

Universität Karlsruhe (TH)

Forschungsuniversität · gegründet 1825

# Softwaretechnik 1

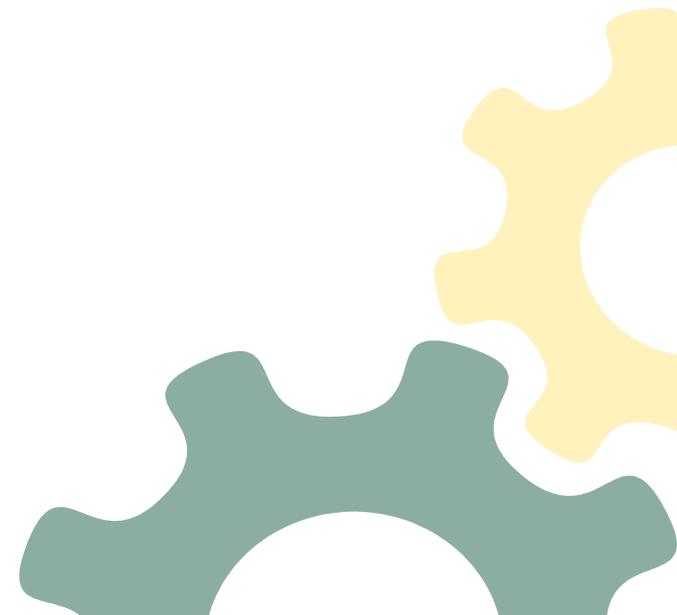
## Übung 1

07.05.2009



Fakultät für **Informatik**

Lehrstuhl für Programmiersysteme





# Aufgabe 1

Gegeben sei folgendes Szenario:

*Ihre Firma möchte die Kosten der Personalabteilung reduzieren. Dazu soll eine Internetseite zur Annahme von Onlinebewerbungen entwickelt werden. Interessenten für eine ausgeschriebene Stelle sollen die Möglichkeiten haben, ihre Bewerbungsunterlagen über diese Internetseite einzureichen. Die Personalabteilung Ihrer Firma soll zudem die eingereichten Bewerbungen über diese Internetseite verwalten können.*



# Aufgabe 1

Erstellen Sie ein Lastenheft für die Internetseite zur Annahme und Verwaltung von Online-Bewerbungen in dem aus der Vorlesung bekannten Format.

Das obige Szenario dient nur zur Illustration der Aufgabe und ist nicht vollständig! Ihr Lastenheft muss detaillierter ausgearbeitet sein. Geben Sie mindestens 5 funktionale Anforderungen, 3 Produktdaten und 3 nichtfunktionale Anforderungen an.



# Gliederungsschema Lastenheft

1. Zielbestimmung
2. Produkteinsatz
3. Funktionale Anforderungen
4. Produktdaten
5. Nichtfunktionale Anforderungen
6. Systemmodelle
  - a) Szenarien
  - b) Anwendungsfälle
7. Glossar



# 1. Zielbestimmung

Unsere Firma XYZ soll durch die Software in die Lage versetzt werden, freie Stellen im Internet auszuschreiben und Online-Bewerbungen zu den Stellen entgegenzunehmen und zu verwalten. Initiativbewerbungen sollen ebenfalls online entgegengenommen werden können.



## 2. Produkteinsatz

Die Software soll auf dem Webserver unserer Firma laufen und die Abwicklung von Online-Stellenausschreibungen unterstützen. Zielgruppe der Software sind die Bewerber, die Mitarbeiter der Personalabteilung und der zuständigen Fachabteilung.



## 3. Funktionale Anforderungen

- /FA 10/ Ersterfassung, Änderung und Löschung von Stellenangeboten
- /FA 20/ Entgegennahme, Bearbeitung und Löschung von Online-Bewerbungen
- /FA 30/ Versenden von Statusmitteilungen per E-Mail und/oder Post an den Bewerber bei:
  - a) Eingang
  - b) Einladung erstes (zweites) Vorstellungsgespräch
  - c) Zu- oder Absage
- /FA 40/ Weiterleitung der Bewerbung an den zuständigen Ansprechpartner in der Fachabteilung
- /FA 50/ Automatische Wiedervorlage von Initiativbewerbungen bei passenden neuen Stellenangeboten



## 4. Produktdaten

/PD 10/ Für das Stellenangebot sind seine Gültigkeit, Beschreibung und Titel sowie der Ansprechpartner /PD 30/ in der zuständigen Fachabteilung zu speichern.

