

Universität Karlsruhe (TH)

Forschungsuniversität · gegründet 1825

Softwaretechnik 1

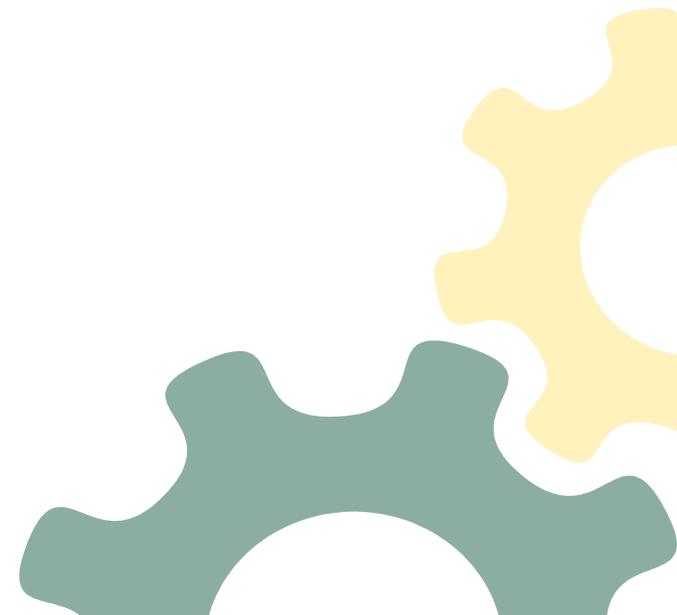
Übung 3

4.6.2009



Fakultät für **Informatik**

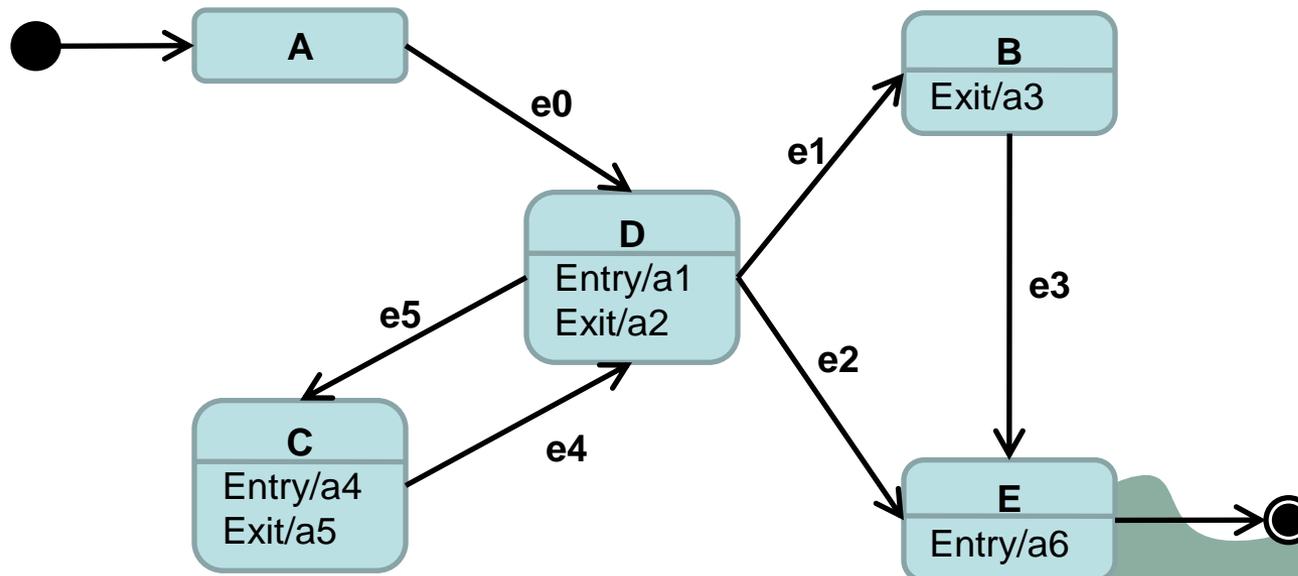
Lehrstuhl für Programmiersysteme





Aufgabe 1a)

