

Faktor-IPS

Schulung

Inhalt

1. Motivation
2. UML Refresh
3. Modellierung & Produktdefinition mit Faktor-IPS
4. Modellgetriebene Softwareentwicklung

Inhalt

1. Motivation
2. UML Refresh
3. Modellierung & Produktdefinition mit Faktor-IPS
4. Modellgetriebene Softwareentwicklung

Demo Faktor-IPS

- Webanwendung zur Erstellung eines Hausratangebots

Inhalt

1. Motivation
2. UML Refresh
3. Modellierung & Produktdefinition
4. Modellgetriebene Softwareentwicklung

UML Refresh

- Basics
 - Klassen
 - Attribute
 - Beziehungen
 - Instanzen
- Fortgeschrittene Modellierungskonzepte
 - Stereotypes
 - derived Attributes
 - derived Associations
 - Aggregation und Composition, AggregateRoot
 - Qualified Associations

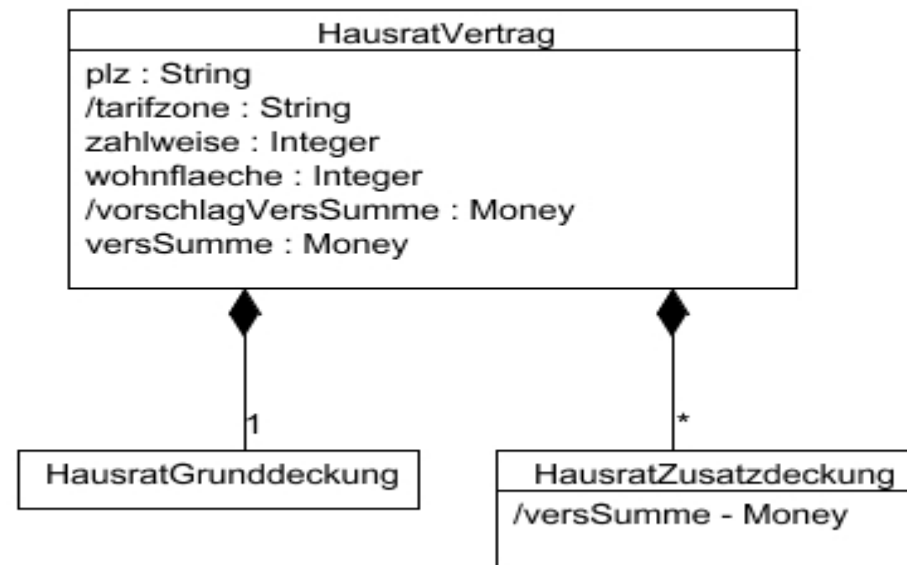
Inhalt

1. Motivation
2. UML Refresh
3. Modellierung & Produktdefinition
4. Modellgetriebene Softwareentwicklung

Inhalt: Modellierung & Produktdefinition

1. Projekt einrichten, erste Klasse anlegen
2. Modellierung der Vertragsseite
3. Modellierung der Produktseite
4. Verwendung von Tabellen
5. Flexible Modelle / Verwendung von Formeln
6. Plausibilisierung
7. Vererbung / Derived union Beziehungen
8. Testunterstützung

Schulungsbeispiel: Hausratmodell



Demo: Projekt einrichten, erste Klasse anlegen

- Java-Projekt „Hausratmodell“ anlegen
- Faktor-IPS Nature hinzufügen
- IPS-Package „hausrat“ anlegen
- Klasse „HausratVertrag“ anlegen
- Generierten Sourcecode erläutern
- Faktor-IPS Modellexplorer erläutern

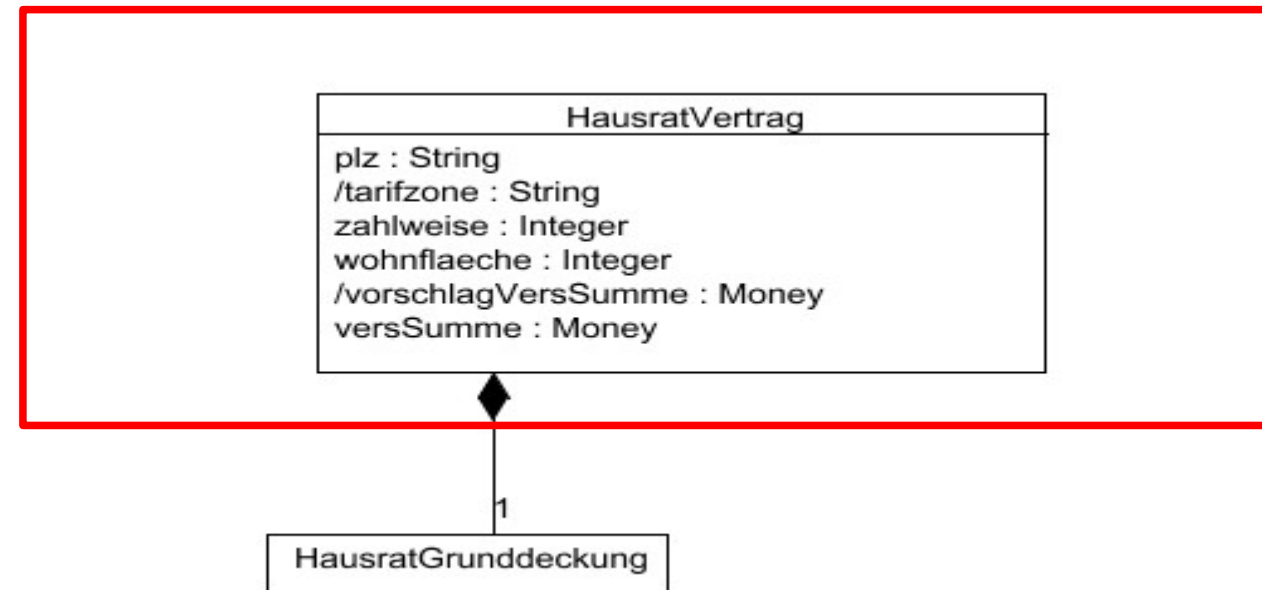
Übungen zu Kapitel 1:

- analog zur Demo

Inhalt: Modellierung & Produktdefinition

1. Projekt einrichten, erste Klasse anlegen
2. Modellierung der Vertragsseite
 1. Attribute
 2. Beziehungen
3. Modellierung der Produktseite
4. Verwendung von Tabellen
5. Flexible Modelle / Verwendung von Formeln
6. Plausibilisierung
7. Vererbung / Derived union Beziehungen
8. Testunterstützung

Demo: Modellierung von Attributen

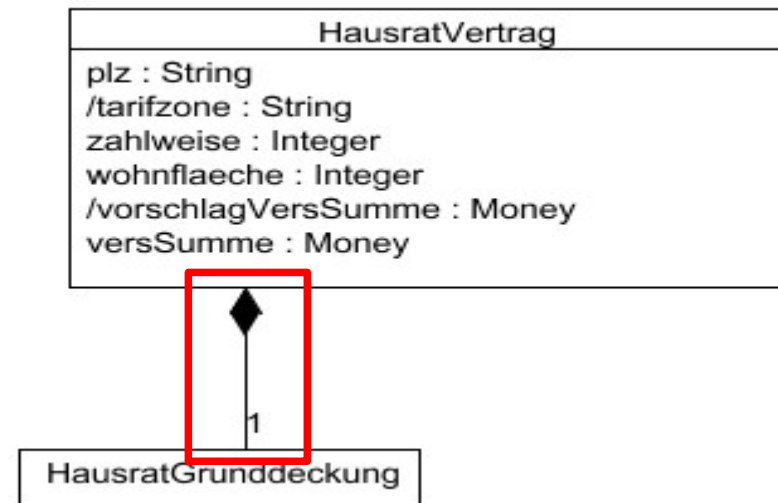


Übung zu Kapitel 2-1

Inhalt: Modellierung & Produktdefinition

1. Projekt einrichten, erste Klasse anlegen
2. Modellierung der Vertragsseite
 1. Attribute
 2. Beziehungen
3. Modellierung der Produktseite
4. Verwendung von Tabellen
5. Flexible Modelle / Verwendung von Formeln
6. Plausibilisierung
7. Vererbung / Derived union Beziehungen
8. Testunterstützung

Demo: Modellierung von Beziehungen



Übungen zu Kapitel 2-2

Inhalt: Modellierung & Produktdefinition

1. Projekt einrichten, erste Klasse anlegen
2. Modellierung der Vertragsseite
3. Modellierung der Produktseite
 1. Grundlagen & inkl. Produktattribute
 2. Produktänderungen im Zeitablauf
 3. Konfigurierbare Vertragsattribute
 4. Beziehungen
 5. Zugriff auf Produktdaten zur Laufzeit
4. Verwendung von Tabellen
5. Flexible Modelle / Verwendung von Formeln
6. Plausibilisierung
7. Vererbung / Derived union Beziehungen
8. Testunterstützung

