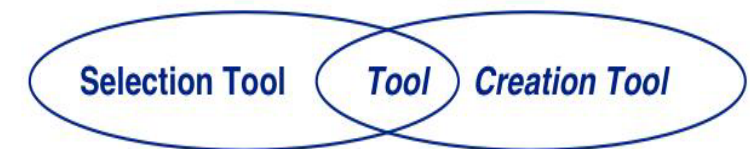
Mehrfachvererbung

Bestimme, ob eine Klasse **instanziiert** wird, um zu entscheiden ob sie **abstrakt oder konkret** ist.

Verantwortlichkeiten von Subklassen sind **größer** als von **Oberklassen**.

Überschneidungen repräsentieren **gemeinsame Oberklassen**.
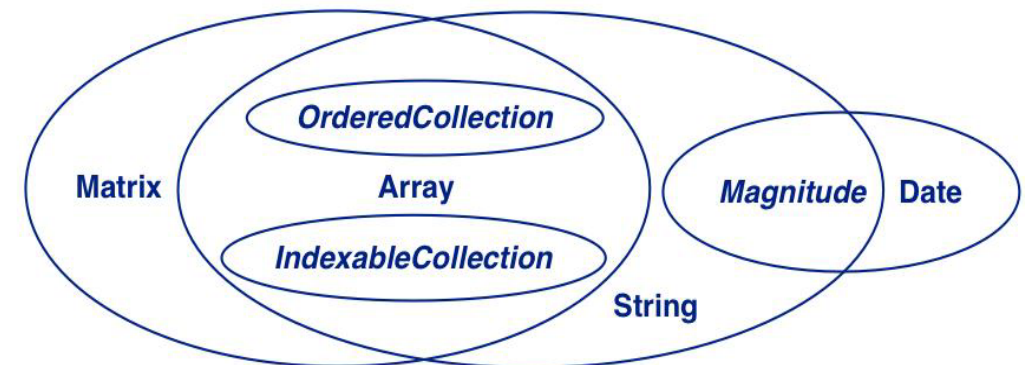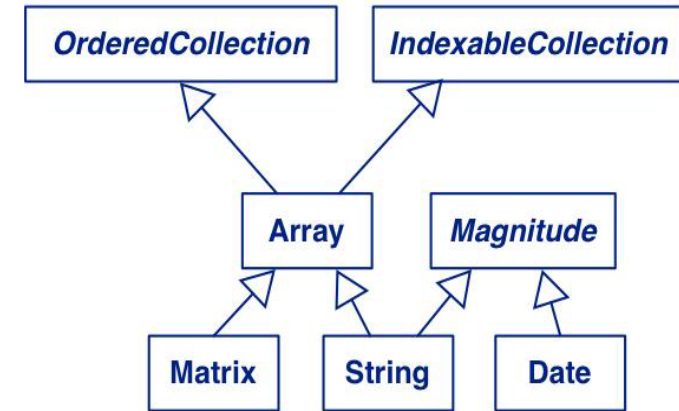
Entwerfen von Guten Hierarchien I

Modelliere eine "kind-of" Hierarchie:

- Unterklassen sollten alle geerbten Verantwortlichkeiten unterstützen, und eher noch mehr

Schiebe gemeinsame Verantwortlichkeiten so hoch wie möglich:

- Klassen, die gemeinsame Verantwortlichkeiten teilen sollten von einer gemeinsamen abstrakten Superklasse erben; Führe fehlende Superklassen ein

Entwerfen von Guten Hierarchien II

Stelle sicher, dass abstrakte Klassen nicht von konkreten Klassen erben:

- Eliminiere dies durch die Einführung weiterer **gemeinsamer abstrakter Superklassen**: abstrakte Klassen sollten Verantwortlichkeiten in einem implemtierungsunabhängigen Weg unterstützen

Eliminiere Klassen, die keine neue Funktionalität hinzufügen:

- Klassen sollten entweder neue Verantwortlichkeiten oder eine bestimmte Implementierung von vererbten Verantwortlichkeiten hinzufügen

Entwerfen von Kind-Of Hierarchien I

Korrekt gebildete Verantwortlichkeiten von Unterklassen:

C übernimmt *alle* Verantwortlichkeiten von A und B
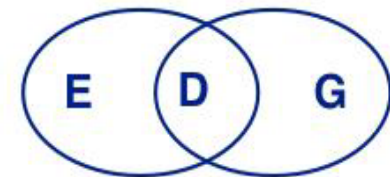
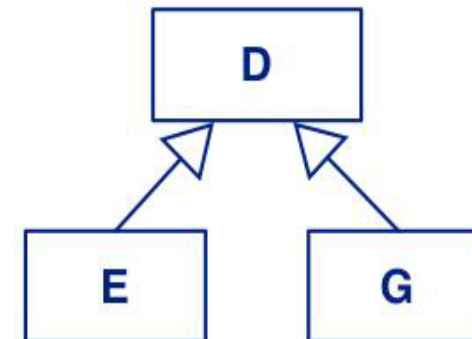# Entwerfen von Kind-Of Hierarchien II

