GI-Fachgruppe
Informatische Bildung in NRW

Informatische Bildung in NRW

Unterrichtsmaterialien für die Sekundarstufe II

Johannes Pieper

Eine Alarmanlage mit dem MVC-Konzept

Modellierung, Implementierung und Umsetzung am Raspberry Pi

Autoren

Dieses Dokument ist ein sogenanntes Modul der Materialsammlung, die von der Fachgruppe Informatische Bildung in NRW der Gesellschaft für Informatik heraus gegeben wird. Verantwortliche Autoren für dieses Dokument: Johannes Pieper

Lizenz

Dieses Dokument steht unter der Creative Commons by-nc-sa-Lizenz. Folglich darf es beliebig kopiert und bearbeitet werden, sofern das Folgeprodukt wiederum unter dieser Lizenz vertrieben wird. Eine kommerzielle Nutzung ist nicht erlaubt. https://creativecommons.org/licenses/by-nc-sa/4.0/deed.de

Verweise auf Material

Dieses Modul enthält Hinweise und Verweise (Links) auf Material.Wird eine Seite des Bezugsmaterials dargestellt, ist dieses immer die erste, auch wenn das entsprechende Material aus mehreren besteht. Die Weltkugel verweist dabei auf die entsprechende Webseite und das Dokumentensymbol hat einen relativen Pfad hinterlegt, der die Nutzung der Verzeichnisstruktur der Materialsammlung voraussetzt.

Kontakt

Ihre Ansprechpartner für die vorliegenden Materialien sind Daniel Losch daniel.losch@gi.de und Johannes Pieper pieper@informatiktag-nrw.de

bau_einer_alarmanlage.tex

git

Version: dd1fc2
Stand: 26. Juni 2026 18:18
Zuletzt bearbeitet von: Johannes Pieper

1 Beschreibung

Fast alle Softwareprojekte setzen heute das MVC-Konzept ein. Auch im Unterricht sollte daher dieser Form der Struktur einer Software behandelt werden. Gleichzeitig lassen sich damit die Konzepte von Schnittstellen und abstrakten Klassen behandeln. Während der Durchführung des Moduls werden die Schülerinnen und Schüler schrittweise die mit den neuen Elementen vertraut gemacht.

Als Aufhänger dieses Moduls dient das Funktionsprinzip einer Alarmanlage die es zu modellieren und implementieren gilt. Dabei soll die Implementierung nicht auf die virtuelle Welt des Computers beschränkt bleiben, sondern auf dem Raspberry Pi die Kommunikation mit der Umwelt durchführen.

2 Ablauf und Material

Der Ablauf ist in die mehreren Teile des Moduls unterteilt. Das entsprechende Material ist direkt damit verlinkt, bzw. eingebunden.

Achtung: Es gibt verschiedene Auslegungen beim MVC, welche Aufgabe die drei Teile übernehmen und wie sie miteinander kommunizieren.

2.1 Einstieg in die Modellierung der Alarmanlage

Im ersten Teil geht es darum, dass sich die Schülerinnen und Schüler mit den allgemeinen Funktionsweisen von Alarmanlagen befassen. Dabei sollen die Lernenden selber ihr Objekt bestimmen, dass durch die Alarmanlage beschützt werden soll. Auch die Komponenten der entsprechenden Alarmanlage wie z. B. eine Sirene, eine Warnleuchte oder ein Taster zur Detektion, werden selber ausgewählt. Dieses geschieht alles in einer Gruppe, die ihre selbstständige erste Modellierung dem gesamten Kurs vorstellen sollen. Die Vorstellung der einzelnen Gruppen hat den Hintergrund, dass erkannt wird, dass alle Alarmanlagen in etwa den gleichen Kern haben und bei den Funktionsweisen der Komponenten Überschneidungen gibt.

Im Hinblick auf die Implementierung wird im Unterrichtsgespräch mit den Lernenden geklärt, dass wichtige Komponenten für alle Gruppen nur ein einziges Mal implementiert werden müssen. Voraussetzung dafür ist aber, dass es abgesprochene Schnittstellen gibt. Auch die Unterteilung in verschiedene Arten von Komponenten der Software sollte hier angesprochen werden.