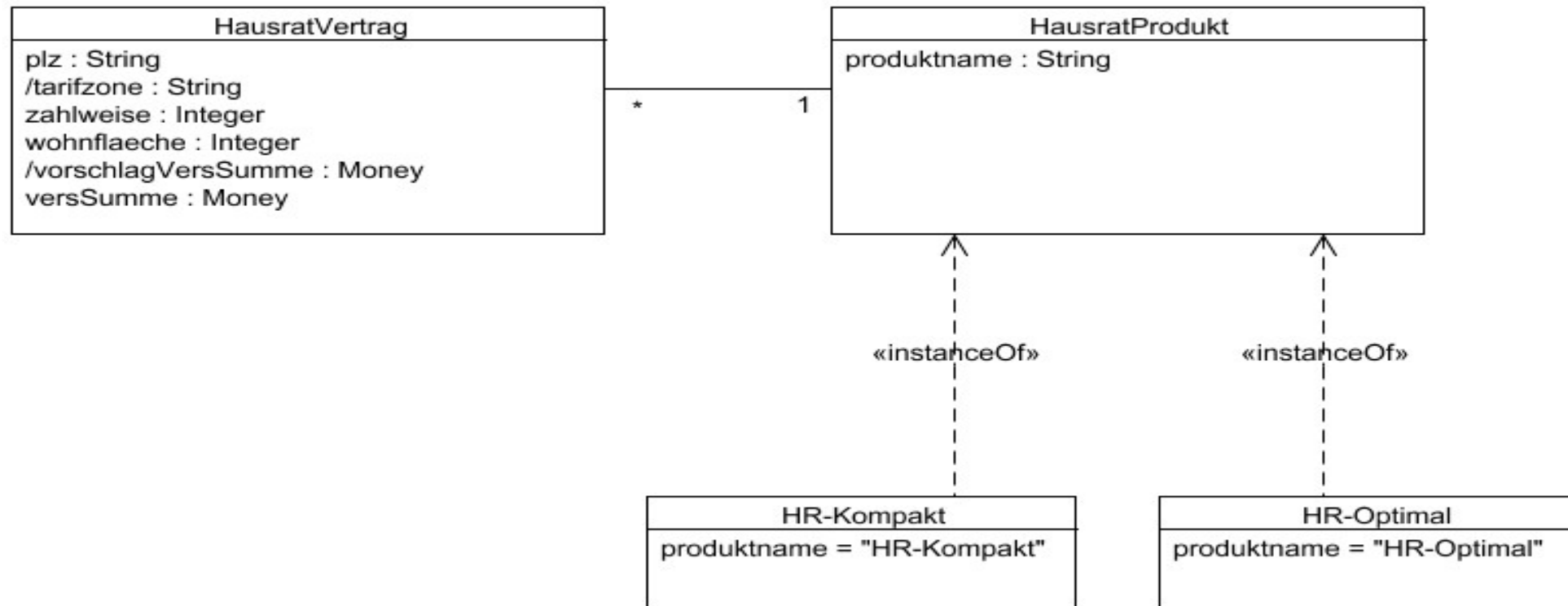
Vor der Modellierung ...

- Projekt „Hausratprodukte“ für die Produktdaten anlegen

Motivation

- Es gibt zwei Hausratprodukte:
 - HR-Kompakt: Günstiger Basisschutz
 - HR-Optimal: Optimaler Schutz
- Jeder HausratVertrag wird entweder auf Basis von HR-Kompakt oder HR-Optimal abgeschlossen.

Abbildung der Hausratprodukte im Modell



Demo: Anlegen der Hausratprodukte

- Produktklasse „HausratProdukt“ anlegen
- Attribut „produktname“ definieren.
- Produkte HR-Kompakt & HR-Optimal anlegen

Übungen zu Kapitel 3-1

- analog zur Demo

Inhalt: Modellierung & Produktdefinition

1. Projekt einrichten, erste Klasse anlegen
2. Modellierung der Vertragsseite
3. Modellierung der Produktseite
 1. Grundlagen inkl. Produktattribute
 2. Produktänderungen im Zeitablauf
 3. Konfigurierbare Vertragsattribute
 4. Beziehungen
 5. Zugriff auf Produktdaten zur Laufzeit
4. Verwendung von Tabellen
5. Flexible Modelle / Verwendung von Formeln
6. Plausibilisierung
7. Vererbung / Derived union Beziehungen
8. Testunterstützung

Produktänderungen im Zeitablauf

- Theorie: Anderer Foliensatz

Modell unter Berücksichtigung zeitl. Änderungen

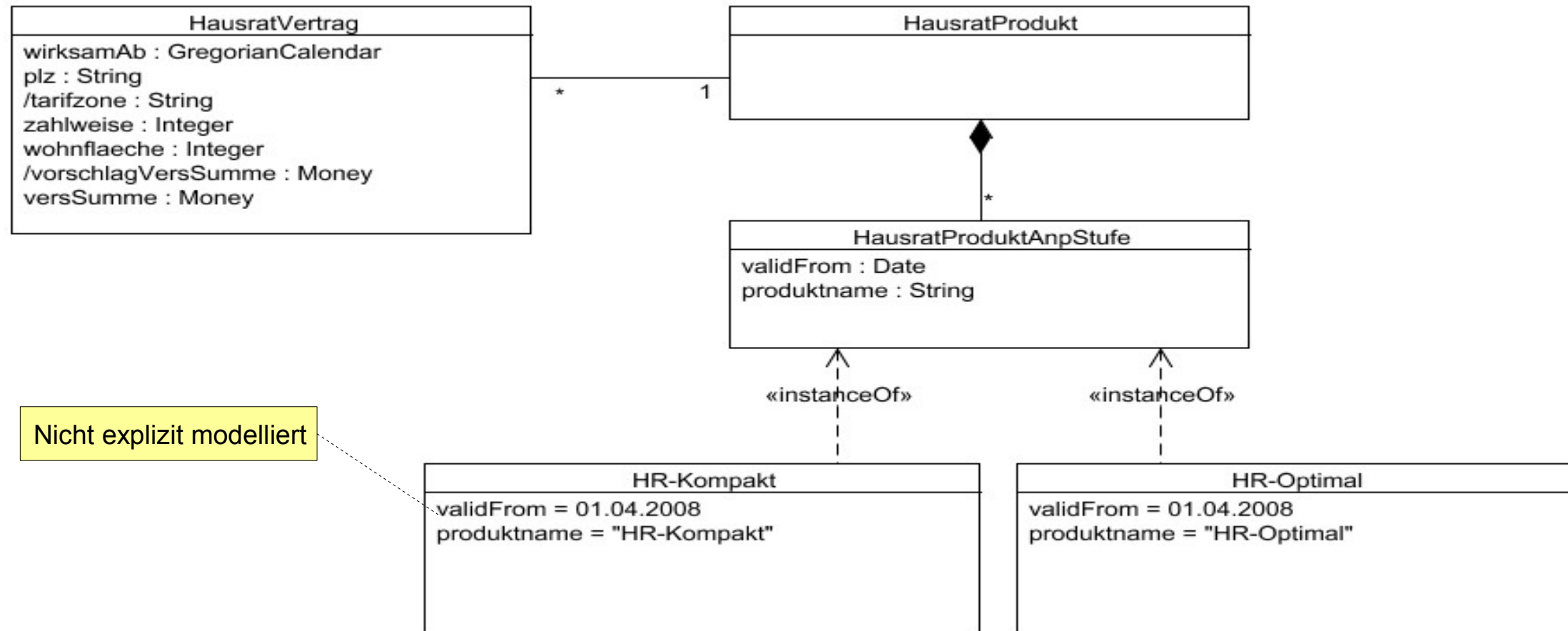


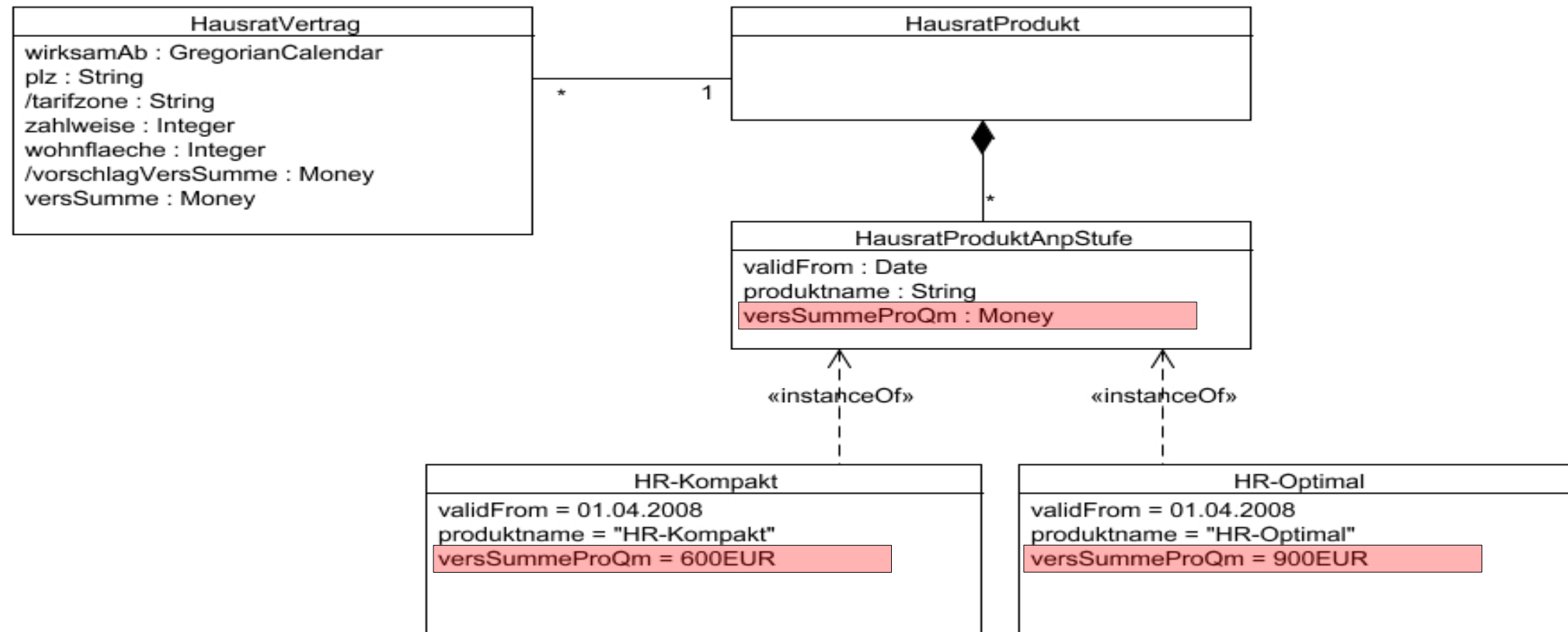
Abbildung zeitlicher Änderungen im Sourcecode