/PD 20/ Für eine Online-Bewerbung sind die persönlichen Daten des Bewerbers /PD 40/, ein Anschreiben, Lebenslauf, Foto und Zeugnisse/Empfehlungsschreiben zu speichern. Bezieht sich die Bewerbung auf ein Stellenangebot, ist dieser Bezug zu speichern.

/PD 30/ Zu einem Ansprechpartner werden Nachname, Vorname, Telefon, E-Mail-Adresse und Anschrift sowie der Name der Fachabteilung gespeichert.

/PD 40/ ...



## 5. Nichtfunktionale Anforderungen

- /NF 10/ Die Software darf höchstens 1 Tag im Jahr nicht verfügbar sein.
- /NF 20/ Die Software soll 1000 Bewerbungen pro Stunde entgegennehmen können.
- /NF 30/ Zwischen Absenden der Bewerbung und Versand der Bestätigungs-E-Mail darf nicht mehr als 1 Minute vergehen.



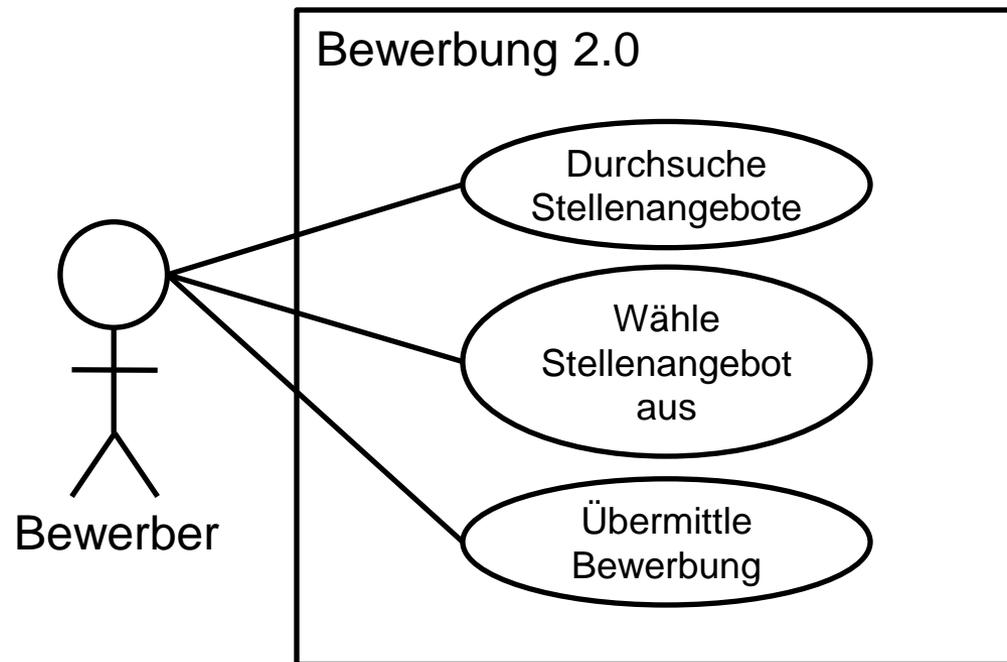
## 6. Systemmodelle – Szenarien

### Einreichen einer Bewerbung

Der Bewerber geht auf die Webseite und begutachtet die Stellenangebote. Er wählt ein passendes Stellenangebot aus und gelangt zu einem Formular, in welches er seine persönlichen Daten eingibt. Dort lädt er zusätzlich Anschreiben, Lebenslauf, Zeugnisse und Foto hoch. Er bestätigt alle Daten und erhält abschließend die Empfangsbestätigung per E-Mail.



# 6. Systemmodelle – Anwendungsfälle





## 7. Glossar

**Arbeitszeugnis** In einem Arbeitszeugnis beurteilt der Arbeitgeber die Arbeitsleistung einschließlich der Qualifikation und das dienstliche Verhalten des Arbeitnehmers, wenn dieser das Unternehmen verlässt (Endzeugnis). [Quelle: Wikipedia]

**Lebenslauf** Listet in tabellarischer Form die wichtigsten für die Stelle relevanten Daten einer Person auf.

...