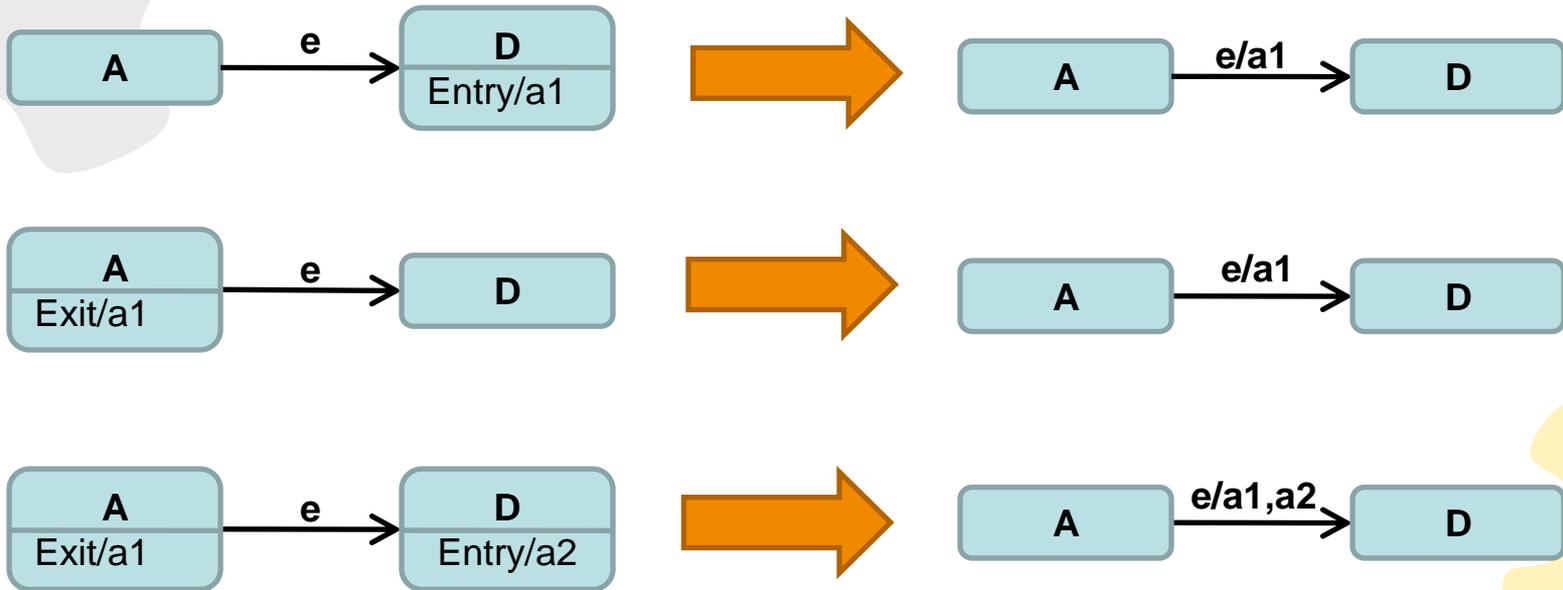
- Wandeln Sie folgenden UML-Zustandsautomaten in einen äquivalenten Mealey-Automaten um. Die im UML-Zustandsautomaten angegebenen Entry-/Exit-Aktionen können Sie als Ausgabewerte auffassen.





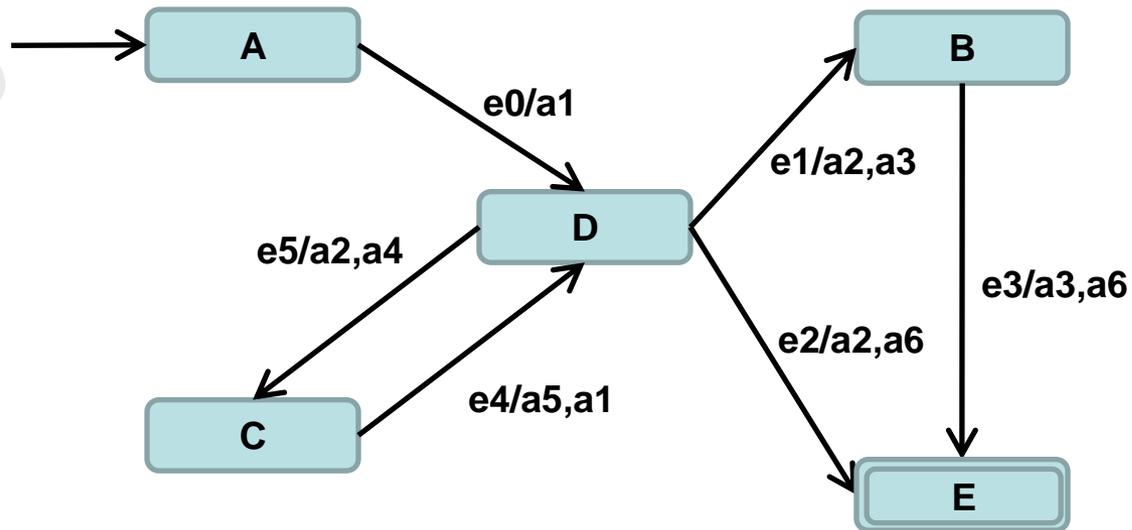
Aufgabe 1a)