- Analyse generierter Sourcecode

Beispiel: Berechnung VorschlagVersSumme

- Bisher:
 - $\text{vorschlagVersSumme} = \text{wohnflaeche} * 650 \text{ Euro.}$
- Soll:
 - HR-Kompakt: $\text{vorschlagVersSumme} = \text{wohnflaeche} * 600$
 - HR-Optimal: $\text{vorschlagVersSumme} = \text{wohnflaeche} * 900$

Beispiel: Berechnung VorschlagVersSumme



Demo: Produktänderungen im Zeitablauf

- Hinzufügen des Attributes `wirksamAb` im Vertrag. Implementierung der Methode `getEffectiveFromAsCalendar()`.
- Definition des Attributes „`vorschlagVersSummeProQm`“ und Implementierung der Berechnung des Vorschlags für die Versicherungssumme.

Übung: Produktänderungen im Zeitablauf

- Analog zur Demo

Inhalt: Modellierung & Produktdefinition

1. Projekt einrichten, erste Klasse anlegen
2. Modellierung der Vertragsseite
3. Modellierung der Produktseite
 1. Grundlagen inkl. Produktattribute
 2. Produktänderungen im Zeitablauf
 3. Konfigurierbare Vertragsattribute
 4. Beziehungen
 5. Zugriff auf Produktdaten zur Laufzeit
4. Verwendung von Tabellen
5. Flexible Modelle / Verwendung von Formeln
6. Plausibilisierung
7. Vererbung / Derived union Beziehungen
8. Testunterstützung

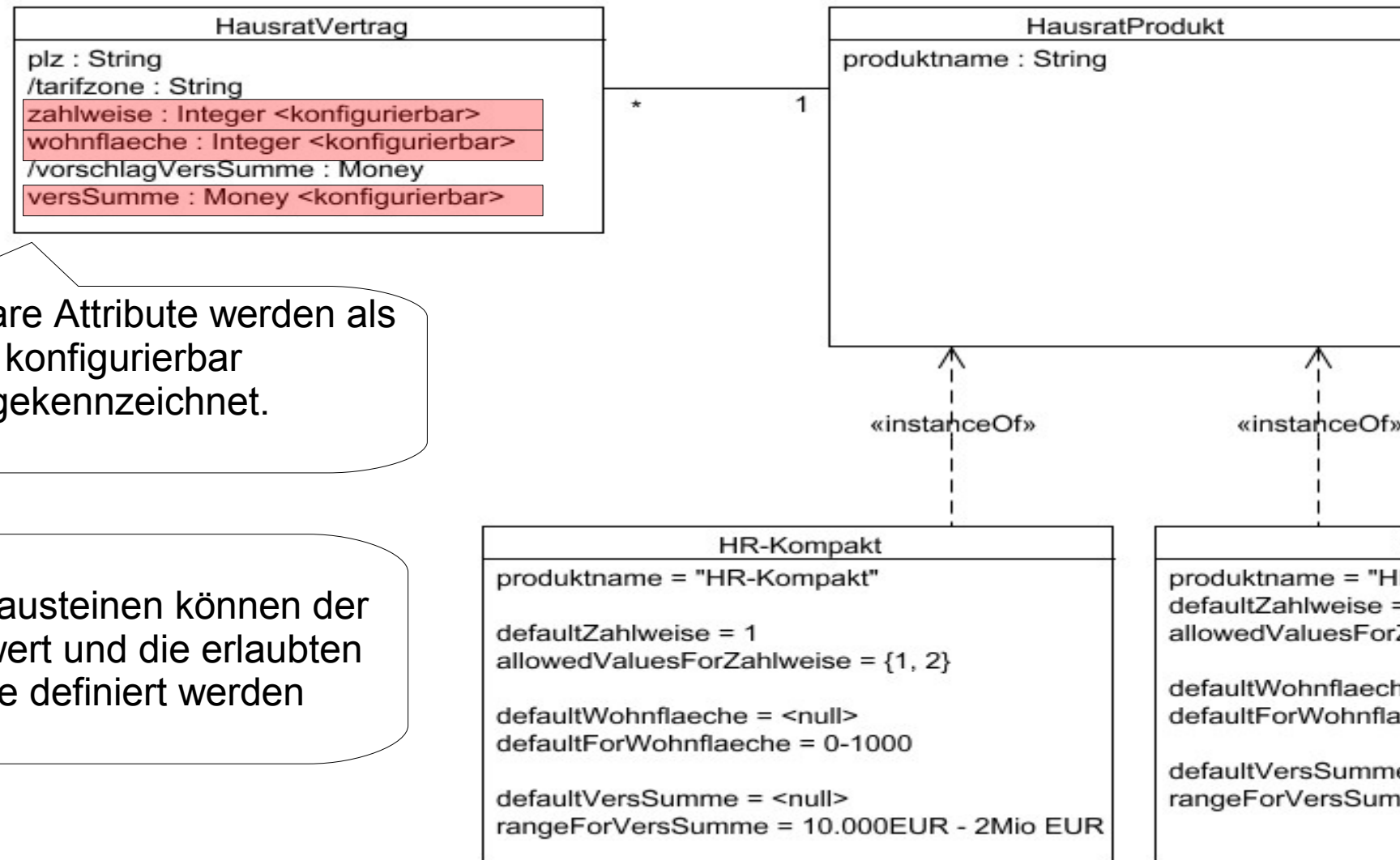
Konfigurationsmöglichkeiten des Hausratvertrags

Änderbare Eigenschaft des Hausratvertrags	Konfigurationsmöglichkeiten
zahlweise	Die im Vertrag erlaubten Zahlungsweisen. Der Defaultwert für die Zahlungsweise bei Erzeugung eines neuen Vertrags.
wohnflaeche	Bereich (min, max), in dem die Wohnfläche liegen muss. Der Defaultwert für die Wohnfläche bei Erzeugung eines neuen Vertrags.
versSumme	Bereich, in dem die Versicherungssumme liegen muss. Der Defaultwert für die Versicherungssumme bei Erzeugung eines neuen Vertrags.

Beispielprodukte: HR-Kompakt & HR-Optimal

Konfigurationsmöglichkeit	HR-Kompakt	HR-Optimal
Defaultwert Zahlweise	jährlich	jährlich
Erlaubte Zahlweisen	halbjährlich, jährlich	monatlich, vierteljährlich, halbjährlich, jährlich
Defaultwert Wohnfläche	<null>	<null>
Erlaubter Bereich Wohnfläche	0-1000 qm	0-2000 qm
Defaultwert VersSumme	<null>	<null>
Erlaubter Bereich VersSumme	10Tsd – 2Mio Euro	10Tsd – 5Mio Euro