Material

- ab_alarmanlagen_modellierung.pdf



Informatik *Alarmanlagen* *Arbeitsblatt*

Alarmanlagen

Bei vielen Objekten wie Häusern, Autos, einem Firmengelände oder Koffern werden Alarmanlagen eingesetzt um diese vor Diebstahl zu schützen. Teilweise ist mit einer solchen Alarmanlage auch eine Zugriffskontrolle verbunden. Dabei gibt es verschiedene Arten, wie ein Alarm überhaupt ausgelöst wird und wie dieser angezeigt wird. Auch eine Benachrichtigung des Besitzers ist teilweise möglich. Nicht nur die Art und Weisen der Alarmanlagen ist sehr unterschiedlich, sondern auch welche Objekte durch sie geschützt werden. Dieses wird deutlich bei dem Vergleich zwischen der klassischen Alarmanlage am Haus und einer für Autos, Fahrräder oder vielen anderen Objekten.

Im Unterricht soll nun das Modell einer Alarmanlage als Gruppenprojekt zuerst modelliert und später auch implementiert, sowie praktisch umgesetzt werden.

Aufgabe 1

Legen Sie in Ihrer Gruppe fest, welche Art von Alarmanlage sie näher betrachten wollen.

Aufgabe 2

Überlegen Sie aus welchen Komponenten Ihre Alarmanlage bestehen soll und wie diese zusammenhängen sollen. Beachten Sie dabei, dass hier es hier eher auf Details ankommt als auf eine möglichst große Anzahl von Komponenten.

Aufgabe 3

Modellieren Sie die Alarmanlage ihrer Gruppe. Erstellen Sie dazu auch passende Klassenkarten und setzten Sie diese in einem Klassendiagramm in Beziehung. Arbeiten Sie dazu in Ihrer Gruppe arbeitsteilig.

Aufgabe 4

Bereiten Sie Ihre Modellierung so vor, dass jeder aus der Gruppe in der Lage ist, die Alarmanlage vorzustellen.

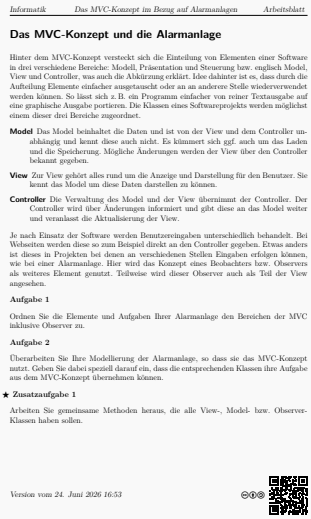
Version vom 24. Juni 2026 16:53

2.2 Das Konzept von MVC und Schnittstellen

Mit dem Arbeitsblatt zu MVC bekommen die Lernenden einen genaueren Einblick und sind dazu aufgefordert ihre Modellierung im Sinne von MVC anzupassen. Schnellere können sich dabei auch schon die ersten Gedanken zu Schnittstellen machen, was in der anschließenden Plenumsphase genutzt werden kann. In dieser geht es hauptsächlich noch mal genauer die Unterschiede zwischen den drei Bereichen von MVC und den zusätzlichen Beobachtern deutlich zu machen. Außerdem sollte die Zuordnung überprüft werden. Mit einem zusätzlichen Informationsblatt kann man das Verständnis für MVC und dessen Abläufe stärken, da es mit einem Klassendiagramm und einem Sequenzdiagramm wichtige Elemente visualisiert.

Material

- ab_alarmanlagen_mvc.pdf



Informatik *Das MVC-Konzept im Bezug auf Alarmanlagen* *Arbeitsblatt*

Das MVC-Konzept und die Alarmanlage

Hinter dem MVC-Konzept versteckt sich die Einteilung von Elementen einer Software in drei verschiedene Bereiche: Modell, Präsentation und Steuerung bzw. englisch Model, View und Controller, was auch die Abkürzung erklärt. Idee dahinter ist es, dass durch die Aufteilung Elemente einfacher ausgetauscht oder an an anderere Stelle wiederverwendet werden können. So lässt sich z. B. ein Programm einfacher von reiner Textausgabe auf eine graphische Ausgabe portieren. Die Klassen eines Softwareprojekts werden möglichst einem dieser drei Bereiche zugeordnet.

Model Das Model beinhaltet die Daten und ist von der View und dem Controller unabhängig und kennt diese auch nicht. Es kümmert sich ggf. auch um das Laden und die Speicherung. Mögliche Änderungen werden der View über den Controller bekannt gegeben.

View Zur View gehört alles rund um die Anzeige und Darstellung für den Benutzer. Sie kennt das Model um diese Daten darstellen zu können.