- Vorgehensweise





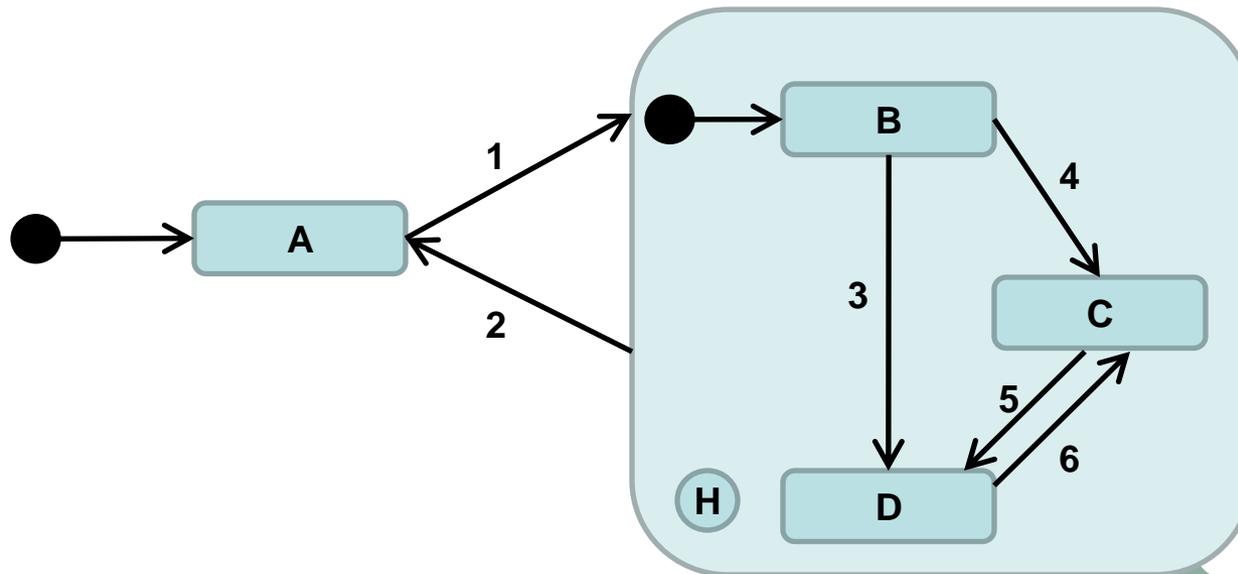
Aufgabe 1a)





Aufgabe 1b)

- Wandeln Sie folgenden, hierarchischen UML-Zustandsautomaten mit Gedächtnis in einen äquivalenten, nicht hierarchischen UML-Zustandsautomaten ohne Gedächtnis um.



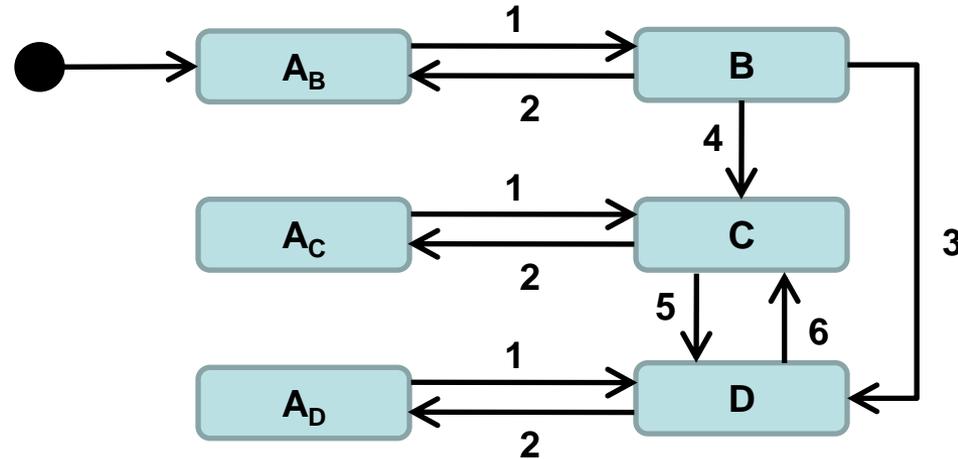


Aufgabe 1b)

- Das Gedächtnis des Zustandsautomaten kann durch vervielfältigen des Zustands A modelliert werden.
- Von den 3 „A“-Zuständen muss der korrekte Zustand als Startzustand markiert werden.