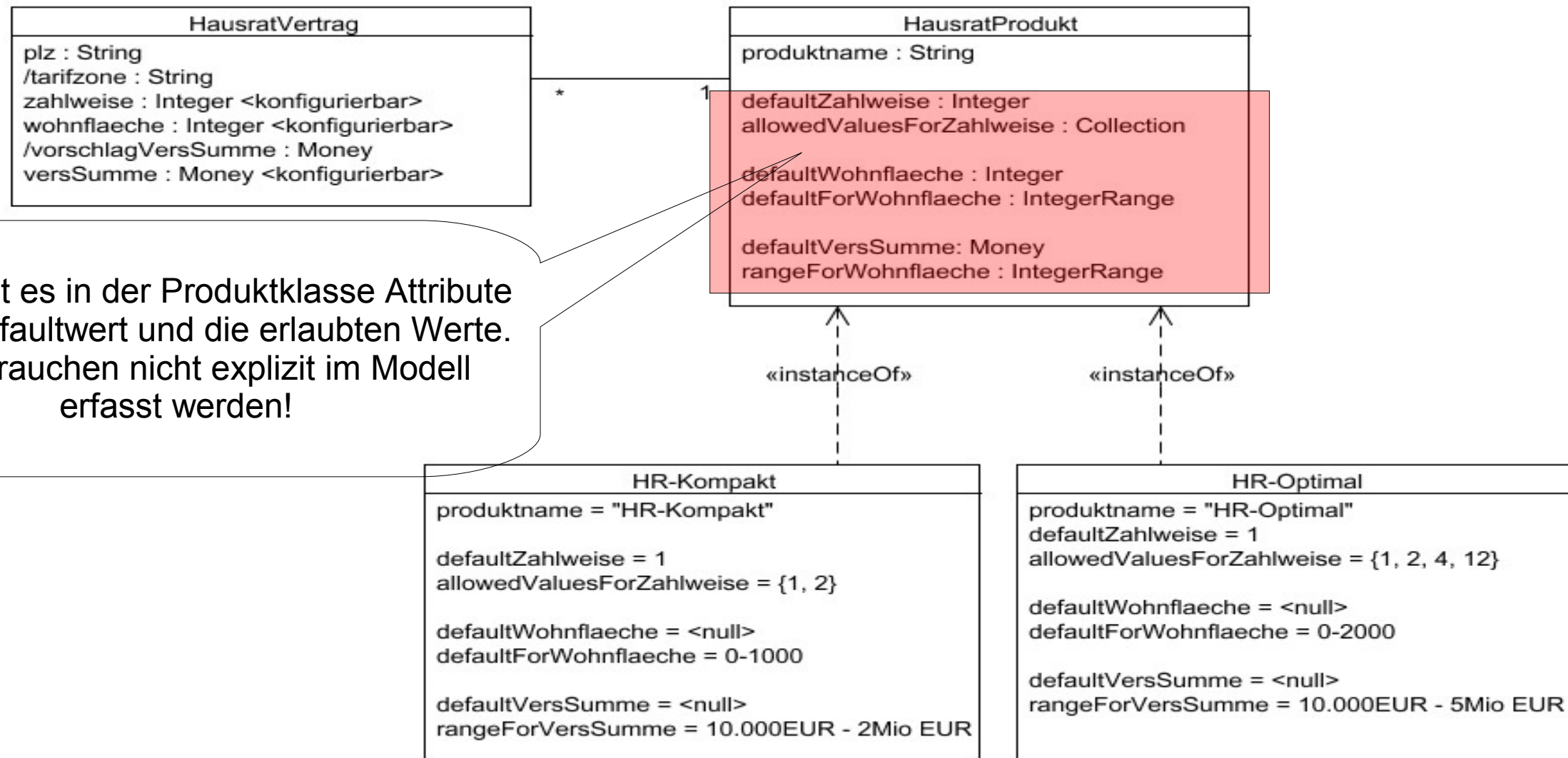
Abbildung im Modell (ohne zeitliche Änderungen)



Änderbare Attribute werden als konfigurierbar gekennzeichnet.

In den Bausteinen können der Defaultwert und die erlaubten Werte definiert werden

Abbildung im Modell



Demo: Modellerweiterung in Faktor-IPS

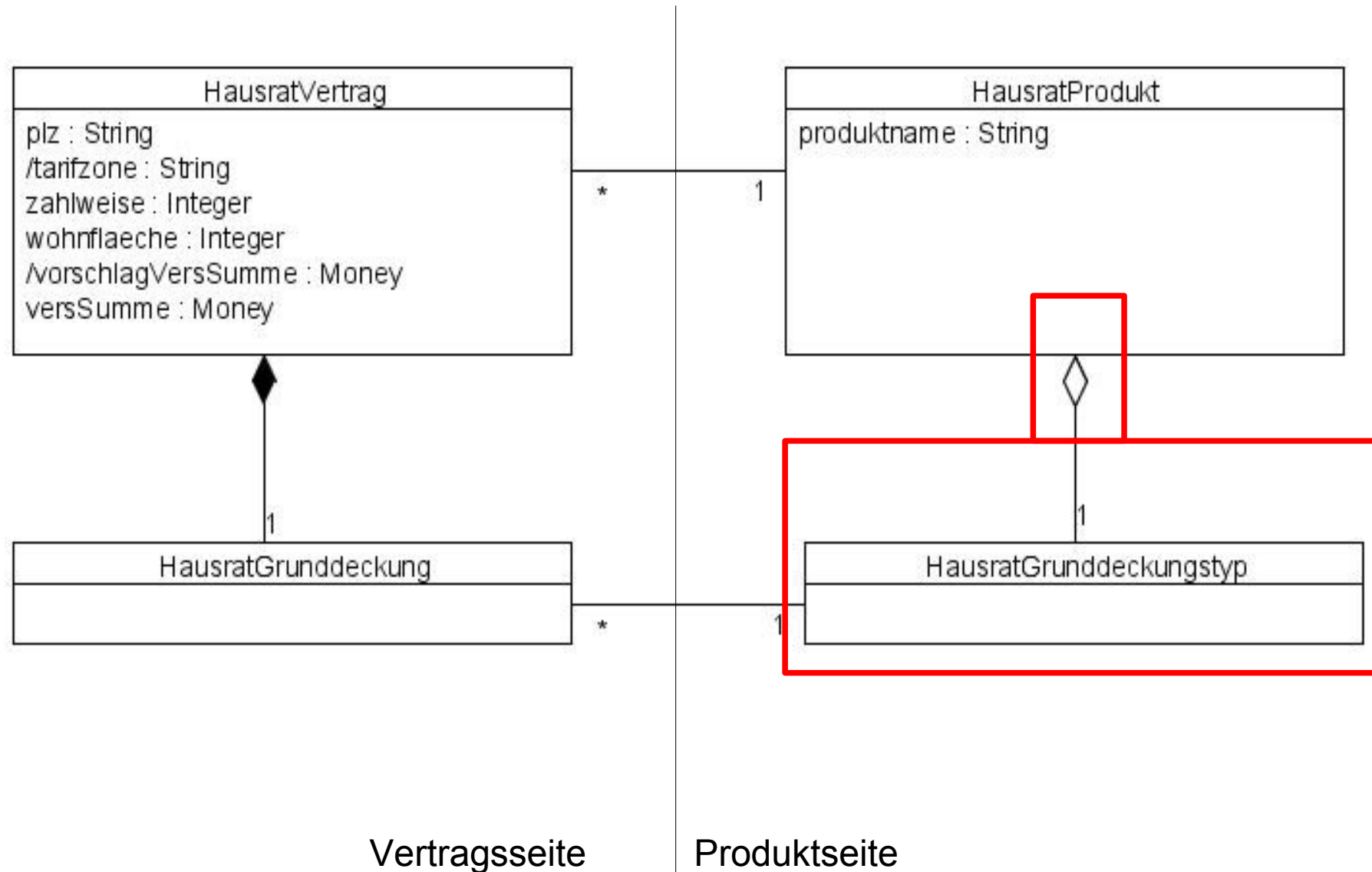
- Markieren des Attributes „zahlweise“ als konfigurierbar.
- Hinzufügen der Defaultzahlweise und der möglichen Zahlweisen für die beiden Produkte
- Analyse des Sourcecodes
- Markieren des Attributes „wohnflaeche“ als konfigurierbar.
- Definition des erlaubten Bereichs für die Wohnfläche in den beiden Produkten
- Analyse des Sourcecodes

Übungen zu Kapitel 3-3

Inhalt: Modellierung & Produktdefinition

1. Projekt einrichten, erste Klasse anlegen
2. Modellierung der Vertragsseite
3. Modellierung der Produktseite
 1. Grundlagen inkl. Produktattribute
 2. Produktänderungen im Zeitablauf
 3. Konfigurierbare Vertragsattribute
4. Beziehungen
5. Zugriff auf Produktdaten zur Laufzeit
4. Verwendung von Tabellen
5. Flexible Modelle / Verwendung von Formeln
6. Plausibilisierung
7. Vererbung / Derived union Beziehungen
8. Testunterstützung

Beziehungen auf Produktseite



Übung: Beziehungen auf Produktseite

Inhalt: Modellierung & Produktdefinition

1. Projekt einrichten, erste Klasse anlegen
2. Modellierung der Vertragsseite
3. Modellierung der Produktseite
 1. Grundlagen inkl. Produktattribute
 2. Produktänderungen im Zeitablauf
 3. Konfigurierbare Vertragsattribute
 4. Beziehungen
 5. Zugriff auf Produktdaten zur Laufzeit
4. Verwendung von Tabellen
5. Flexible Modelle / Verwendung von Formeln
6. Plausibilisierung
7. Vererbung / Derived union Beziehungen
8. Testunterstützung

Demo: Zugriff auf Informationen zur Laufzeit

- XML-Files im „derived“ Verzeichnis
- Definition des toc-files im „.ipsproject“ ansehen
- Code im Testfall erläutern
- Testfall ausführen

Übung: Zugriff auf Informationen zur Laufzeit

- analog zur Demo

Inhalt: Modellierung & Produktdefinition

1. Projekt einrichten, erste Klasse anlegen
2. Modellierung der Vertragsseite
3. Modellierung der Produktseite
4. Verwendung von Tabellen

1. Grundlagen

2. Beziehungen zwischen Produktbausteinen und Tabellen
5. Flexible Modelle / Verwendung von Formeln
6. Plausibilisierung
7. Vererbung / Derived union Beziehungen
8. Testunterstützung

Beispiel Tarifzonentabelle

<i>Plz-Von</i>	<i>Plz-bis</i>	<i>Tarifzone</i>
17235	17237	II
45525	45549	III
59174	59199	IV
47051	47279	V
63065	63075	VI