Falsche Unter-Oberklassen-Beziehung

- G übernimmt nur **einige** der Verantwortlichkeiten, welche von E geerbt wurden
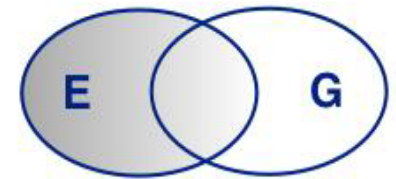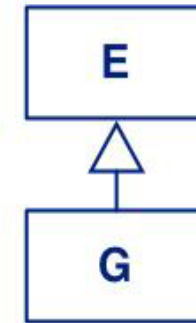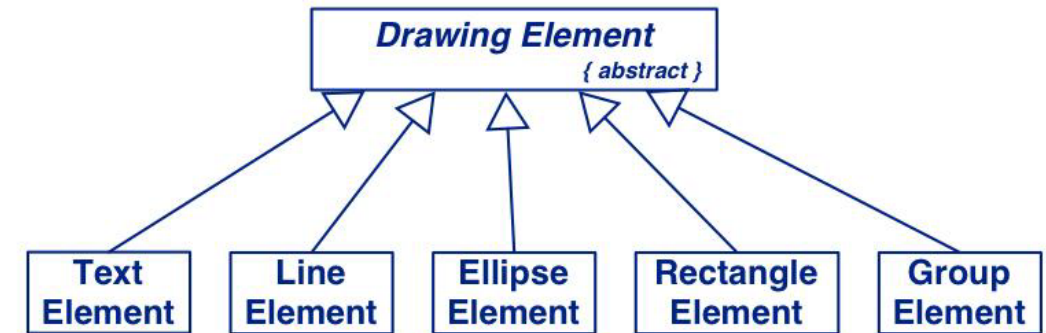
Verfeinerte Vererbungsbeziehung

- Führe eine **abstrakte Oberklasse** eine, welche die gemeinsamen Verantwortlichkeiten kapselt

Refaktorisierung von Verantwortlichkeiten

Lines, Ellipses und Rectangles sind verantwortlich, um die Breite und Farbe ihrer Linien zu speichern.

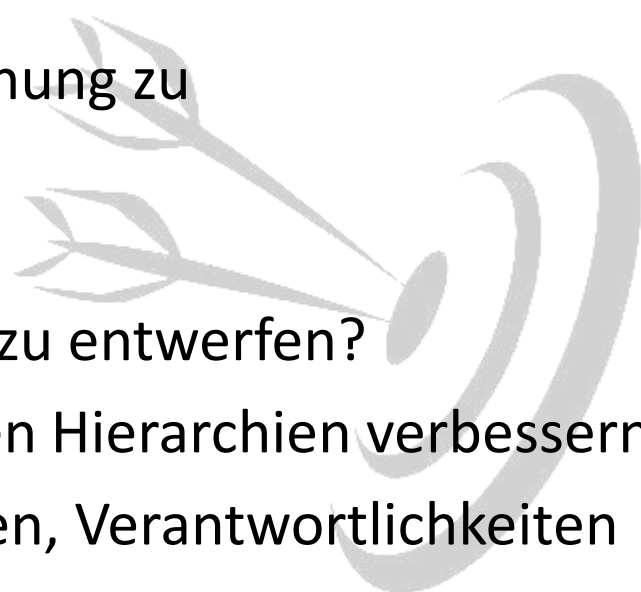Dies weißt auf eine **gemeinsame Superklasse hin.**

Protokolle (Interfaces)

Ein Protokoll ist eine **Menge von Signaturen** (d.h., ein **Interface**), die zu einer Klasse gehören.

– Generell, Protokolle sind für **öffentliche Verantwortlichkeiten** spezifiziert.

– Protokolle für **private** Verantwortlichkeiten sollten spezifiziert werden, falls sie benutzt oder implementiert werden in ihren **Sub(unter)klassen**

1. Entwerfe eine Protokoll für jede Klasse
2. Schreibe eine Design-Spezifikation für jede Klasse und Subsystem
3. Schreibe eine Design-Spezifikation für jeden Kontrakt/Interface

Was Sie mitgenommen haben sollten

- Welche Kriterien gibt es, mit denen ich potentielle Klassen identifizieren kann?

- Was sind Verantwortlichkeiten von Klassen und wie kann ich sie identifizieren?

- Wie kann das Identifizieren von Verantwortlichkeiten beim Identifizieren von Klassen helfen?

- Was sind Kollaborationen und wie stehen sie in Beziehung zu Verantwortlichkeiten?

- Wie kann ich abstrakte Klassen identifizieren?

- Welche Kriterien gibt es, um gute Klassenhierarchien zu entwerfen?

- Wie kann das Refaktorisieren von Verantwortlichkeiten Hierarchien verbessern?

- Leiten Sie aus einer Anforderungsbeschreibung Klassen, Verantwortlichkeiten und Kollaborationen ab

Literatur

- *Designing Object-Oriented Software*, R. Wirfs-Brock, B. Wilkerson, L. Wiener, Prentice Hall, 1990.