Aufgabe 1b)





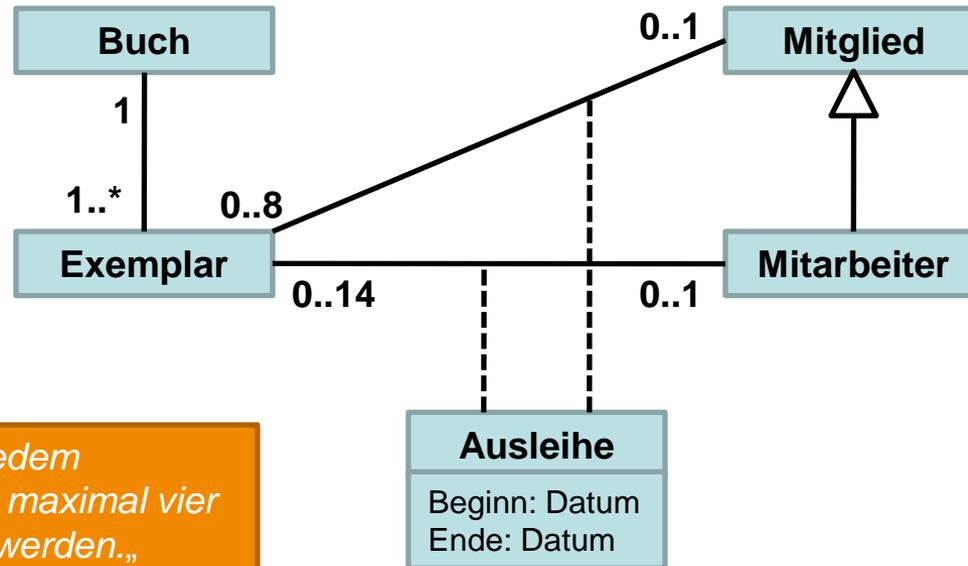
Aufgabe 2)

In einer Bibliothek können Bücher ausgeliehen werden. Von jedem Buch kann es ein oder mehrere Exemplare geben. Bücher können von jedem Bibliotheksmitglied für maximal vier Wochen ausgeliehen werden. Mitglieder der Bibliothek dürfen maximal acht Exemplare entleihen. Ist das Mitglied jedoch ein Mitarbeiter der Bibliothek, so darf es maximal 14 Exemplare ausleihen.

Finden Sie in obigem Szenario alle beteiligten Klassen sowie Assoziationen zwischen den Klassen und zeichnen Sie ein Klassendiagramm. Berücksichtigen Sie dabei Multiplizitäten, Rollen und Leserichtungen.



Aufgabe 2)

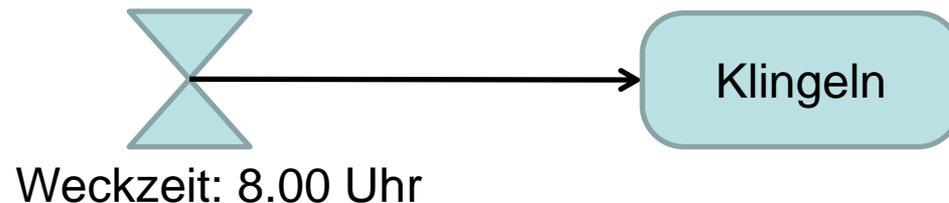


„Bücher können von jedem Bibliotheksmitglied für maximal vier Wochen ausgeliehen werden.“