## Aufgabe 2 – Bepunktung

- Sie sollten alle Akteure und Anwendungsfälle erkannt haben.
- Mehrfach verwendete Funktionalität sollten Sie erkannt und herausfaktoriert haben.
- Das Diagramm muss vollständig und plausibel sein.
- Die Notation sollte der in der Vorlesung vorgestellten Notation entsprechen.



## Aufgabe 2

Gegeben sei folgendes Szenario:

*In der Karlsruher Innenstadt brennt es in einem großen Kaufhaus. Die im Kaufhaus installierten Feuermelder registrieren das Feuer und melden den Notfall automatisch an die Leitstelle der Feuerwehr Karlsruhe. Der Disponent auf der Leitstelle wird durch ein Signal auf den Notfall hingewiesen und alarmiert die notwendigen Einsatzkräfte. Da es sich um einen Gebäudebrand handelt, werden neben der Feuerwehr auch automatisch die Polizei und zwei Rettungswagen alarmiert. Nach Ankunft der Feuerwehr stellt der Einsatzleiter fest, dass weitere Einsatzkräfte zur Brandbekämpfung benötigt werden. Er funkt die Leitstelle an und fordert weitere Einsatzkräfte nach. Der Disponent bestätigt die Anforderung und alarmiert einen weiteren Löschzug der Feuerwehr.*



## Aufgabe 2

Gegeben sei folgendes Szenario (Fortsetzung):

*Eine alte Frau, die währenddessen am Kaufhaus vorbei läuft, verliert vor lauter Aufregung das Bewusstsein und verletzt sich am Kopf. Ein aufmerksamer Passant bemerkt dies und meldet der Leitstelle den Unfall. Der Disponent wird auch auf diesen Notfall hingewiesen und sendet einen Notarzt zur Einsatzstelle. Zusammen mit dem Notarzt wird ein weiterer Rettungswagen zur Einsatzstelle geschickt.*



## Aufgabe 2

- a) Erstellen Sie ein UML-Anwendungsfall-diagramm zu diesem Szenario. Identifizieren Sie alle teilnehmenden Akteure sowie die vorkommenden Anwendungsfälle. Faktorisieren Sie mehrfach verwendete Funktionalität aus den Anwendungsfällen heraus und verwenden Sie **<<include>>**- und **<<extend>>**-Beziehungen dort, wo es notwendig ist.



## Aufgabe 2a

Gegeben sei folgendes Szenario:

In der Karlsruher Innenstadt brennt es in einem großen Kaufhaus. Die im Kaufhaus installierten **Feuermelder** registrieren das **Feuer und melden** den Notfall automatisch an die Leitstelle der Feuerwehr Karlsruhe. Der **Disponent** auf der Leitstelle wird durch ein Signal auf den Notfall hingewiesen und **alarmiert die notwendigen Einsatzkräfte**. Da es sich um einen Gebäudebrand handelt, werden neben der Feuerwehr auch automatisch die Polizei und zwei Rettungswagen alarmiert. Nach Ankunft der Feuerwehr stellt der **Einsatzleiter** fest, dass weitere Einsatzkräfte zur Brandbekämpfung benötigt werden. Er funkt die Leitstelle an und **fordert weitere Einsatzkräfte nach**. Der Disponent bestätigt die Anforderung und **alarmiert einen weiteren Löschzug** der Feuerwehr.

Akteur

Anwendungsfall



## Aufgabe 2a

Gegeben sei folgendes Szenario (Fortsetzung):

