



# **Faktor-IPS**

## **Übungsunterlagen zur Schulung**

**(Dokumentversion 1326)**

## Übungen zu Kapitel 1: Projekt einrichten, erste Klasse

### Übung 1

Legen Sie ein neues Java-Projekt mit dem Namen „Hausratmodell“ an.

Compliance Level des Java Compilers auf 1.4 setzen

(In den Preferences unter: Java→Compiler: Compiler Compliance Level).

Fügen Sie die Faktor-IPS Nature zu dem Projekt hinzu mit den folgenden Eigenschaften:

- Project type = Model project
- Sourceverzeichnis = modell
- Base Package Name = org.faktorips.schulung.modell
- Runtime ID Prefix = hausrat.

### Übung 2

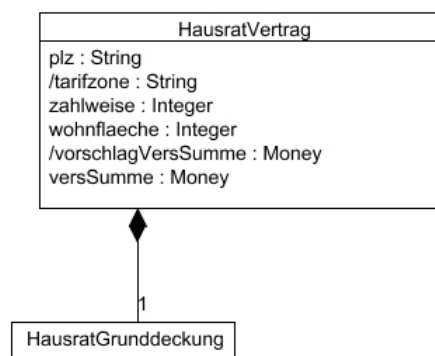
Legen Sie im Sourceverzeichnis „modell“ das Faktor-IPS Package „hausrat“ an.

Legen sie die Klasse „hausrat.HausratVertrag“ an.

Sehen Sie sich den generierten Sourcecode für das published Interface und die Klasse an.

Schauen Sie sich das Projekt im Modellexplorer an.

### Beispielmodell



## Übungen zu Kapitel 2.1: Attribute auf Vertragsseite

### Übung 1

Legen Sie die Attribute der Klasse HausratVertrag an.

<i>Name : Datentyp</i>	<i>Beschreibung, Bemerkung</i>
plz : String	Postleitzahl des versicherten Hausrats
/tarifzone : String	Die Tarifzone ergibt sich aus der Postleitzahl und ist maßgeblich für den zu zahlenden Beitrag. => Achten sie also bei der Eingabe darauf den AttributeType auf derived (computation on each method call) zu setzen!
wohnflaeche : Integer	Die Wohnfläche des versicherten Hausrats in Quadratmetern. Der erlaubte Wertebereich ist min=0 und max=unbeschränkt. Hierzu in das Feld max <null> eintragen lassen.
/vorschlagVersSumme : Money	Vorschlag für die Versicherungssumme. Wird auf Basis der Wohnfläche bestimmt. => Achten sie bei der Eingabe darauf den AttributeType auf derived (computation on each method call) zu setzen!
versSumme : Money	Die Versicherungssumme. Der erlaubte Wertebereich ist min=0 EUR und max=<null>.

### Übung 2

Probieren Sie die Wirkungsweise der Annotations @generated, @generated NOT aus.

### Übung 3

Implementieren Sie die Ermittlung der Tarifzone. Geben Sie immer Tarifzone „I“ zurück. (Später werden wir die Tarifzone anhand der Postleitzahl aus einer Tarifzontabelle ermitteln.)

Implementieren Sie die Ermittlung des Vorschlagswertes für die Versicherungssumme. Der Vorschlag ergibt sich durch  $\text{wohnfläche} * 650 \text{ Euro}$ . (Später werden wir die 650 Euro produktseitig konfigurierbar machen.)

## Übungen zu Kapitel 2-2: Beziehungen auf Vertragsseite

### Übung

Legen Sie die Klasse „hausrat.HausratGrunddeckung“ an.

Legen Sie die Composite-Beziehung zwischen HausratVertrag und HausratGrunddeckung an.

## Übungen zu Kapitel 3-1: Attribute auf Produktseite

### Übung 1

Legen Sie ein neues Java-Projekt "Hausratprodukte" an und fügen die IPS-Nature mit den folgenden Eigenschaften hinzu:

- Project type = Product definition project
- Sourceverzeichnis = produktdaten
- Base Package Name = org.faktorips.schulung.produktdaten
- Runtime ID Prefix = hausrat.

Fügen sie in der Datei ".ipproject" dem IpsObjectPath die Referenz auf das Projekt Grundmodell hinzu.

```
<IpsObjectPath ...>
  <Entry .../>
  <Entry type="project" referencedIpsProject="Hausratmodell"/>
</IpsObjectPath ...>
```

Fügen sie dem Buildpath des Java-Projektes die Referenz auf das Projekt Hausratmodell hinzu.

### Übung 2

Legen Sie die Klasse „HausratProdukt“ an und verknüpfen Sie „HausratVertrag“ mit „HausratProdukt“.