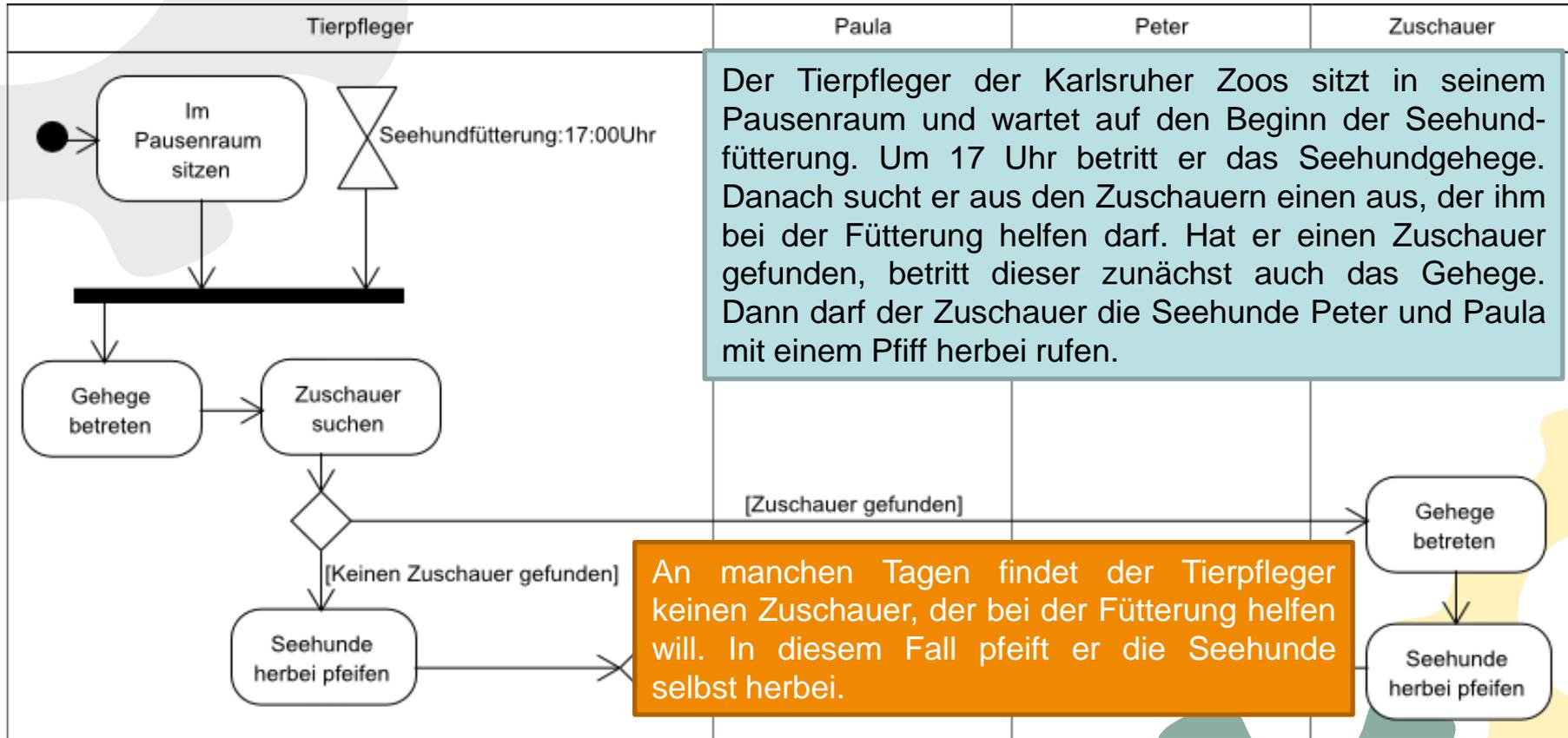
Aufgabe 3)

- Erstellen Sie ein Aktivitätsdiagramm zum gegebenen Szenario.
 - Geben Sie evtl. vorhandene Ein-/Ausgabeobjekte an
 - Beachten Sie vorhandene Zeitsignale und stellen Sie sie dar.





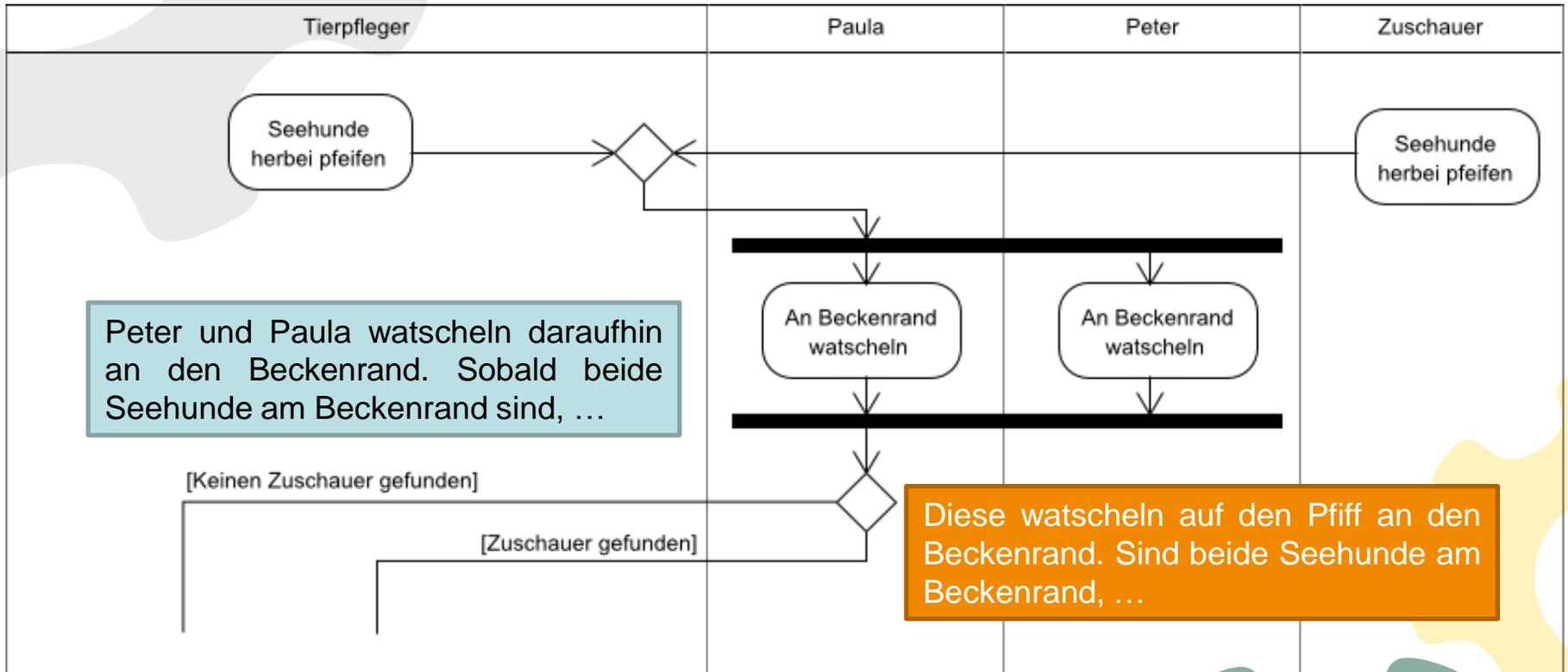
Aufgabe 3)



