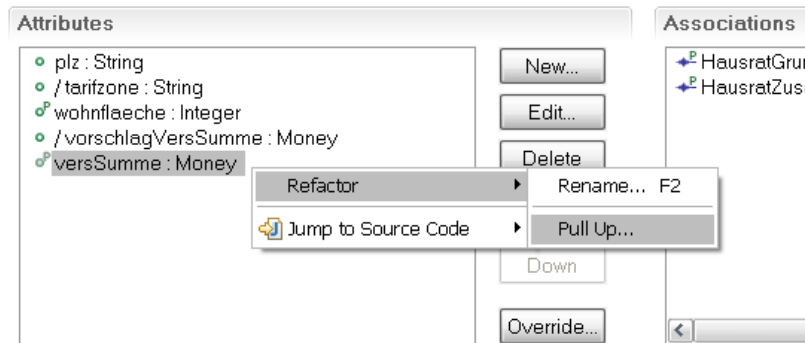


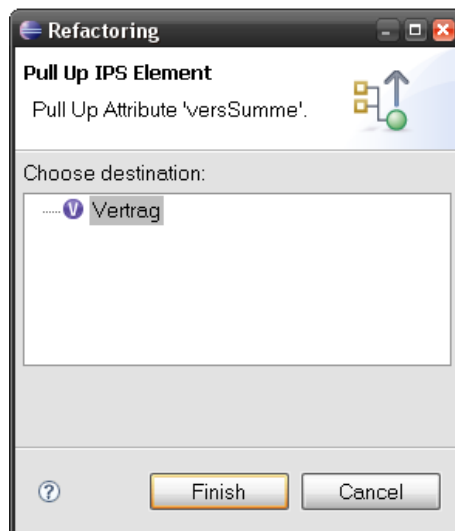
Refactoring

Pull Up Attribute

In Faktor-IPS 3.4 wurde die Refactoring-Unterstützung weiterentwickelt. Attribute können auf diesem Wege jetzt nicht nur umbenannt, sondern auch innerhalb einer Klassenhierarchie "hochgezogen" werden (Pull Up). Diese Funktion wird momentan für Attribute von Produktbausteinklassen, Vertragsteilklassen und Aufzählungstypen angeboten und ist über das entsprechende Kontextmenü erreichbar.



Zur Konfiguration des Refactorings öffnet sich – wie aus den anderen Refactorings bereits bekannt – ein spezieller Wizard. Der Benutzer muss hier lediglich die gewünschte Zielklasse auswählen.



Im Beispiel steht nur eine mögliche Zielklasse zur Verfügung, nämlich die Vertragsteilklassse Vertrag. Diese ist die Superklasse von HausratVertrag, in welcher sich das Attribut versSumme befindet. Ein Klick auf Finish würde bewirken, dass das Attribut versSumme aus HausratVertrag entfernt, und in Vertrag angelegt wird.

Bei dem Refactoring werden Referenzen auf das ursprüngliche Attribut in Vertrag automatisch auf das neue Attribut in HausratVertrag umgeleitet. Auch der für das hochziehende Attribut generierte Java-Quellcode wird automatisch in die entsprechende Java-Superklasse verschoben.