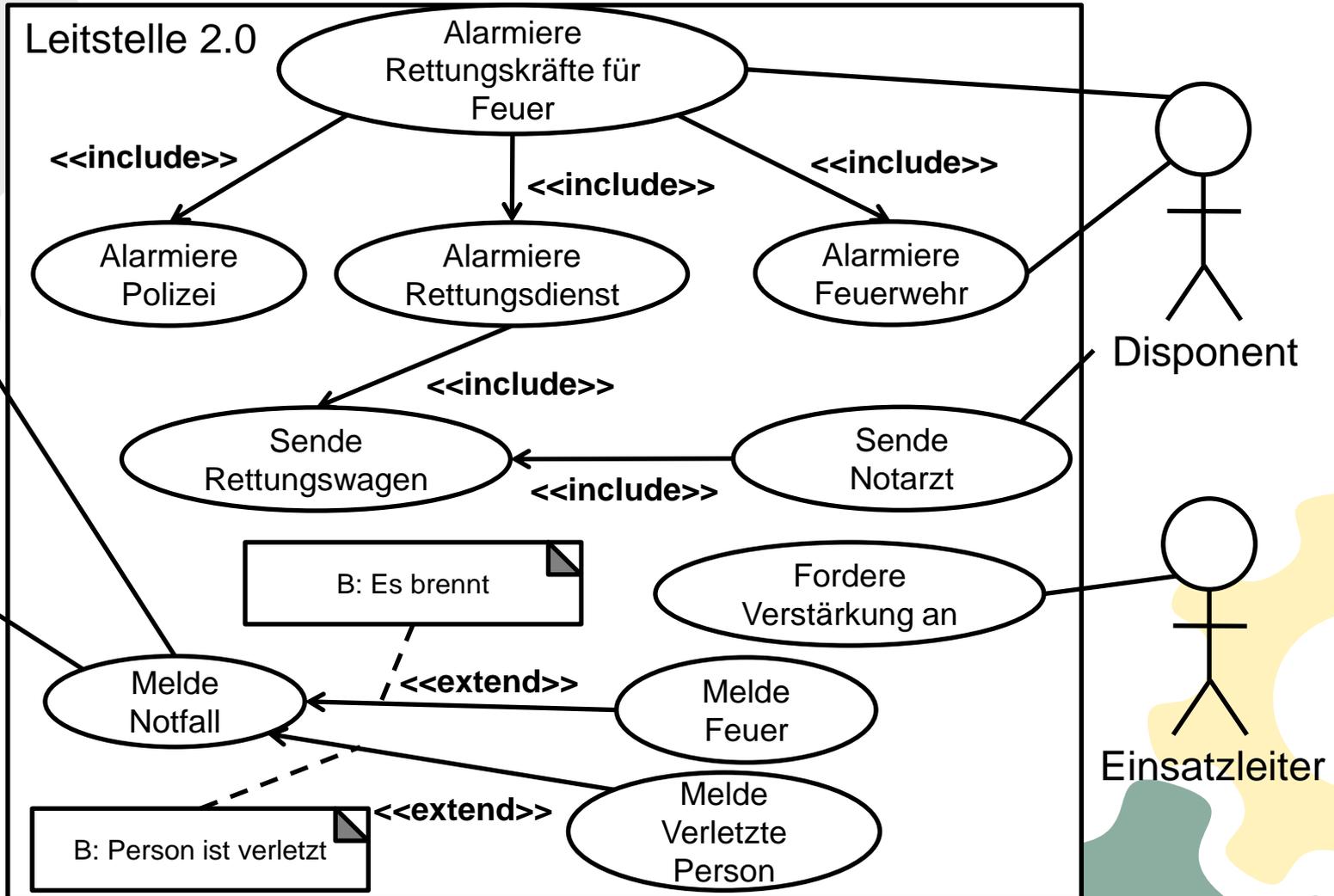
*Eine alte Frau, die währenddessen am Kaufhaus vorbei läuft, verliert vor lauter Aufregung das Bewusstsein und verletzt sich am Kopf. Ein aufmerksamer **Passant** bemerkt dies und **meldet der Leitstelle den Unfall**. Der Disponent wird auch auf diesen Notfall hingewiesen und **sendet einen Notarzt** zur Einsatzstelle. Zusammen mit dem Notarzt wird ein **weiterer Rettungswagen** zur Einsatzstelle geschickt.*

