



XML スキーマの作成および Java Bean の生成

目次

XML スキーマの作成および Java Bean

の生成 1

演習 1.1: XML スキーマの作成 1

リソース・パースペクティブへの切り替え 1

プロジェクトの作成 1

XML スキーマを作成する 2

演習のチェックポイント 2

演習 1.2: スキーマ・コンポーネントの追加 2

スキーマ・コンポーネントの追加 2

演習のチェックポイント 4

演習 1.3: XML スキーマの編集および妥当性検査 . . . 4

変更の実行、およびリファクタリングの動作方法の
確認 4

名前空間接頭部およびターゲット名前空間を変更す
る 4

スキーマの妥当性検査 6

演習のチェックポイント 6

演習 1.4: XML スキーマ用 Java Bean の生成 7

Bean を含む Java プロジェクトの作成 7

Bean の生成 7

演習のチェックポイント 7

要約 7

XML スキーマの作成および Java Bean の生成

このチュートリアルは、XML スキーマおよび Java™ Bean の概要を提供します。これから行う 4 つの演習を通して、XML スキーマ・エディターを使用して、これらのタイプの文書の作成および編集を行う方法を理解してください。

学習目標

このチュートリアルは、いくつかの演習に分かれており、順番に行っていく必要があります。演習を行う際に、以下の作業を実行してください。

- XML スキーマを作成する。
- 作成した XML スキーマにコンポーネントを追加する。
- リファクタリングの概念を調べる。
- XML スキーマの名前空間接頭部とターゲット名前空間を変更する。
- XML スキーマから Java Bean を生成する。

所要時間

このチュートリアルは、完了するのにおよそ 30 分かかります。このチュートリアルに関連したその他の概念を調べる場合は、完了により長くかかります。

関連情報



PDF バージョンの表示

演習 1.1: XML スキーマの作成

XML スキーマ・ファイルは、XML 文書の構造とセマンティクスを定義する一連の規則を提供します。

XML スキーマを作成し、コンポーネントの追加を行う前に、そのスキーマを格納するためのプロジェクトを作成しておく必要があります。

リソース・パースペクティブへの切り替え

XML プロジェクトの作成は、リソース・パースペクティブ内で行う必要があります。リソース・パースペクティブに切り替えるには、以下のステップに従ってください。

1. メニュー・バーから、「ウィンドウ」 → 「パースペクティブを開く」 → 「その他」の順に選択します。「パースペクティブを開く」のダイアログ・ボックスが表示されます。
2. 「リソース」をクリックします。
3. 「OK」をクリックします。リソース・パースペクティブが開きます。

プロジェクトの作成

これで、以下のステップに従い、お客様の XML スキーマを含めるプロジェクトを作成できます。

1. 「ファイル」 → 「新規」 → 「プロジェクト」の順にクリックします。
2. 「一般」フォルダーを展開し、「プロジェクト」をクリックしてから、「次へ」をクリックします。

3. 「プロジェクト名」フィールドに XMLProject と入力します。
4. 「終了」をクリックします。作成したプロジェクトが、「ナビゲーター」ビューに表示されます。

XML スキーマを作成する。

これで、以下のステップに従い、お客様の XML スキーマを含めるプロジェクトを作成できます。

1. 「ファイル」→「新規」→「その他」の順にクリックします。
2. 「XML」フォルダーを展開して、「XML スキーマ」をクリックします。「XML」フォルダーが表示されない場合は、「すべてのウィザードを表示」チェック・ボックスを選択します。
3. 「次へ」をクリックします。XML 開発機能を使用可能にするかどうかを尋ねるプロンプトが出されたら、「OK」をクリックします。
4. 「XMLProject」を選択し、「ファイル名」フィールドに University.xsd と入力します。
5. 「終了」をクリックします。

University.xsd ファイルが作成され、XML スキーマ・エディターで開かれます。

演習のチェックポイント

ウィザードを使用すると、XML スキーマ用の最初のテンプレートを簡単に作成することができます。

以上で、この演習は終了です。この演習では、次のことを学習しました。

- パースペクティブを切り替える。
- 新規プロジェクトを作成する。
- XML スキーマを作成する。

演習 1.2: スキーマ・コンポーネントの追加

XML スキーマ・エディターは、XML スキーマ・セマンティクスについての十分な知識を持たなくても、簡単に XML スキーマ・コンポーネントを生成できるツールを提供します。

この演習を始める前に、1 ページの『演習 1.1: XML スキーマの作成』を終了しておく必要があります。

XML スキーマ仕様は、複合型、単純型、グループ、注釈、要素、属性などの多数のコンポーネントを定義します。有効なスキーマを作成するには、これらのコンポーネント間の関係を理解する必要があります。例えば、<include>、<import>、または <redefine> 要素は、スキーマ要素の他のどの子よりも前になければなりません。属性は、複合型には追加できますが、単純型には追加できません。この他にも多数の関係があります。

XML スキーマ・エディターを使用すると、そのような詳細をすべて覚えなければならないという負担がなくなります。このエディターの「設計」ビューを使用して、すべてのスキーマ・コンポーネントを追加できます。