Definieren Sie das Attribute „produktname“

Legen Sie im Projekt „ Hausratprodukte“ im Sourceverzeichnis „produktdaten“ ein neues Package „produkte“ an.

Setzen Sie das aktuelle Wirksamkeitsdatum auf den nächsten Quartalsbeginn.

Legen Sie die beiden Produkte HR-Kompakt und HR-Optimal an und geben Sie im Editor den Produktnamen ein.

### Übung 3

Definieren Sie an der Klasse „HausratProdukt“ ein neues Attribut „vertriebsname“.

Geben Sie für die beiden Produkte einen Vertriebsnamen an.

## Übungen zu Kapitel 3-2: Änderungen im Zeitablauf

Fügen Sie das Attribut "wirksamAb" vom Typ GregorianCalendar zum Vertrag hinzu und implementieren sie die getEffectiveFromAsCalendar() Methode am Vertrag.

Definieren Sie das das Attribut „vorschlagVersSummeProQm“ am HausratratProdukt und implementieren Sie die Berechnung des Vorschlags für die Versicherungssumme in der Klasse HausratVertrag unter Verwendung dieses Attributes. (Lösung im Sourcecode-Ausschnitt 1 im Anhang)

## Übungen zu Kapitel 3-3: Konfigurierbare Vertragsattribute

Markieren Sie die folgenden Attribute der Klasse „HausratVertrag“ als konfigurierbar:

- zahlweise
- wohnflaeche
- versSumme

Geben Sie die Daten für die beiden Produkte gemäß der folgenden Tabelle ein:

<i>Konfigurationsmöglichkeit</i>	<i>HR-Kompakt</i>	<i>HR-Optimal</i>
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

## Übungen zu Kapitel 3-4: Beziehungen auf Produktseite

Legen Sie die Produktklasse „HausratGrunddeckungstyp“ an.

Legen Sie die Beziehungen zwischen „HausratProdukt“ und „HausratGrunddeckungstyp“ an. (Jedes Hausratprodukt soll genau einen Grunddeckungstypen enthalten).

## Übungen zum Kapitel 3-5: Zugriff auf Produktdaten zur Laufzeit

### Übung 1

Legen Sie einen neuen JUnit Testfall HausratTest an. Zum Zugriff auf das Repository verwenden Sie folgenden Setup-Code.

```
private IRuntimeRepository repository;

public void setUp() {
    // Repository erzeugen
    repository= ClassloaderRuntimeRepository.create(
        "org/faktorips/schulung/produktdaten/internal/faktorips-repository-toc.xml");
}
```

Geben sie die folgenden Informationen über das Kompakt-Produkt (in einer test() Methode ) auf der Konsole aus:

- Produktname
- Vorschlag für einen Quadratmeter Wohnfläche

- Defaultwert für die Zahlweise
- Erlaubte Zahlweisen
- Erlaubter Wertebereich für die Versicherungssumme
- Erlaubter Wertebereich für die Wohnfläche

Das Ausführen des Testfalls sollte folgendes Ergebnis liefern:

```
Produktname: Hausrat Kompakt
Vorschlag Vs pro lqm: 600.00 EUR
Default Zahlweise : 1
Erlaubte Zahlweisen: [1, 2]
Bereich Vs: 10000.00 EUR-2000000.00 EUR
Bereich Wohnflaeche: 0-1000
```

## Übung 2

Testen Sie die Methode `HausratVertrag.getVorschlagVersSumme()`, indem Sie den JUnit-Testfall um eine Methode `testGetVorschlagVersSumme()` erweitern.

## Übungen zu Kapitel 4.1: Grundlagen der Verwendung von Tabellen

### Übung 1

Legen Sie die Tabellenstruktur „Tarifzontabelle“ im Projekt „Hausratmodell“ an. Definieren Sie einen Bereich für `plzVon`, `plzBis` und legen Sie den `UniqueKey` auf diesen Bereich an.

<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

Legen Sie den zugehörigen Tabelleninhalt „Tarifzontabelle“ im Projekt „Hausratprodukte“ an.

### Übung 2

Implementieren Sie die Methode `HausratVertrag.getTarifzone()` unter Verwendung der Tarifzontabelle.

Testen Sie ihre Implementierung, indem Sie den JUnit Test um eine Methode `testGetTarifzone()` erweitern.

## Übungen zu Kapitel 4.2: Zusammenhang Bausteine-Tabellen

### Übung 1

Legen Sie die Tabellenstruktur „HausratTariftabelle“ im Projekt „Hausratmodell“ an. Stellen Sie den Tabellentyp auf „Multiple Contents“.