Demo: Einführung Tarifzonentabelle

Übung: Einführung Tarifzonentabelle

- Analog zur Demo

Inhalt: Modellierung & Produktdefinition

1. Projekt einrichten, erste Klasse anlegen
2. Modellierung der Vertragsseite
3. Modellierung der Produktseite
4. Verwendung von Tabellen
 1. Grundlagen
 2. Beziehungen zwischen Produktbausteinen und Tabellen
5. Flexible Modelle / Verwendung von Formeln
6. Plausibilisierung
7. Vererbung / Derived union Beziehungen
8. Testunterstützung

Beispiel: Tariftabelle für die Grunddeckungen der Hausratprodukte

Produkt	Tarifzone	Beitragssatz
HR-Optimal	I	0.8
HR-Optimal	II	1.0
HR-Optimal	III	1.44
HR-Optimal	IV	1.70
HR-Optimal	V	2.00
HR-Optimal	VI	2.20
HR-Kompakt	I	0.6
HR-Kompakt	II	0.8
HR-Kompakt	III	1.21
HR-Kompakt	IV	1.50
HR-Kompakt	V	1.80
HR-Kompakt	VI	2.00

Beispiel: Trennung der Tabelle nach Produkt

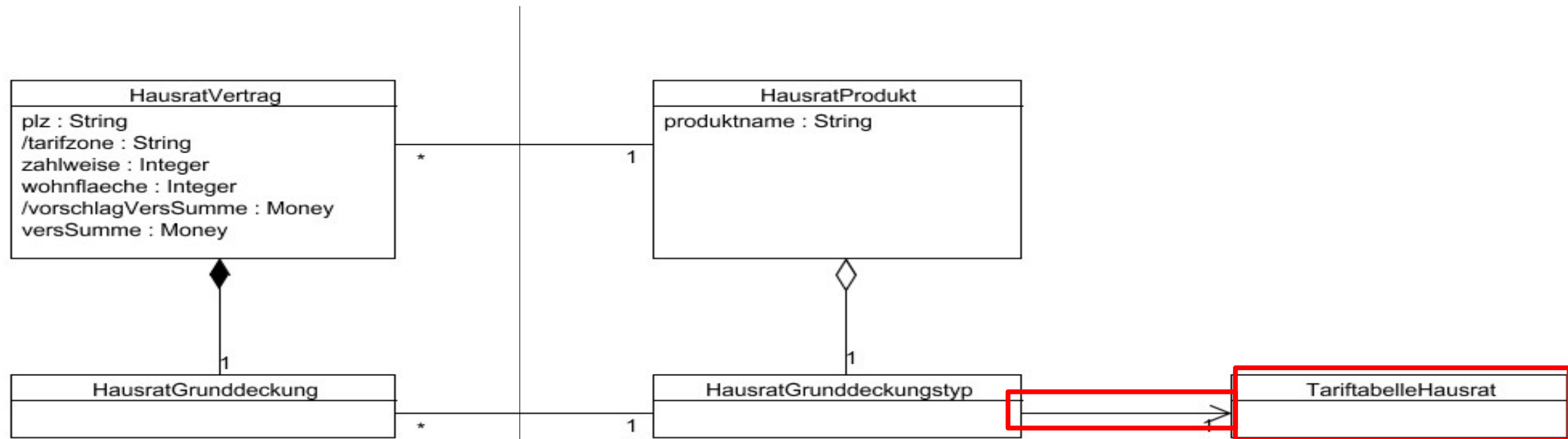
Tabelle für Grunddeckung
von HR-Optimal

<i>Tarifzone</i>	<i>Beitragssatz</i>
I	0.8
II	1.0
III	1.44
IV	1.70
V	2.00
VI	2.20

Tabelle für Grunddeckung
HR-Kompakt

<i>Tarifzone</i>	<i>Beitragssatz</i>
I	0.6
II	0.8
III	1.21
IV	1.50
V	1.80
VI	2.00

Abbildung im Modell

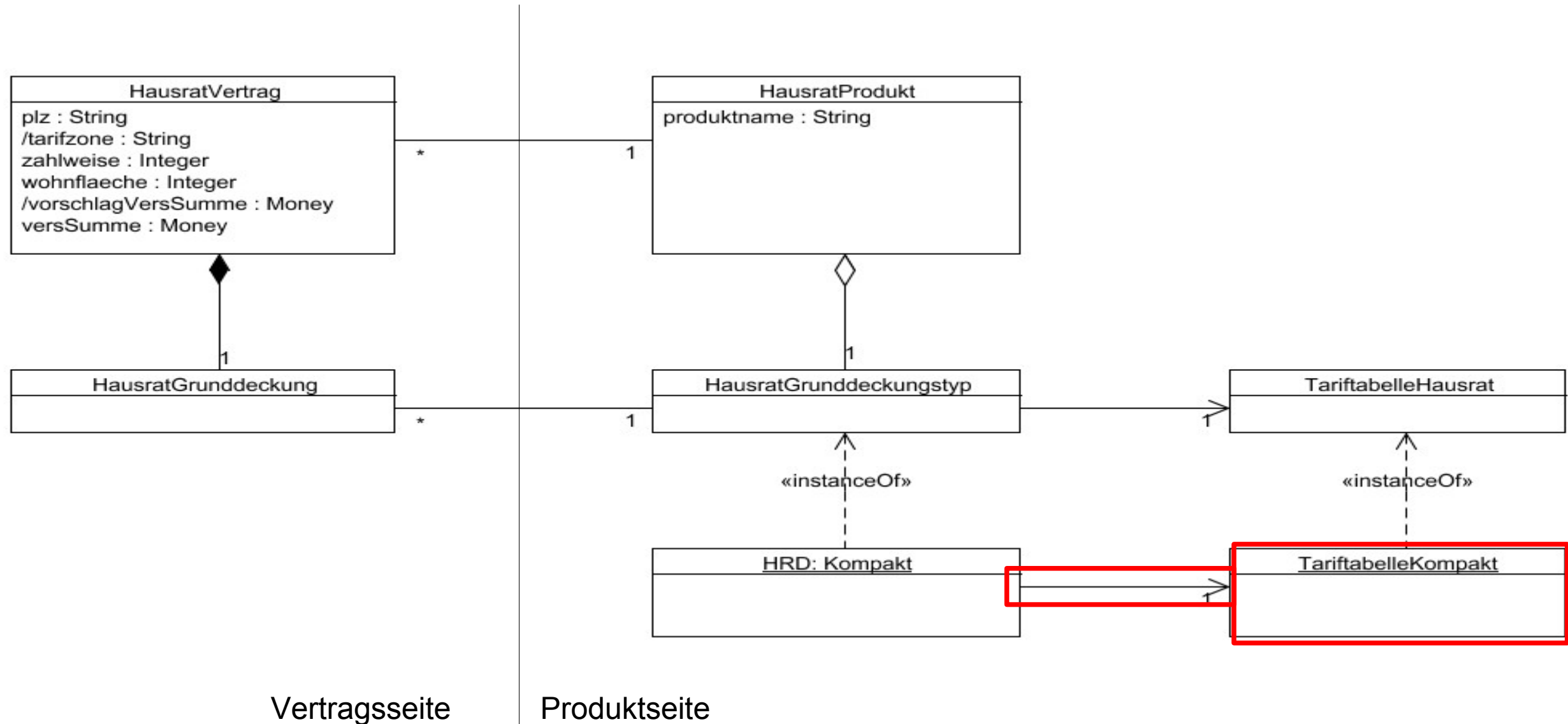


Ein Hausratgrunddeckungstyp verwendet eine Hausrattariftabelle
(zur Beitragsberechnung)

Vertragsseite

Produktseite

Abbildung im Modell inklusive Instanzen



Beitragsberechnung für die Grunddeckungen

- Berechnungsvorschrift
 - Ermittlung des Beitragsatzes pro 1000 Euro Versicherungssumme aus der Tariftabelle
 - Division der Versicherungssumme durch 1000 Euro und Multiplikation mit dem Beitragssatz
- Implementierung in der Übung

Demo: Tariftabellen für die Hausratprodukte

- Anlegen des derived (cached) Attributes *jahresbasisbeitrag* in der Modellklasse Hausratgrunddeckung
- Definition der Methode *berechneJahresbasisbeitrag()* in der Modellklasse Hausratgrunddeckung
- Implementierung der Methode *berechneJahresbasisbeitrag()* in der Java Klasse Hausratgrunddeckung
- Testfall für die Methode implementieren