Akteur

Anwendungsfall

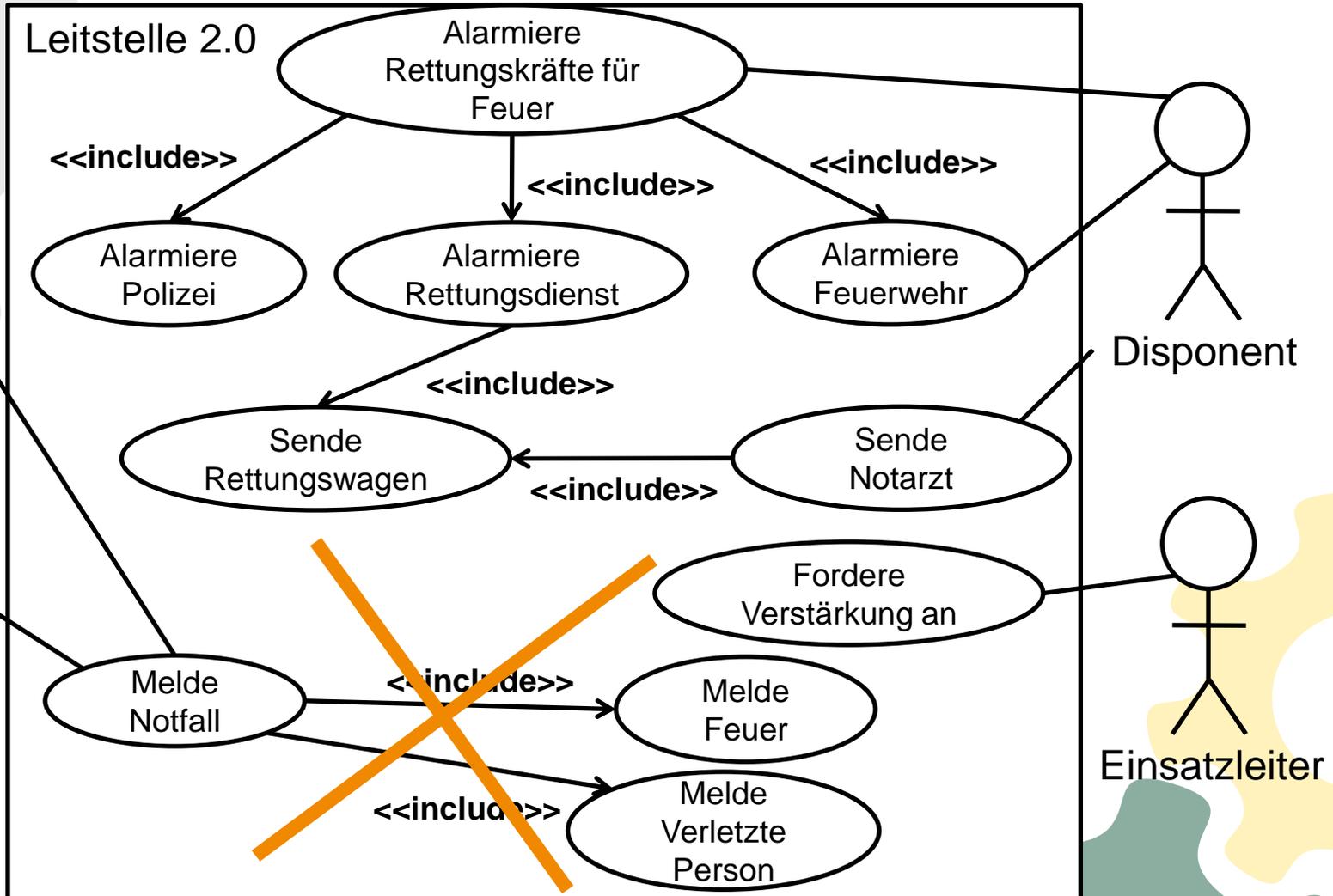


# Aufgabe 2a



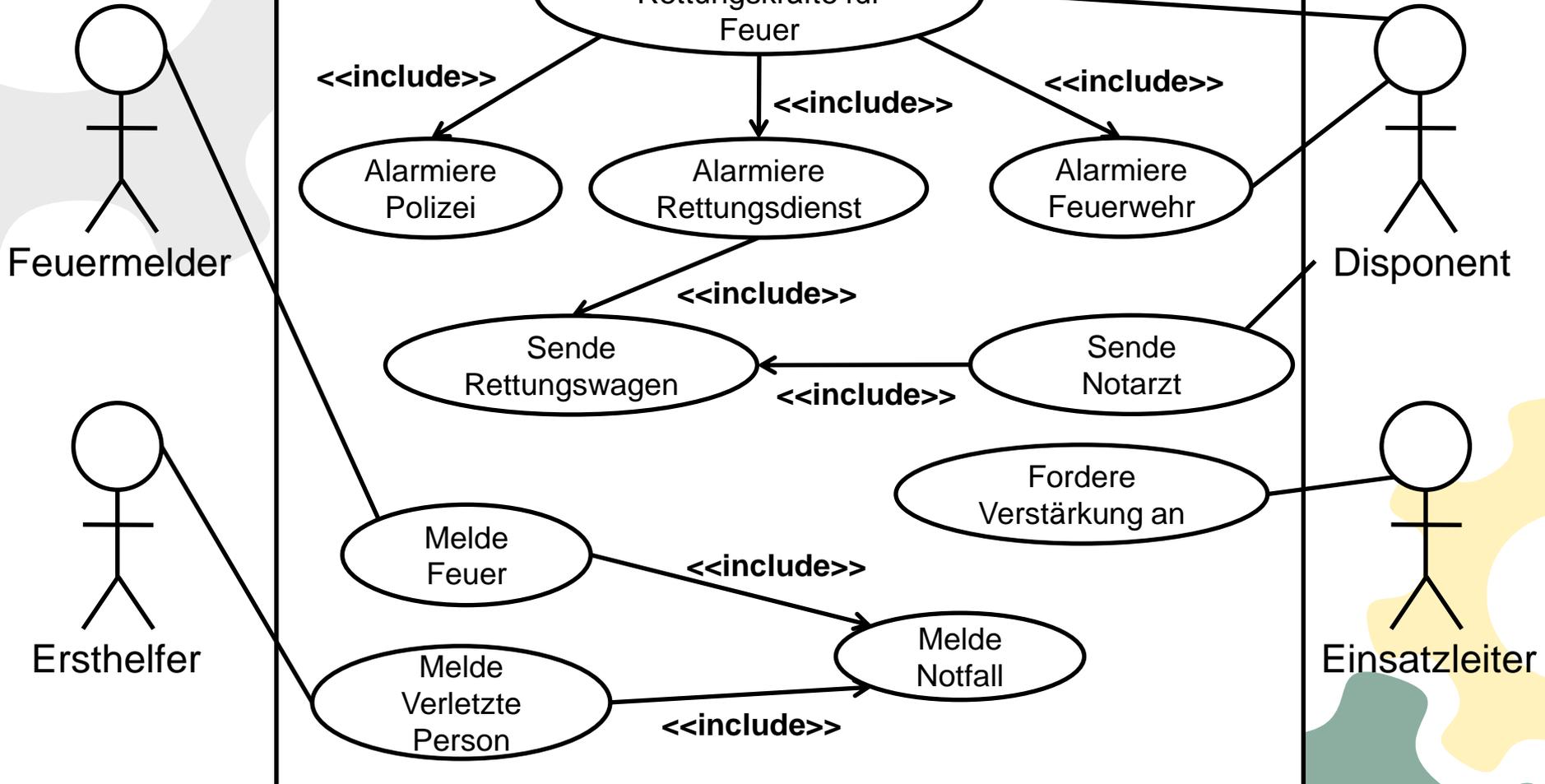


# Aufgabe 2a





# Aufgabe 2a





# Aufgabe 2

- b) Beschreiben Sie mindestens zwei der von Ihnen ermittelten Anwendungsfälle in Textform. Orientieren Sie sich dabei an dem in der Vorlesung vorgestellten Beispiel.



# Aufgabe 2b

- **Name:**

- Alarmiere Feuerwehr

- **Teilnehmender Akteur:**

- Disponent

- **Eingangsbedingung:**

- Feuer wird gemeldet
- Einsatzkräfte sind vorhanden

- **Ausgangsbedingung:**

- Feuerwehr ist auf dem Weg zu Einsatzort

- **Ereignisfluss:**

- Disponent wird Feuer gemeldet
- Disponent gibt Einsatz in seinen Computer ein
- Disponent ermittelt die notwendige Feuerwehr
- Disponent informiert die Feuerwehr über Einsatzart und -ort

- **Spezielle Anforderungen:**

- Keine



# Aufgabe 2b

- **Name:**

- Fordere Verstärkung an

- **Teilnehmender Akteur:**

- Einsatzleiter

- **Eingangsbedingung:**

- Einsatzleiter ist am Einsatzort
- Einsatzleiter hat die Gefahrenlage eingeschätzt