Controller Die Verwaltung des Model und der View übernimmt der Controller. Der Controller wird über Änderungen informiert und gibt diese an das Model weiter und veranlasst die Aktualisierung der View.

Je nach Einsatz der Software werden Benutzereingaben unterschiedlich behandelt. Bei Webseiten werden diese so zum Beispiel direkt an den Controller gegeben. Etwas anders ist dieses in Projekten bei denen an verschiedenen Stellen Eingaben erfolgen können, wie bei einer Alarmanlage. Hier wird das Konzept eines Beobachters bzw. Observers als weiteres Element genutzt. Teilweise wird dieser Observer auch als Teil der View angesehen.

Aufgabe 1

Ordnen Sie die Elemente und Aufgaben Ihrer Alarmanlage den Bereichen der MVC inklusive Observer zu.

Aufgabe 2

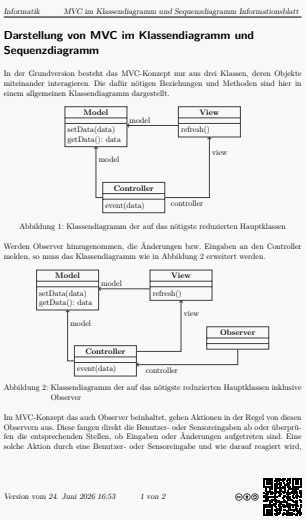
Überarbeiten Sie Ihre Modellierung der Alarmanlage, so dass sie das MVC-Konzept nutzt. Geben Sie dabei speziell darauf ein, dass die entsprechenden Klassen ihre Aufgabe aus dem MVC-Konzept übernehmen können.

★ **Zusatzaufgabe 1**

Arbeiten Sie gemeinsame Methoden heraus, die alle View-, Model- bzw. Observer-Klassen haben sollen.

Version vom 24. Juni 2026 16:53

- ib_mvc_klassenmodell_ablaeufe.pdf



Informatik *MVC im Klassendiagramm und Sequenzdiagramm Informationsblatt*

Darstellung von MVC im Klassendiagramm und Sequenzdiagramm

In der Grundversion besteht das MVC-Konzept nur aus drei Klassen, deren Objekte miteinander interagieren. Die dafür nötigen Beziehungen und Methoden sind hier in einem allgemeinen Klassendiagramm dargestellt.

Abbildung 1: Klassendiagramm der auf das nötigste reduzierten Hauptklassen

Werden Observer hinzugenommen, die Änderungen bzw. Eingaben an den Controller melden, so muss das Klassendiagramm wie in Abbildung 2 erweitert werden.

Abbildung 2: Klassendiagramm der auf das nötigste reduzierten Hauptklassen inklusive Observer

Im MVC-Konzept das auch Observer beinhaltet, gehen Aktionen in der Regel von diesen Observern aus. Diese fangen direkt die Benutzer- oder Sensoreingaben ab oder überprüfen die entsprechenden Stellen, ob Eingaben oder Änderungen aufgetreten sind. Eine solche Aktion durch eine Benutzer- oder Sensoreingabe und wie darauf reagiert wird,

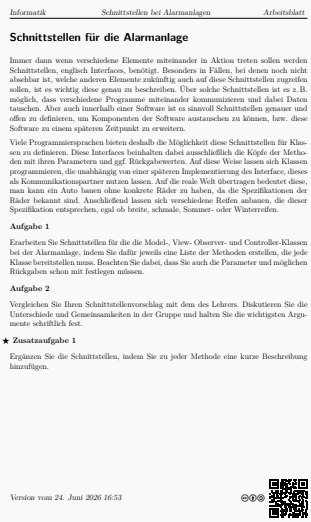
Version vom 24. Juni 2026 16:53 *1 von 2*

Durch ein weiteres Arbeitsblatt wird den Lernenden die Bedeutung von Schnittstellen und ihre Funktionsweise nahegelegt. Sie müssen eigene Schnittstellen erstellen und diese mit den vorgegebenen vergleichen. Die Vorgaben haben den Hintergrund, dass der Controller für die Lernenden schon fertig programmiert ist und bereits mit diesen Schnittstellen arbeitet.