Übung: Tariftabellen für die Hausratprodukte

- Analog Demo

Inhalt

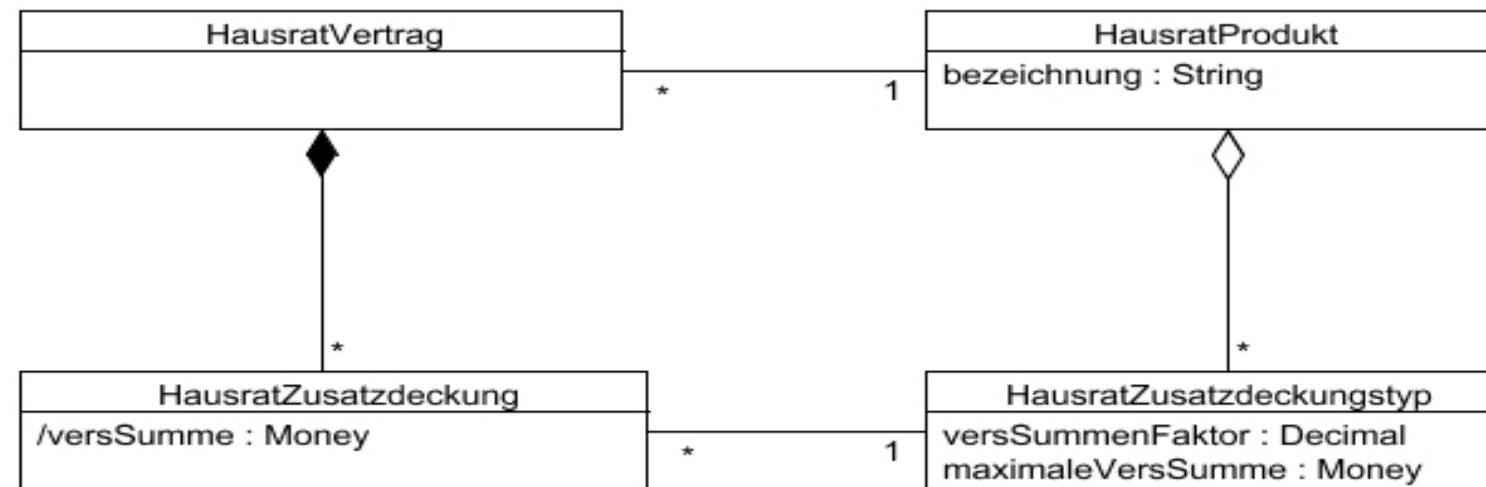
1. Projekt einrichten, erste Klasse anlegen
2. Modellierung der Vertragsseite
3. Modellierung der Produktseite
4. Verwendung von Tabelle
5. Flexible Modelle / Verwendung von Formeln
6. Plausibilisierung
7. Vererbung / Derived union Beziehungen
8. Testunterstützung

Fachliche Anforderungen

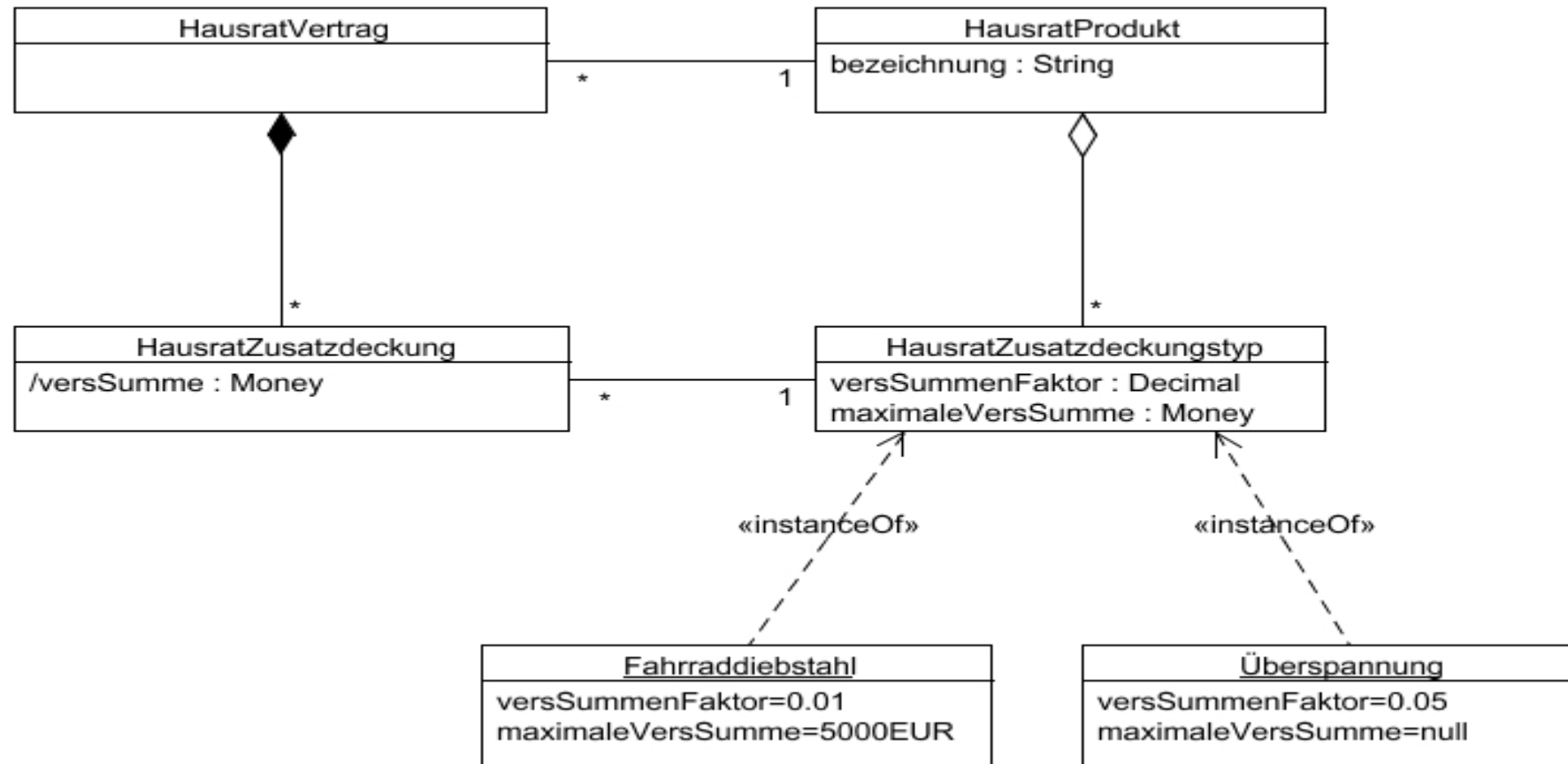
- Erweiterung des Modells, so dass Zusatzdeckungen durch die Fachabteilung hinzugefügt werden können, ohne dass das Modell geändert werden muss.
- Jede Zusatzdeckung verfügt über eine eigene Versicherungssumme und einen eigenen Jahresbasisbeitrag. Die Versicherungssumme ergibt sich aus der im Vertrag vereinbarten Summe.
- Beispiele:

	<i>Fahrraddiebstahl</i>	<i>Überspannung</i>
Versicherungssumme der Zusatzdeckung	1% der im Vertrag vereinbarten Summe, maximal 5000 Euro.	5% der im Vertrag vereinbarten Summe. Keine Deckelung.
Jahresbasisbeitrag	10% der Versicherungssumme der Fahrraddiebstahldeckung	10Euro + 3% der Versicherungssumme der Überspannungsdeckung

