EMF - Kalender

Lösungsbeschreibung

1. Zu Beginn wurde ein neues leeres EMF Projekt angelegt und darin ein *Ecore Model* mit dem Namen *emfKalender.ecore* erstellt. Anschließend wurden die 3 Klassen *Year*, *Week* und *Day* mit den vorgegebenen Attributen erstellt. Die Klasse *Year* und *Week* erhalten dabei zusätzlich Referenzen bzw. Containments (*Year* -> *Week*, *Week* -> *Day*).

Da sich für Monate sowie Wochentage Aufzählungstypen eignen kamen zum Modell noch die beiden Enumerationen *WeekDayEnum* und *MonthEnum* hinzu. Die Wochen beginnen am Montag mit dem Wert 1 und die Jahre im Januar mit ebenfalls dem Wert 1.

Das Modell konnte nun bspw. über den generierten Editor instanziiert werden.

1. Die Modellinstanz Darstellung des Editors wurde im generierten Projekt *EMFKalender.edit* angepasst. Die jeweiligen Objektproviderklassen enthalten die Mehode *getText(Object object)*, welche einfach an die gewünschte Darstellung angepasst werden muss. Für die Icons wurden neue Icons über Google-Bilder heruntergeladen und in dem Ressourcen Ordner hinzugefügt. Die alten Icons haben das Dateiende .gif , sodass für die neuen in der Methode *getImage(Object object)* der Pfad noch den Suffix *.png* bekommt.
2. Die Instanz des Metamodells kann über eine Erweiterung der Fabrikklasse *EmfKalenderFactoryImpl* im Projekt *EMFKalender* erzeugt werden. Dazu wurde die Klasse und das dazugehörige Interface(*EmfKalenderFactory*) um die Methode

**public** Year createYear(Integer year) erweitert.

Diese beginnt am ersten Tag des übergebenen Jahres mit einer while-Schleife alle Tage des Jahres durchzugehen, die Methode nutzt dabei die Klasse LocalDate aus Java 1.8. In einer inneren for-Schleife werden die einzelnen Wochentage zu einer *Week* zusammengebaut und diese anschließend dem Jahr zugeordnet. In der letzten Jahreswoche können bereits Tage des Folgejahres enthalten sein, diese werden nicht hinzugefügt!

In der Klasse *Main* (Paket *main*) wird die Fabrikmethode nun genutzt um eine Instanz des Metamodells für das Jahr 2015 zu erstellen.

Die Konsolenausgabe wird mithilfe der Methoden *printYear(Year year)* und *printWeeks(Week week, MonthEnum aktMonth)* realisiert. Eine „Schwierigkeit“ ensteht beim Übergang zwischen den einzelnen Monaten. Dabei kommt es vor, dass die letzte Woche Tage des Folgemonats enthält. Diese dürfen nicht mit ausgegeben werden sondern müssen im Folgemonat unter derselben Kalenderwoche ausgegeben werden.

1. Der Lösungsweg hier war sehr ähnlich zu Aufgabenteil a), weshalb auch Teile wie die Enumerationen daraus kopiert wurden. Der Rest wurde entsprechend der Aufgabenstellung angepasst. Markante Änderung ist die Jahreseinteilung in Monaten anstatt in Kalenderwochen.
2. Die Transformation wurde im Projekt ATL-Kalender in der Datei K1ToK2.atl definiert.

Der einfachste Fall ist hier die Transformation eines Kalender1(a) *Day* in einen Kalender2(d) *Day*. Tagnummer sowie Wochentag werden einfach übernommen. Die Wochennummer muss jedoch aus der gegen Kompositum - Referenz von Kalender1!Day zu Kalender1!Week gewonnen werden, dies erledigt der h*elper getWeekNumber().*

Schwieriger stellte sich die Transformation der einzelnen Kalenderwochen zu Monaten heraus. Dazu werden zunächst im *helper getMonth(month : Kalender2!MonthEnum)* alle Tage zu einem gegebenen Monat aus den Wochen herausgefiltert und in einem *OrderedSet* vereinigt. Als Ergebnis erhält man nun eine Sammlung aller Tage zu einem bestimmten Monat.

Der *helper getMonths()* itertiert nun über alle möglichen Monate und ruft jeweils den *helper getMonth(month: Kalender2!MonthEnum)* auf.

Wenn dem Monat Tage zugeordnet sind, wird ein neuer *Kalender2!Month* erstellt und diesem der Monat und seine Tage zugeordnet. Dieser neu erstellte Ordner landet nun im *OrderedSet* vom *Typ Kalender2!Month*. Zuletzt wird dieses in der *rule Year1Year2* dem *Kalender2!Year* zugeordnet.

Da wir nicht herausfinden konnten, wie man über ein Enum (z.B MonthEnum) iteriert, sind im Quelltext alle Monate explizit angegeben worden. Da ATL und EMF wohl nicht denselben Zeichensatz verwenden, wurde der Monat März zu Maerz umbenannt. Ohne diese Änderung funktioniert das Filtern aller Wochen nach dem Monat März leider nicht.

Vermutete Problemzeile 43: m.union(it.days->select(day | day.month = month))));