An manchen Tagen findet der Tierpfleger keinen Zuschauer, der bei der Fütterung helfen will. In diesem Fall pfeift er die Seehunde selbst herbei.



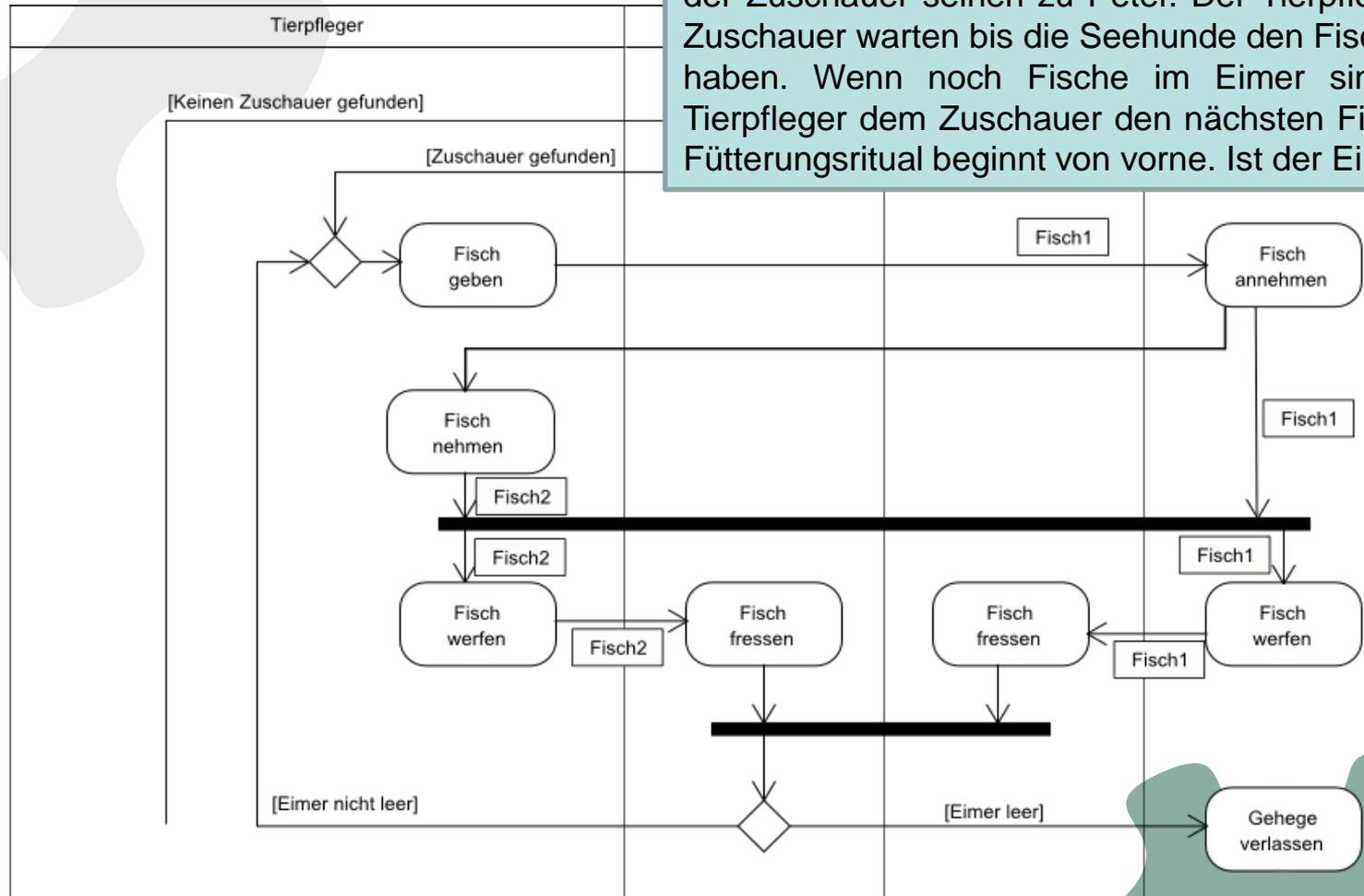
Aufgabe 3)