In der Plenumsphase kann bei den Ergebnisse der Gruppen besonders noch mal auf die Unterschiede zwischen den eigenen und den vorgegebenen Schnittstellen eingegangen werden. Es könnte sich dann empfehlen, die Schnittstellen noch für die Schülerinnen und Schüler anzupassen und auch den Controller entsprechend zu modifizieren, bevor er herausgegeben wird.

Material

- ab_alarmanlagen_schnittstellen.pdf



Informatik *Schnittstellen bei Alarmanlagen* *Arbeitsblatt*

Schnittstellen für die Alarmanlage

Immer dann wenn verschiedene Elemente miteinander in Aktion treten sollen werden Schnittstellen, englisch Interfaces, benötigt. Besonders in Fällen, bei denen noch nicht absehbar ist, welche anderen Elemente zukünftig auch auf diese Schnittstellen zugreifen sollen, ist es wichtig diese genau zu beschreiben. Über solche Schnittstellen ist es z. B. möglich, dass verschiedene Programme miteinander kommunizieren und dabei Daten tauschen. Aber auch innerhalb einer Software ist es sinnvoll Schnittstellen genauer und offen zu definieren, um Komponenten der Software austauschen zu können, bzw. diese Software zu einem späteren Zeitpunkt zu erweitern.

Viele Programmiersprachen bieten deshalb die Möglichkeit diese Schnittstellen für Klassen zu definieren. Diese Interfaces beinhalten dabei ausschließlich die Köpfe der Methoden mit ihren Parametern und ggf. Rückgabewerten. Auf diese Weise lassen sich Klassen programmieren, die unabhängig von einer späteren Implementierung des Interface, dieses als Kommunikationspartner nutzen lassen. Auf die reale Welt übertragen bedeutet diese, man kann ein Auto bauen ohne konkrete Räder zu haben, da die Spezifikationen der Räder bekannt sind. Anschließend lassen sich verschiedene Reifen anbauen, die dieser Spezifikation entsprechen, egal ob breite, schmale, Sommer- oder Winterreifen.

Aufgabe 1

Erarbeiten Sie Schnittstellen für die die Model-, View- Observer- und Controller-Klassen bei der Alarmanlage, indem Sie dafür jeweils eine Liste der Methoden erstellen, die jede Klasse bereitstellen muss. Beachten Sie dabei, dass Sie auch die Parameter und möglichen Rückgaben schon mit festlegen müssen.

Aufgabe 2

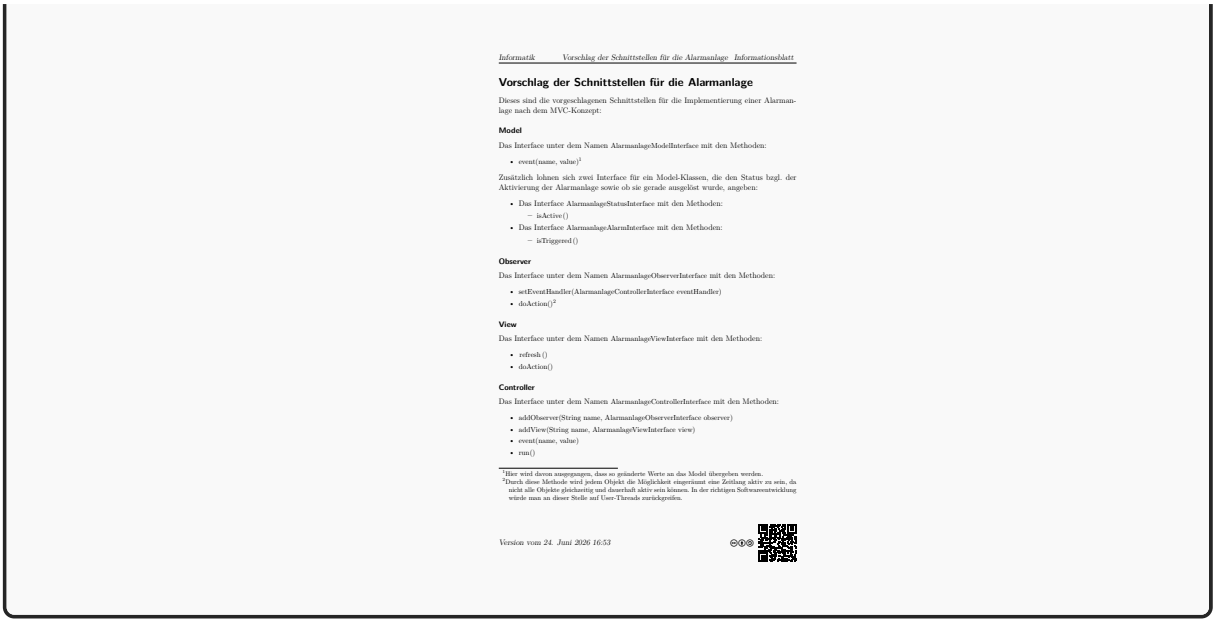
Vergleichen Sie Ihren Schnittstellenvorschlag mit dem des Lehrers. Diskutieren Sie die Unterschiede und Gemeinsamkeiten in der Gruppe und halten Sie die wichtigsten Argumente schriftlich fest.