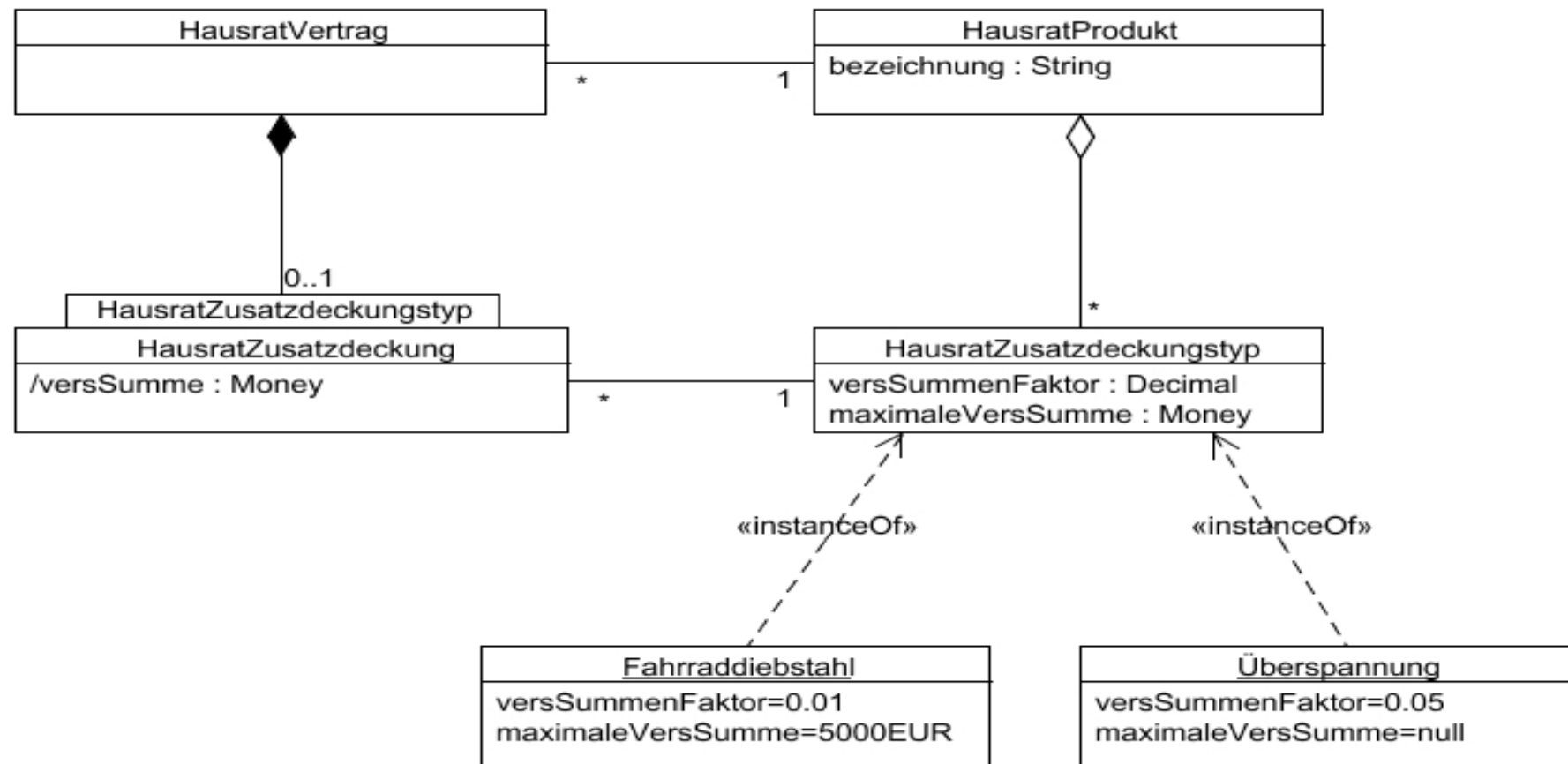
Modell der Zusatzdeckungen



Modell der Zusatzdeckungen mit Instanzen



Modell der Zusatzdeckungen mit Qualifier



Demo: Anlegen der Zusatzdeckungen

	HRD-Fahrraddiebstahl 2008-04	HRD-Überspannung 2008-04
Bezeichnung	Fahrraddiebstahl	Überspannungsschutz
VersSummenFaktor	0.01	0.05
MaximaleVersSumme	3000EUR	<null>

Demo: Beitragsberechnung für die Zusatzdeckungen

- Basisbeiträge
 - Fahrraddiebstahl 10% der Versicherungssumme
 - Überspannung 10EUR + 3% der Versicherungssumme

Übungen

Inhalt

1. Projekt einrichten, erste Klasse anlegen
2. Modellierung der Vertragsseite
3. Modellierung der Produktseite
4. Verwendung von Tabelle
5. Flexible Modelle / Verwendung von Formeln
6. Plausibilisierung
7. Vererbung / Derived union Beziehungen
8. Testunterstützung

Plausibilisierungen

- Aspekte einer Plausibilisierung
 - Auskunft
 - Prüfung
- Arten von Plausibilisierung
 - Wertebereich von Attributen
 - Struktur (Menge der erlaubten Links einer Beziehung)

Framework für Plausibilisierung in Faktor-IPS

- executeRule...() Methoden
- validate(), validateSelf(), validateDependents Methoden
- MessageList, Message
 - ObjectProperty
 - Message Code
 - MsgReplacementParameters
- getAllowedValuesFor...() Methoden
- getRangeFor...() Methoden
- getCardinalityFor...() Methoden

Übung

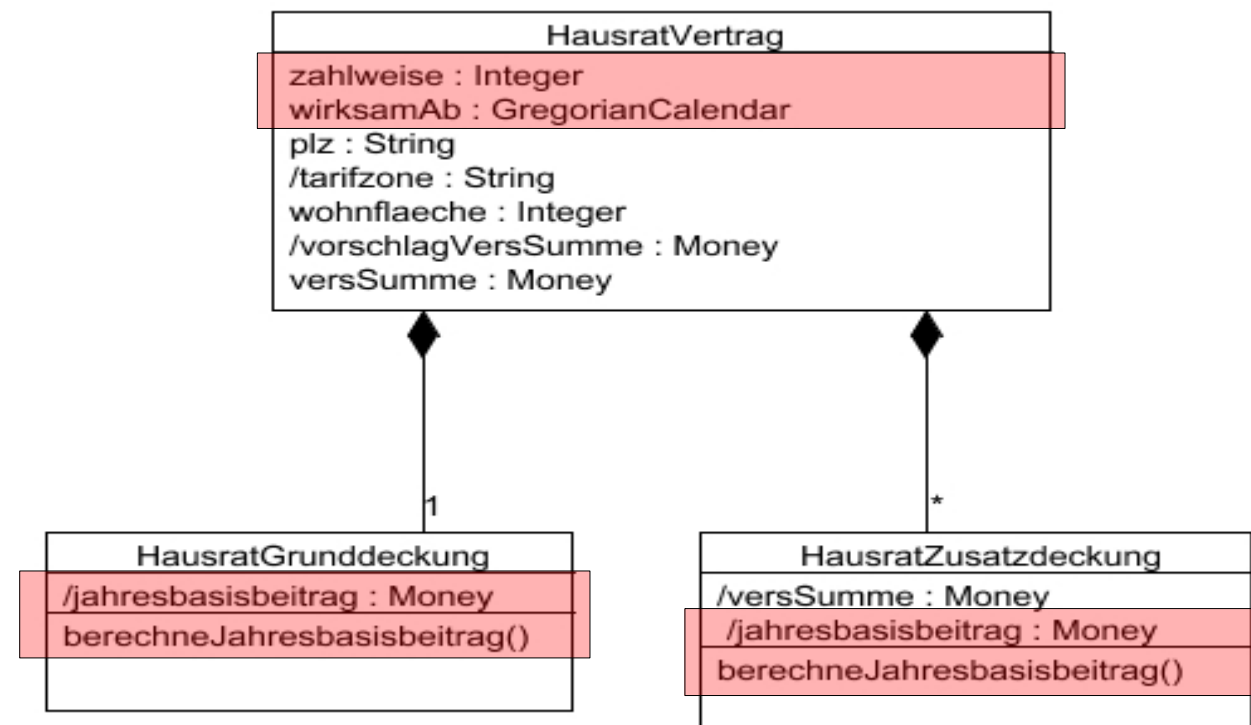
Plausibilisierungen – weiterführende Themen

- Abbildung komplexer Wertebereichsregeln
 - Verwendung von Tabellen
- Abbildung komplexer Strukturregeln
 - Verwendung von Beziehungen
 - Verwendung von Tabellen

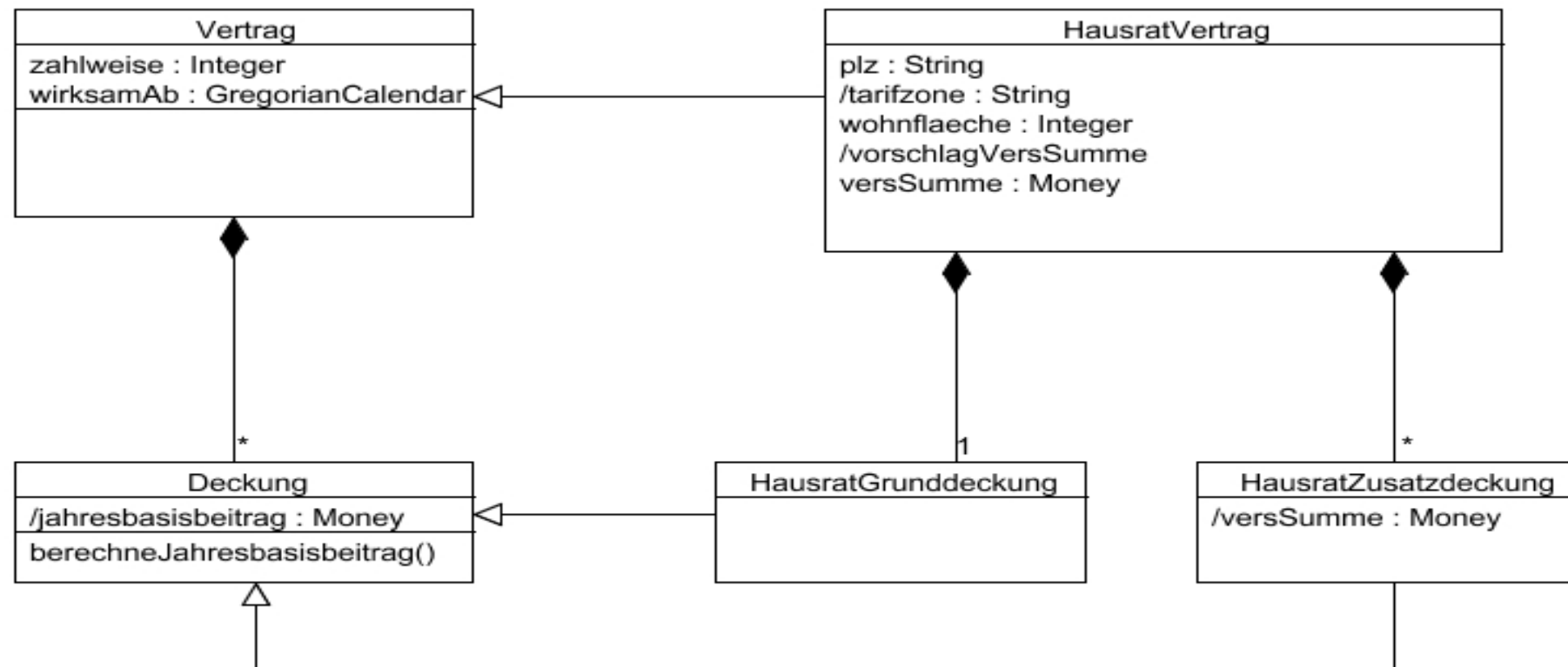
Inhalt

1. Projekt einrichten, erste Klasse anlegen
2. Modellierung der Vertragsseite
3. Modellierung der Produktseite
4. Verwendung von Tabellen
 1. Grundlagen
 2. Beziehungen zwischen Produktbausteinen und Tabellen
5. Flexible Modelle / Verwendung von Formeln
6. Plausibilisierung
7. Vererbung / Derived union Beziehungen
8. Testunterstützung