- **Ausgangsbedingung:**

- Verstärkung wird alarmiert

- **Ereignisfluss:**

- Einsatzleiter kommt an Einsatzort an
- Einsatzleiter verschafft sich einen Überblick über die aktuelle Lage
- Einsatzleiter fordert weitere Einsatzkräfte an
- Einsatzleiter bekommt Rückmeldung, dass Verstärkung unterwegs ist

- **Spezielle Anforderungen:**

- Keine



## Aufgabe 3

Erstellen Sie ein Programm mit einer grafischen Swing-Benutzeroberfläche, das es dem Benutzer ermöglicht, auf einer mindestens  $400 \times 400$  Pixel großen weißen Zeichenfläche frei mit der Maus zu zeichnen. Dem Benutzer sollten dabei außer Schwarz mindestens drei weitere Farben zur Auswahl stehen. Zudem soll der Benutzer die Auswahl zwischen mindestens 5 Pinselgrößen zwischen 1 und 20 Pixel haben. Zusätzlich soll der Benutzer die Zeichenfläche wieder leeren können, d.h. ein neues Bild zeichnen können, ohne die Anwendung neu starten zu müssen.



## Aufgabe 3 – Bepunktung

- Mindestanforderungen an den Quellcode/das Programm:
  - Der eingereichte Quellcode muss von uns ohne Fehler übersetzt werden können.
  - Das übersetzte Programm muss lauffähig sein.
- Erfüllt das Programm eine dieser Anforderungen nicht, wird die Aufgabe mit **0 Punkten** bewertet.

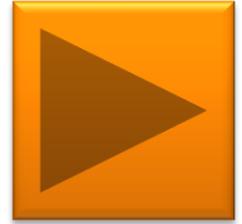


# Aufgabe 3 – Bepunktung

- Fehlende Funktionalität (zu wenig angebotene Farben, zu wenig Pinselgrößen, etc.) wird mit Punktabzug bewertet.
- Schlecht kommentierter und schlecht lesbarer Code führen in Extremfällen ebenfalls zu Punktabzug. Darunter fallen unter anderem:
  - Grobe Verstöße gegen die Richtlinien für Java-Quellcode
  - Formatierung des Codes nicht einheitlich bzw. nicht vorhanden
  - Keine einheitliche Benennung von Variablen, Konstanten, ...



## Aufgabe 3



- Besonderheiten
  - Beim naiven Zeichnen in das Fenster gehen beim neuzeichnen des Fensters alle Zeichnungen verloren, sofern sie nicht zwischengespeichert wurden.
    - Zeichne in Puffer (z.B. ein `Image`) oder speichere alle Zeichnungen in einer separaten Datenstruktur.
  - Der `MouseListener` ist zu langsam
    - Interpoliere zwischen zwei Ereignissen z.B. mit einer Linie



# Aufgabe 3

```
private Image image;
```

```
[...]
```

```
private void init() {
```

```
    image = new BufferedImage(400, 400,  
        BufferedImage.TYPE_INT_RGB);
```

```
    Graphics2D g2d = (Graphics2D) image.getGraphics();
```

```
    g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING,  
        RenderingHints.VALUE_ANTIALIAS_ON);
```

```
    g2d.setBackground(Color.WHITE);
```

```
    g2d.clearRect(0, 0, 400, 400);
```

```
}
```



# Aufgabe 3

Besser `paintComponent()` statt  
`paint()` überschreiben.

```
@Override  
public void paintComponent(Graphics g) {  
    g.drawImage(image, 0, 0, this);  
}
```



## Aufgabe 3

```
private void drawLine(Point from, Point to) {  
    Graphics2D g2d = (Graphics2D) image.getGraphics();  
    g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING,  
        RenderingHints.VALUE_ANTIALIAS_ON);  
    g2d.setColor(strokeColor);    // Gesetzt durch GUI  
    g2d.setStroke(new BasicStroke(strokeSize,  
        BasicStroke.CAP_ROUND, BasicStroke.JOIN_MITER));  
    g2d.drawLine(from.x, from.y, to.x, to.y);  
}
```



## Aufgabe 3

```
addMouseListener(new MouseMotionAdapter() {  
    @Override  
    public void mouseDragged(MouseEvent e) {  
        Point currPos = e.getPoint();  
        drawLine(currPos, lastPos);  
        repaint();  
        lastPos = currPos;  
    }  
});
```