★ Zusatzaufgabe 1

Ergänzen Sie die Schnittstellen, indem Sie zu jeder Methode eine kurze Beschreibung hinzufügen.

Version vom 24. Juni 2026 16:53

- ib_alarmanlagen_schnittstellen_vorschlag.pdf



Informatik *Vorschlag der Schnittstellen für die Alarmanlage* *Informationsblatt*

Vorschlag der Schnittstellen für die Alarmanlage

Dieses sind die vorgeschlagenen Schnittstellen für die Implementierung einer Alarmanlage nach dem MVC-Konzept:

Model

Das Interface unter dem Namen AlarmanlageModelInterface mit den Methoden:

- event(name, value)[1]

Zusätzlich kohnen sich zwei Interface für ein Model-Klassen, die den Status bzgl. der Aktivierung der Alarmanlage sowie ob sie gerade ausgelöst wurde, angeben:

- Das Interface AlarmanlageStatusInterface mit den Methoden:
  - isActive()
- Das Interface AlarmanlageAlarmInterface mit den Methoden:
  - isTriggered()

Observer

Das Interface unter dem Namen AlarmanlageObserverInterface mit den Methoden:

- setEventHandler(AlarmanlageControllerInterface eventHandler)
- doAction()[2]

View

Das Interface unter dem Namen AlarmanlageViewInterface mit den Methoden:

- refresh()
- doAction()

Controller

Das Interface unter dem Namen AlarmanlageControllerInterface mit den Methoden:

- addObserver(String name, AlarmanlageObserverInterface observer)
- addView(String name, AlarmanlageViewInterface view)
- event(name, value)
- run()

[1] Hier wird davon ausgegangen, dass so geänderte Werte an das Model übergeben werden.
[2] Durch diese Methode wird jedem Objekt die Möglichkeit eingeräumt eine Zeitlang aktiv zu sein, da nicht alle Objekte gleichzeitig und dauerhaft aktiv sein können. In der richtigen Softwareentwicklung würde man an dieser Stelle auf User-Threads zurückgreifen.

Version vom 24. Juni 2026 16:53

Im nächsten Schritt wird eine Schnittstelle von den Lernenden in ihrer Programmiersprache umgesetzt. Für sie kann im gleichen Zug eine Dummy-Klasse erstellt werden, die zusammen mit dem Controller und einer vorgegebenen Dummy-Klasse des entsprechend anderen Bereichs bereits lauffähig sind.

Die nötigen Informationen über die Implementierung von Schnittstellen in der gewählten Programmiersprache sind auf einem passenden allgemeinen Informationsblatt zu finden. Dieses ist aktuell nur für die Sprache Groovy vorhanden und müsste für andere Sprachen portiert werden.

Als weitere Aufgabe sollte die Modellierung im Klassendiagramm an die neuen Schnittstellen angepasst werden. Auch dazu gibt es ein passendes Informationsblatt.

Material

- ib_interfaces_groovy.pdf



Informatik *Interface in Groovy* *Informationsblatt*

Interface in Groovy

Ein Interface, zu deutsch Schnittstelle, ermöglicht es die Köpfe von Methoden zu definieren, ohne einen Methodenrumpf definieren zu müssen. Diese dürfen auch nicht definiert werden. Außerdem dürfen keine Attributen aufgeführt werden oder die Methoden als `protected` oder `private` deklariert werden.