Aktuelles Modell



Modell mit spartenübergreifenden Basisklassen



Demo: Einführung der Basisklassen

Erwartete Semantik des Modells

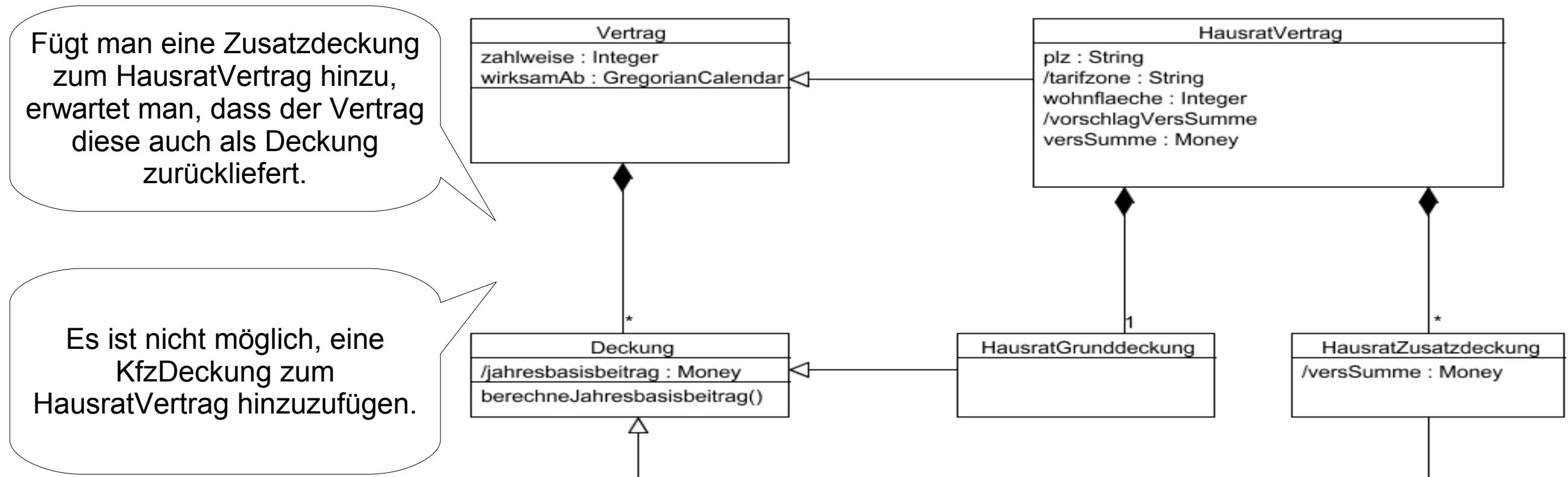
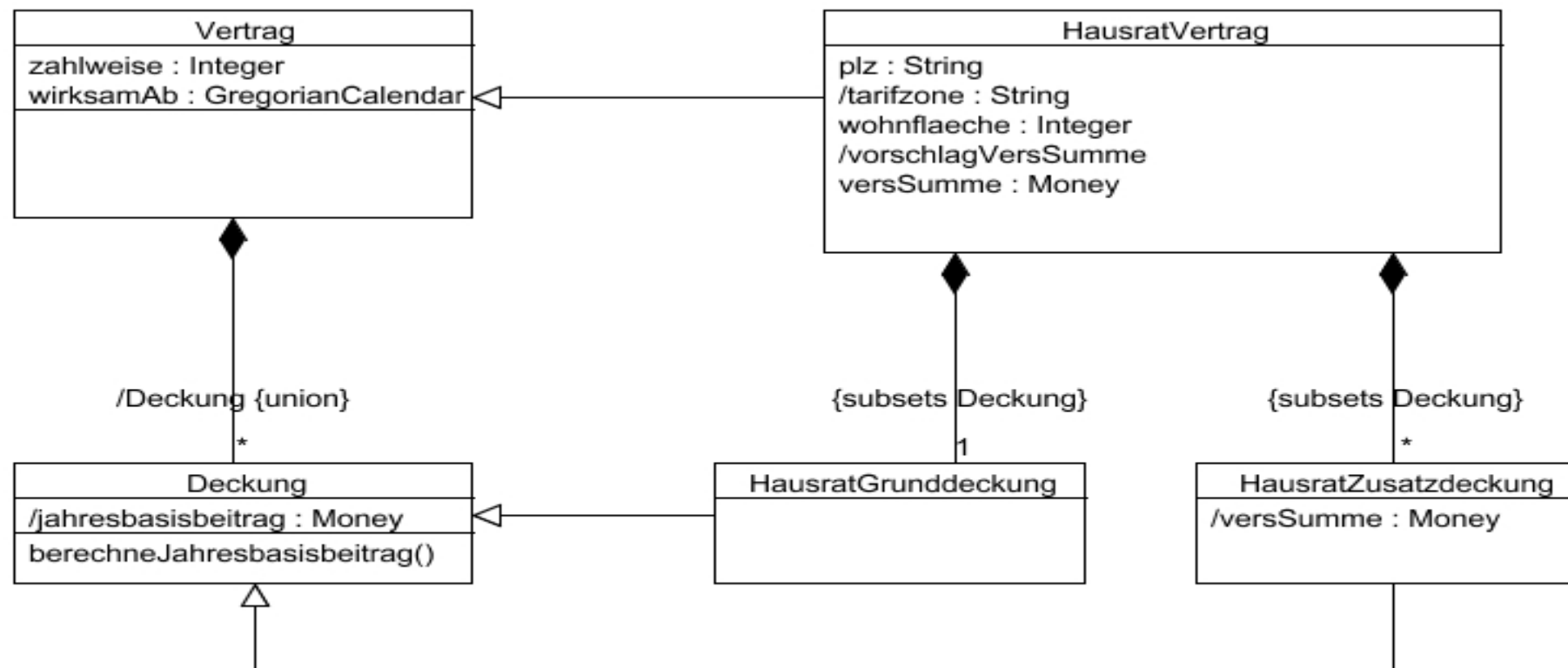


Abbildung der Semantik durch derived unions



Demo: derived union

Inhalt

1. Projekt einrichten, erste Klasse anlegen
2. Modellierung der Vertragsseite
3. Modellierung der Produktseite
4. Verwendung von Tabellen
5. Flexible Modelle / Verwendung von Formeln
6. Plausibilisierung
7. Vererbung / Derived union Beziehungen
8. Testunterstützung
 - 1. Grundlagen
 - 2. Unit Testing ohne produktive Produktdaten
 - 3. Fachliche Tests mit dem FaktorIPS Testwerkzeug

Testen

- Unit Testing ohne produktive Produktdaten
 - Verwendung von JUnit für Modultests / einzelne Funktionen
 - Testfälle können unabhängig von produktiven Produktdaten sein
 - Isoliertes Testen möglich, Testfälle verwenden nicht die gleichen Testdaten
 - Testen von Spezialfällen möglich (die möglicherweise durch produktive Produktdaten nicht abgebildet werden)
- Fachliche Tests mit dem Faktor-IPS Testwerkzeug
 - Integrationstest von Produktdaten & Modell/Sourcecode
 - Fachliche Tests können vom Fachbereich erstellt werden
- Testfälle und die benötigten Testdaten bilden ein Einheit, werden zusammen mit dem Sourcecode im KM-Tool verwaltet.

Demo: Unit Testing ohne produktive Produktdaten

- Aufbauen des Inhalts des InMemoryRepositories
- Ableitung der Klasse HausratGrunddeckungstypAnpStufe
 - Zugriff auf das protected Attribut tariftabelleName
- Ableitung der Tabellen
 - Initialisierung des rows Attribut
 - Aufruf der initKeyMaps() Method
- Testmethode für die Berechnung des Grunddeckungsbeitrags auf den Inhalt des InMemoryRepositories erstellen

Übung: Unit Testing ohne produktive Produktdaten

Demo: Fachliche Tests mit dem FaktorIPS Testwerkzeug

- Im Modellprojekt ein Testfalltyp *BerechnungsTest* erstellen
 - Eingabe und Erwartete Attribute anlegen
- Testfalltyp Klasse implementieren
 - Methode: `executeBusinessFunction()`
 - Methode: `executeAsserts()`
- Im Produktdatenprojekt Testfall erstellen

Übung: Fachliche Tests mit dem FaktorIPS Testwerkzeug

Weitere Informationen

- www.faktorips.org
- Völter, Stahl: Modellgetriebene Softwareentwicklung
- Evans: Domain Driven Design
- Martin Fowler: Patterns of Enterprise Application Architecture
 - ValueObjects, SpecialCase, Money
- Joshua Bloch: Effective Java
 - Kapitel 8: Exceptions