Aufgabe 3)

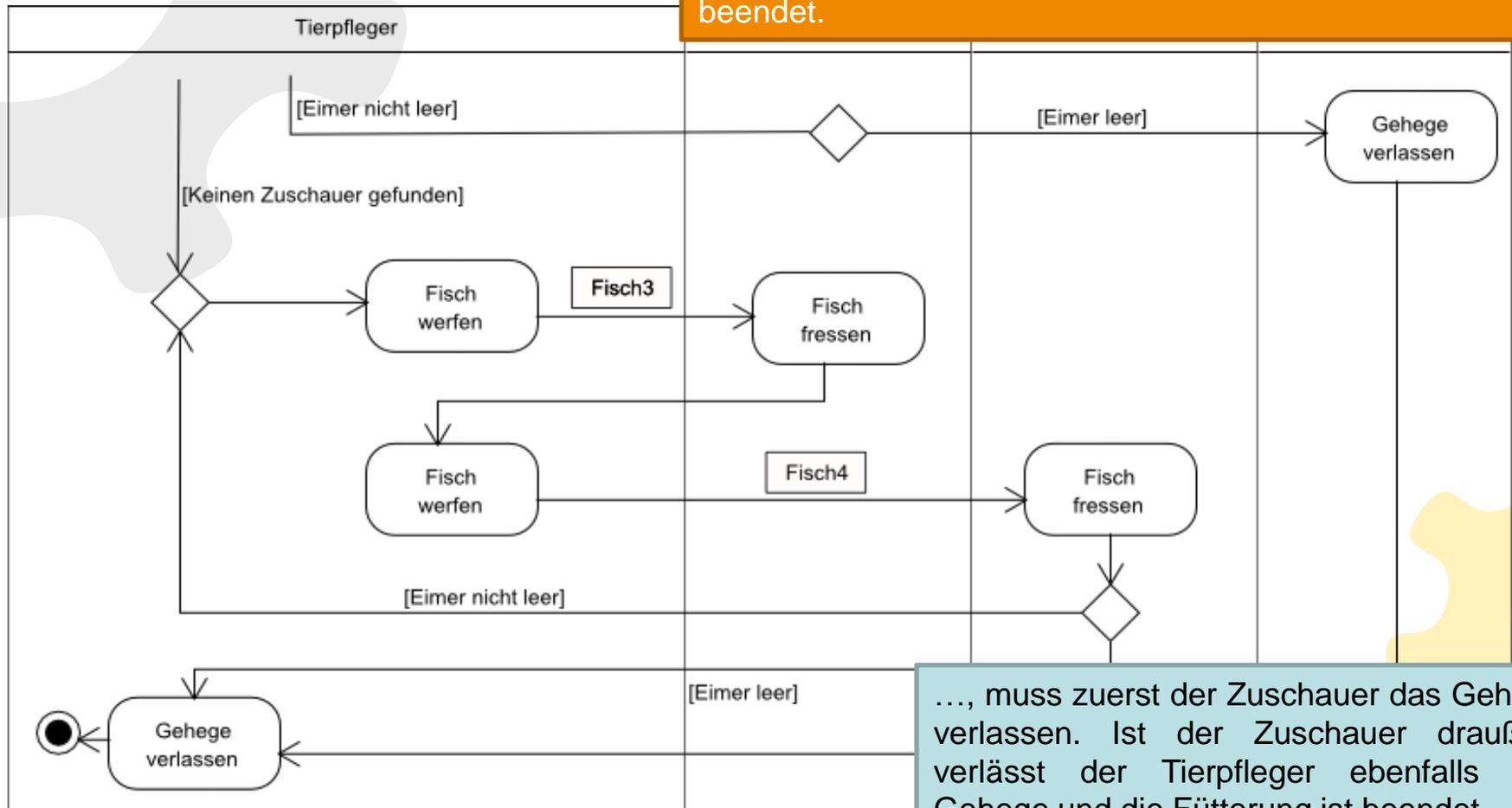
... gibt der Tierpfleger dem Zuschauer einen Fisch aus dem Eimer. Der Zuschauer nimmt den Fisch an. Nun nimmt sich der Tierpfleger auch einen Fisch aus dem Eimer. Anschließend werfen der Zuschauer und der Tierpfleger gleichzeitig den Seehunden ihren Fisch zu. Der Tierpfleger wirft seinen Fisch zur trächtigen Paula, der Zuschauer seinen zu Peter. Der Tierpfleger und der Zuschauer warten bis die Seehunde den Fisch gefressen haben. Wenn noch Fische im Eimer sind, gibt der Tierpfleger dem Zuschauer den nächsten Fisch und das Fütterungsritual beginnt von vorne. Ist der Eimer leer, ...