In Groovy wird die Definition mit `interface` eingeleitet. Danach folgen wie bei einer Klassendefinition auch die Methoden. Die geschweiften Klammern für die Methodenrümpfe sind, wie oben bereits geschrieben, komplett wegzulassen.

```
interface Lampe {
    def einschalten()
    def ausschalten()
    def schalten(status)
    def gibStatus()
}
```

Um die Methoden mit einer Definition zu versehen muss eine Klasse geschrieben werden, die das Interface implementiert. Dieses funktioniert aber nur dann ohne Fehler, wenn alle im Interface aufgeführten Methoden auch in der Klasse definiert worden sind.

```
class Strahler implements Lampe {
    def status = "Aus"

    def einschalten(){
        this.status = "An"
    }

    def ausschalten(){
        this.status = "Aus"
    }

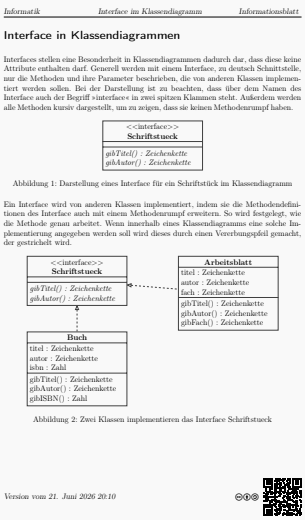
    def schalten(status){
        this.status = status
    }

    def gibStatus() {
        return this.status
    }
}
```

Eine Klasse kann dabei auch mehrere Interface gleichzeitig implementieren. Dazu müssen nur die Namen der Interface durch Komma getrennt hintereinander geschrieben werden. Auch kann die Klasse gleichzeit die Tochterklasse einer anderen Klasse sein.

Version vom 21. Juni 2026 20:10

- ib_interfaces_klassendiagramme.pdf



Informatik *Interface im Klassendiagramm* *Informationsblatt*

Interface in Klassendiagrammen

Interfaces stellen eine Besonderheit in Klassendiagrammen dadurch dar, dass diese keine Attribute enthalten darf. Generell werden mit einem Interface, zu deutsch Schnittstelle, nur die Methoden und ihre Parameter beschrieben, die von anderen Klassen implementiert werden sollen. Bei der Darstellung ist zu beachten, dass über dem Namen des Interface auch der Begriff »interface« in zwei spitzen Klammen steht. Außerdem werden alle Methoden kursiv dargestellt, um zu zeigen, dass sie keinen Methodenrumpf haben.

Abbildung 1: Darstellung eines Interface für ein Schriftstück im Klassendiagramm

Ein Interface wird von anderen Klassen implementiert, indem sie die Methodendefinitionen des Interface auch mit einem Methodenrumpf erweitern. So wird festgelegt, wie die Methode genau arbeitet. Wenn innerhalb eines Klassendiagramms eine solche Implementierung angegeben werden soll wird dieses durch einen Vererbungspfeil gemacht, der gestrichelt wird.

Abbildung 2: Zwei Klassen implementieren das Interface Schriftstueck

Version vom 21. Juni 2026 20:10

2.3 Implementierung mit Unterstützung durch abstrakte Klassen

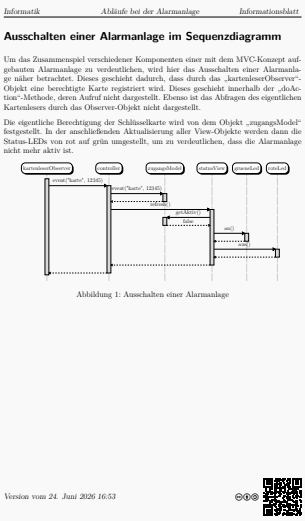
Bei der Implementierung sollen die Schülerinnen und Schüler arbeitsteilig vorgehen und Model-, View- und Observer-Klassen erstellen. Jede dieser Klassen kann durch den vorgegebenen Controller und die entsprechenden Dummy-Klassen direkt auch aktiv genutzt und getestet werden. Hierbei muss abgewogen werden, ob die Dateien mit Quellcode oder nur die kompilierten Klassen zur Verfügung gestellt werden.

Je nach Fertigkeiten der einzelnen Personen sollten die Ansprüche an die Klassen gewählt sein. Dabei können auch weitere bisher noch nicht in der Modellierung berücksichtigte Elemente genutzt werden. Dieses können z. B. das Ein- und Ausschalten mithilfe eines RFID-Lesers, Darstellung der Aktivitäten durch Lampen und Töne oder Ausgabe über eine Webseite.

Um das Verständnis wie die verschiedenen Elemente ineinandergreifen zu stärken, bietet sich das Informationsblatt an, dass die Abläufe konkret für die Alarmanlage visualisiert. Dabei wird auch das Zusammenspiel mit den LEDs und anderen Objekten verdeutlicht, die angeschlossene Hardwarekomponenten steuern.