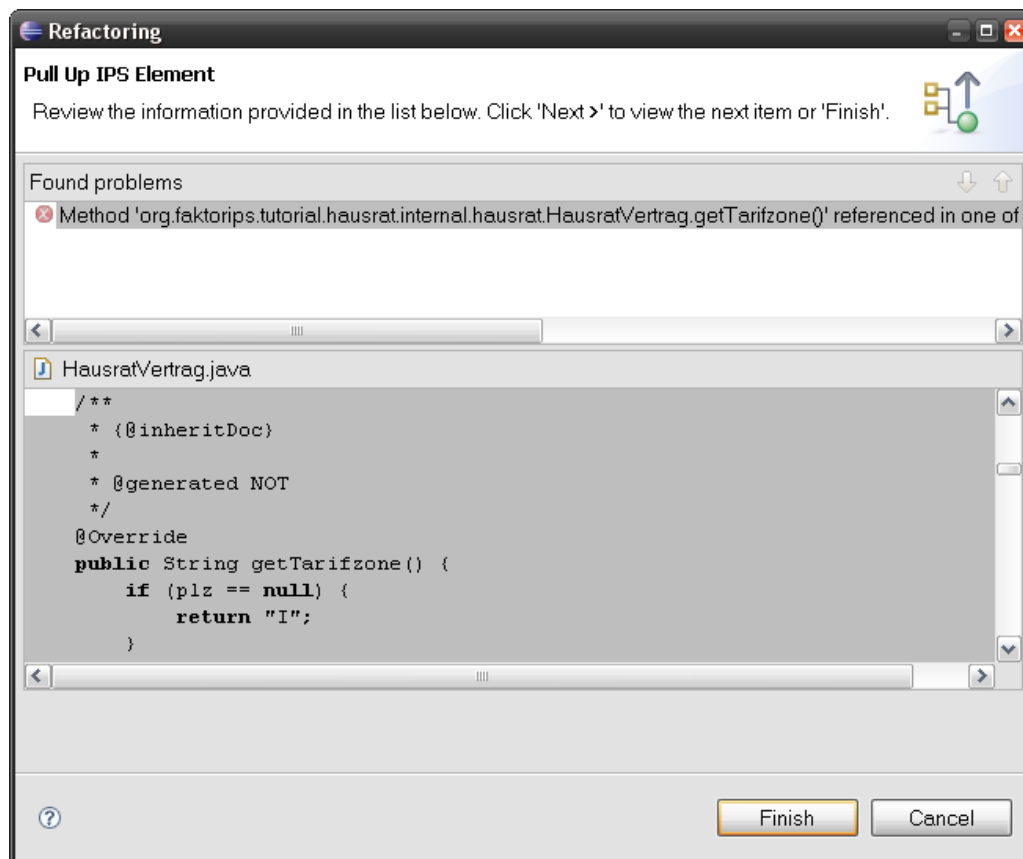
Es kann nicht immer gewährleistet werden, dass der Java-Quellcode nach dem Refactoring frei von Kompilierungsfehlern ist. In diesem Fall wird der Benutzer aber rechtzeitig über diesen Umstand informiert. Beispiel: Das Attribut plz der Klasse HausratVertrag (siehe Tutorial-Projekte) soll in die Superklasse Vertrag hochgezogen werden. Die Methode setPlz (String) wurde manuell bearbeitet:

```

/**
 * {@inheritDoc}
 *
 * @generated NOT
 */
@Override
public void setPlz(String newValue) {
    String oldPlz = plz;
    this.plz = newValue;
    notifyChangeListeners(new PropertyChangeEvent(this, PROPERTY_PLZ,
        oldPlz, plz));
    notifyChangeListeners(new PropertyChangeEvent(this,
        PROPERTY_TARIFZONE, null, getTarifzone()));
}

```

In der letzten Zeile wird die Methode `getTarifzone()` aufgerufen. Die Methode `getTarifzone()` wird aber vom Refactoring nicht automatisch in die Superklasse verschoben. Daher wird es in der Methode `setPlz(String)`, die in die Superklasse `Vertrag` verschoben wird, zu Kompilierungsfehlern kommen, nachdem das Refactoring ausgeführt wurde. Der Refactoring-Wizard weist den Benutzer rechtzeitig auf diesen Umstand hin (siehe Abbildung unten). Es obliegt nun dem Entwickler zu entscheiden, ob das Refactoring trotzdem durchgeführt werden soll.



Move-Wizard unterstützt jetzt das gleichzeitige Verschieben mehrerer Objekte

Der Refactoring-Wizard zum Verschieben von Objekten wurde verbessert. Es ist ab jetzt möglich, den Wizard aufzurufen, auch wenn mehrere Objekte selektiert wurden. Im Wizard wird noch einmal angezeigt, wie viele Elemente verschoben werden. Die Objekte können hierbei aus unterschiedlichen IPS-Paketen stammen. Objekte, die sich bereits im ausgewählten Ziel-Paket befinden, werden vom Refactoring ignoriert.