Legen Sie im Projekt Hausratprodukte die Tariftabellen für die beiden Hausratprodukte an.

Tariftabelle-Optimal

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

Tariftabelle-Kompakt

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

### Übung 2

Definieren Sie im Hausratmodell, dass die Klasse HausratGrundeckungstyp die TariftabelleHausrat verwendet (in der Rolle Tariftabelle).

Ordnen Sie den beiden Grundeckungstypen die jeweilige Tariftabelle zu.

### Übung 3

Legen Sie an der Klasse HausratGrunddeckung ein abgeleitetes (cached) Attribut „jahresbasisbeitrag“ vom Typ Money an. Der Jahresbasisbeitrag ist der jährlich zu zahlende Beitrag Zuschläge und Nachlässe, ohne Ratenzahlungszuschlag und ohne Versicherungssteuer.

Definieren Sie eine Methode `void berechneJahresbasisbeitrag()` in der Klasse Hausratgrunddeckung zur Berechnung des Jahresbasisbeitrags.

Implementieren Sie die Methode. Verwenden Sie hierzu die im Baustein zugeordnete Tariftabelle. Die Berechnungsvorschrift ist wie folgt:

- Ermittlung des Beitragsatzes pro 1000 Euro Versicherungssumme aus der Tariftabelle
- Division der Versicherungssumme durch 1000 Euro und Multiplikation mit dem Beitragsatz

Testen Sie ihre Implementierung, indem Sie den JUnit Test um eine Methode `testBerechneJahresbasisbeitrag()` erweitern.

Hinweis:

Sie müssen wie in den vorangegangenen Testfällen zuerst einen `HausratVertrag` erzeugen und dessen Attribute setzen. Zusätzlich müssen Sie nun eine `HausratGrunddeckung` erzeugen und zum Vertrag hinzufügen. Dies geht wie folgt.

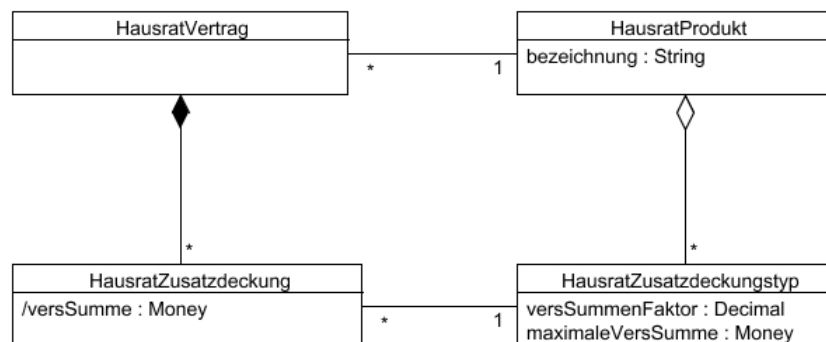
```
// Grunddeckungstyp holen, der dem Produkt in der Anpassungsstufe zugeordnet ist.
IHausratGrunddeckungstyp deckungstyp = kompaktAnpStufe.getHusratGrunddeckungstyp();

// Grunddeckung erzeugen und zum Vertrag hinzufügen
IHausratGrunddeckung deckung = vertrag.newHusratGrunddeckung(deckungstyp);
```

## Übungen zum Kapitel 5

### Übung 1

Erweitern Sie das Hausratmodell um die Zusatzdeckung und den Zusatzdeckungstyp gemäß dem folgenden Diagramm:



Legen Sie die folgenden Zusatzdeckungen an:

	HRD-Fahrraddiebstahl 2008-04	HRD-Aquarien 2008-04
Bezeichnung	Fahrraddiebstahl	Aquarien
VersSummenFaktor	0.01	0.03
MaximaleVersSumme	3000EUR	2000EUR



## Übung 2

Implementieren Sie die Berechnung der Versicherungssumme der Zusatzdeckung.

## Übung 3

Legen Sie das Attribut „jahresbasisbeitrag“ und eine Methode `berechneJahresbasisbeitrag()` zur Berechnung des Attributes an der Klasse Zusatzdeckung an.

Definieren Sie die Formelsignatur für die Berechnung des Jahresbasisbeitrags am Zusatzdeckungstyp. Definieren Sie nun den Nettobasisbeitrag für die Zusatzdeckungstypen in den Bausteinen wie folgt:

- Fahrraddiebstahl: 10% der Versicherungssumme der Deckung
- Aquarien: 5EUR + 3% der Versicherungssumme der Deckung

## Übung 4

Testen Sie die Beitragsberechnung der Zusatzdeckung, indem Sie den JUnit Test um eine Methode `testBerechneJahresbasisbeitrag()` erweitern.