Material

- ib_alarmanlage_ablaeufe.pdf



Informatik *Abläufe bei der Alarmanlage* *Informationsblatt*

Ausschalten einer Alarmanlage im Sequenzdiagramm

Um das Zusammenspiel verschiedener Komponenten einer mit dem MVC-Konzept aufgebauten Alarmanlage zu verdeutlichen, wird hier das Ausschalten einer Alarmanlage näher betrachtet. Dieses geschieht dadurch, dass durch das „kartenleserObserver"-Objekt eine berechtigte Karte registriert wird. Dieses geschieht innerhalb der „doAction"-Methode, deren Aufruf nicht dargestellt. Ebenso ist das Abfragen des eigentlichen Kartenlesers durch das Observer-Objekt nicht dargestellt.

Die eigentliche Berechtigung der Schlüsselkarte wird von dem Objekt „zugangsModel" festgestellt. In der anschließenden Aktualisierung aller View-Objekte werden dann die Status-LEDs von rot auf grün umgestellt, um zu verdeutlichen, dass die Alarmanlage nicht mehr aktiv ist.

Abbildung 1: Ausschalten einer Alarmanlage

Version vom 24. Juni 2026 16:53

Während der Entwicklung kann als weiteres Hilfsmittel die abstrakte Klasse eingeführt werden. Sie ist besonders dann nützlich, wenn verschiedene Model-, View- oder Observer-Klassen bei einigen Methoden die exakt gleichen Tätigkeiten vornehmen.

Material

- ib_abstrakte_klassen_groovy.pdf



Informatik *Abstrakte Klassen in Groovy* *Informationsblatt*

Abstrakte Klassen in Groovy

Durch abstrakte Klassen ist es möglich, eine gemeinsame Grundlage für andere Klassen zu erstellen und gleichzeitig dafür zu sorgen, dass von dieser Grundlage keine Objekte erzeugt werden können. Zusätzlich können diese abstrakten Klassen auch mit abstrakten Methoden versehen werden, bei denen nur der Kopf mit allen Parametern definiert wird.

In Groovy muss bei einer abstrakten Methode vor das `def` ein `abstract` geschrieben werden. Sobald dieses vor einer Methode steht, muss auch bei der Klasse vor dem `class` ein `abstract` stehen.

```
1 abstract class Auto {
2     def geschwindigkeit = 0
3
4     abstract def beschleunigen()
5
6     def bremsen() {
7         geschwindigkeit --
8     }
9 }
```

Jede Klasse die von einer abstrakten Klasse erbt muss entweder alle als abstrakt deklarierten Methoden definieren oder wieder als abstrakt deklariert werden. Das Erben von einer abstrakten Klasse geschieht dabei analog wie bei einer normalen Klasse.

```
1 class Elektrouato extends Auto {
2     def beschleunigen() {
3         geschwindigkeit ++
4     }
5 }
```

Version vom 21. Juni 2026 20:10

- ib_abstrakte_klassen.pdf

Abstrakte Klassen in Klassendiagrammen

Es gibt Methoden bei denen nur der Kopf definiert ist und sie keinen Rumpf haben. Solche Methoden werden als abstrakte Methoden bezeichnet. Jede Klasse die mindestens eine solche Methode hat ist dadurch eine abstrakte Klasse. In Klassendiagrammen wird für abstrakte Klassen über dem Klassennamen der Begriff «abstract» in zwei spitzen Klammen gesetzt. Teilweise findet man auch stattdessen geschweifte Klammern. Alle Methoden die abstrakt sind, werden kursiv geschrieben um sie von den normalen Methoden unterscheiden zu können.



Abbildung 1: Darstellung der abstrakten Klasse Auto im Klassendiagramm

Andere Klassen können von dieser Klasse abstammen und die abstrakten Methoden definieren. Erst wenn alle abstrakten Methoden definiert sind, handelt es sich bei der Kindklasse um eine normale Klasse. Sie müssen diese Methoden also implementieren. Die bereits in der abstrakten Klasse definierten Methoden erben sie, wie sie diese auch von einer normalen Elternklasse erben würden.

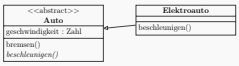


Abbildung 2: Eine Klassen erbt von der abstrakten Klasse

Version vom 21. Juni 2026 20:10