Aufgabe 3)

... wirft der Tierpfleger zuerst der trächtigen Paula einen Fisch zu. Hat Paula den Fisch aufgefressen, bekommt Peter einen Fisch zugeworfen. Hat Peter den Fisch aufgefressen und es befinden sich noch Fische im Eimer, beginnt das Fütterungsritual von vorne. Ist der Eimer leer, verlässt der Tierpfleger das Gehege und die Fütterung ist beendet.



..., muss zuerst der Zuschauer das Gehege verlassen. Ist der Zuschauer draußen, verlässt der Tierpfleger ebenfalls das Gehege und die Fütterung ist beendet.



Aufgabe 4)



- Das haben wir uns vorgestellt:
 - Erstellen eines eigenen `TableCellRenderers`
 - Erstellen eines eigenen `TableModel`s

- Andere Lösungen sind auch möglich!



Aufgabe 4)

- Das `TableModel` ist für die Organisation der Adress-Daten zuständig.
- Die Funktionalität zum Hinzufügen, Aktualisieren und Löschen von Adressen sollte daher im `TableModel` implementiert werden.
- Die Aktionen `Add/Edit/RemoveAddressAction` delegieren die Arbeit an die Tabelle bzw. das `TableModel`.



Aufgabe 4)

```
public class AddressBook extends JFrame {

    private class EditAddressAction
        extends AbstractAction {

        ...
        public void actionPerformed(ActionEvent e){
            Address address =
                table.getSelectedAddress();
            if (address == null) {
                return;
            }
            AddressDialog dialog = new
                AddressDialog(address,
                    AddressBook.this);
            dialog.setVisible(true);
            if (dialog.getSelectedOption() ==
                Option.SAVE) {
                table.updateSelectedAddress(
                    dialog.getAddress());
            }
        }
    }
}
```

```
public class AddressTable extends JTable {

    ...
    public void updateSelectedAddress(
        Address address) {
        int index = getSelectedRow();
        if (index <= -1) {
            return;
        }
        model.updateAddress(index, address);
    }
}
```

```
public class AddressTableModel extends
    AbstractTableModel {
    private List<Address> addresses =
        new ArrayList<Address>();

    public void updateAddress(int index,
        Address address) {
        addresses.get(index).copyAll(address);
        fireTableRowsUpdated(index, index);
    }
}
```



Aufgabe 4)

- Bei der Implementierung eines TableCellRenderer müssen Sie die Methode `getTableCellRendererComponent(JTable table, Object value, boolean isSelected, boolean hasFocus, int row, int column)` implementieren.
- Die Methode gibt eine Komponente zurück, welche mit der `paint()`-Methode den übergebenen Wert darstellt.
- Sie können bspw. ein Objekt der Klasse `JLabel` verwenden.



Aufgabe 4)

```
private JLabel label;
```

```
public Component getTableCellRendererComponent(JTable table, Object value,  
boolean isSelected, boolean hasFocus, int row, int column) {
```

```
    if (label == null) {  
        label = new JLabel();  
        label.setFont(table.getFont());  
        label.setOpaque(true);  
    }
```

Nur ein Label!

```
    String str = value.toString();  
    label.setText(str);
```

```
    int modelRow = table.convertRowIndexToModel(row);  
    AddressTableModel model = (AddressTableModel) table.getModel();  
    Address address = model.getAddress(modelRow);  
    Category category = address.getCategory();  
    ColorScheme colorScheme = category.getColorScheme();  
    Color bg = isSelected ? colorScheme.getSelectionBackground() :  
        colorScheme.getBackground();  
    label.setBackground(bg);
```

```
    return label;
```

```
}
```