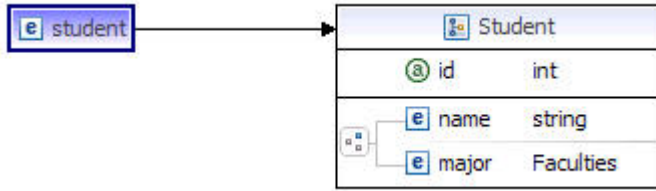
スキーマ・コンポーネントの追加

以下のステップでは、単純型、複合型、およびグローバル要素を含むいくつかのコンポーネントをご使用のスキーマに追加します。

1. University.xsd ファイルには、学生に関する情報を定義するための複合型と、大学内の学部をリストするための単純型とを含める必要があります。そのためには、次のような作業を行います。

- a. 「設計」ビューで、「型 (Types)」カテゴリーを右クリックし、「**複合型の追加**」をクリックします。
 - b. 「NewXSDComplexType」をクリックし、名前を `Student` に変更します。
 - c. 「型 (Types)」カテゴリーを右クリックし、「**単純型の追加**」をクリックします。
 - d. 「XSDSimpleType」をクリックし、名前を `Faculties` に変更します。
2. `Student` 複合型には、学生の名前、専攻、および ID 番号が含まれます。以下のステップに従って、この情報を `Student` 複合型に追加します。
- a. `Student` の詳細にドリルダウンするには、`Student` 複合型をダブルクリックします。
 - b. `Student` の詳細ビューで、`Student` 複合型を右クリックし、「**シーケンスの追加**」を選択します。これにより、新規コンテンツ・モデル・オブジェクトが作成されます。
 - c. 「プロパティ」ビューの「**種類**」ドロップダウン・リストから、すべてを選択します。
- 注: プロパティを表示するには、「**ウィンドウ**」→「**ビューの表示**」→「**プロパティ**」の順にクリックします。
- d. コンテンツ・モデル・オブジェクトを右クリックしてから、「**要素の追加**」をクリックします。新規要素が作成されます。
 - e. 要素を選択し、名前を `name` に変更します。この要素のデフォルトの型が `string` であることに注意してください。
 - f. `Student` 複合型を右クリックし、「**属性の追加**」をクリックします。名前を `id` に変更します。
 - g. 属性タイプをダブルクリックし、ドロップダウン・リストの「**int**」をクリックします。
 - h. コンテンツ・モデル・オブジェクトを右クリックしてから、「**要素の追加**」をクリックします。名前を `major` に変更します。
 - i. 要素タイプをダブルクリックし、ドロップダウン・リストで「**ブラウズ (Browse)**」をクリックします。
 - j. 「**型 (Types)**」リストで、「`Faculties`」を選択し、「**OK**」をクリックします。
3. 単純型 `Faculties` は、単純型 `string` を制限することによって得られ、その値は大学の 1 学部限定されます。このリストを定義するには、次のように、列挙型ファセットを作成します。
- a. `Faculties` 単純型を選択します。
 - b. 「プロパティ」ビューで、「**制約**」タブをクリックします。
 - c. 「**追加**」をクリックし、新規の列挙型として `science` と入力します。
 - d. ステップの a から c までは繰り返し、`Faculties` 単純型に対して、さらに 2 つの列挙型 `arts` と `dentistry` を作成します。
4. この XML スキーマからインスタンス文書を作成するには、XML スキーマにグローバル要素が必要になります。ここで、次のようにして、グローバル要素 `student` を追加します。
- a. スキーマの「設計」ビューで、「要素 (Elements)」カテゴリーを右クリックし、「**要素の追加**」をクリックします。これにより、新規要素が作成されます。
 - b. 「NewElement」をクリックし、名前を `student` に変更します。
 - c. `student` 要素を右クリックし、「**タイプの設定**」→「**ブラウズ (Browse)**」を選択します。「**型 (Types)**」リストで「`Student`」をクリックし、「**OK**」をクリックします。`student` 要素をダブルクリックします。

「設計」ビューが、次のように表示されます。



演習のチェックポイント

XML スキーマ・エディターを使用すると、XML スキーマについての詳細や規則をすべて知らなくても、有効な XML スキーマ・ファイルを作成することができます。

以上で、この演習は終了です。この演習では、次のことを学習しました。

- 作成した XML スキーマにコンポーネントを追加する。
- 単純型または複合型に情報を追加する。
- 単純型の制限事項のリストを定義する列挙型ファセットを作成する。
- 作成した XML スキーマにグローバル要素を追加する。

演習 1.3: XML スキーマの編集および妥当性検査

この演習を始める前に、2 ページの『演習 1.2: スキーマ・コンポーネントの追加』を終了しておく必要があります。

変更の実行、およびリファクタリングの動作方法の確認

スキーマがより大きく、複雑になるにつれ、型定義とそれらの型への参照も増えることになります。それでは、型を定義し、それに対して 10 個の参照を作成した後で、型の名前を変更する必要がある場合にはどのようにすればよいでしょう。XML スキーマ・エディターには、リファクタリング・メカニズムが組み込まれています。これにより、加えられた変更は自動的に伝搬されるため、ユーザーが手動で更新を行う必要はありません。以下のステップでは、この機能を説明します。

ご使用の University.xsd ファイルに **Faculties** と呼ばれる単純型を定義したものとします。**Student** 複合型には、この型に対する参照があります。次の手順で、単純型の名前を **Faculty** に変更することにします。

1. 「設計」ビューで、Faculties 単純型を右クリックし、「リファクタリング」→「名前変更」と選択します。
2. 「新しい名前」フィールドに Faculty と入力し、「OK」をクリックします。
3. 「ソース」ビューに切り替えます (このビューで、ご使用のソース・コードを見ることができます)。要素 major の型が、自動的に tns:Faculty に変更されます。

名前空間接頭部およびターゲット名前空間を変更する。

名前空間は、要素または属性の抽出元を識別する方法を提供します。

University.xsd スキーマ内のターゲット名前空間は、<http://www.example.org/University> です。これは、スキーマ要素内の targetNamespace 属性によって示されます。つまり、このスキーマによって定義される型はすべて、ターゲット名前空間 <http://www.example.org/University> に属していることになります。

スキーマ要素内の次の行は、このターゲット名前空間の接頭部 `tns` を定義しています。

```
xmlns:tns="http://www.example.org/University"
```

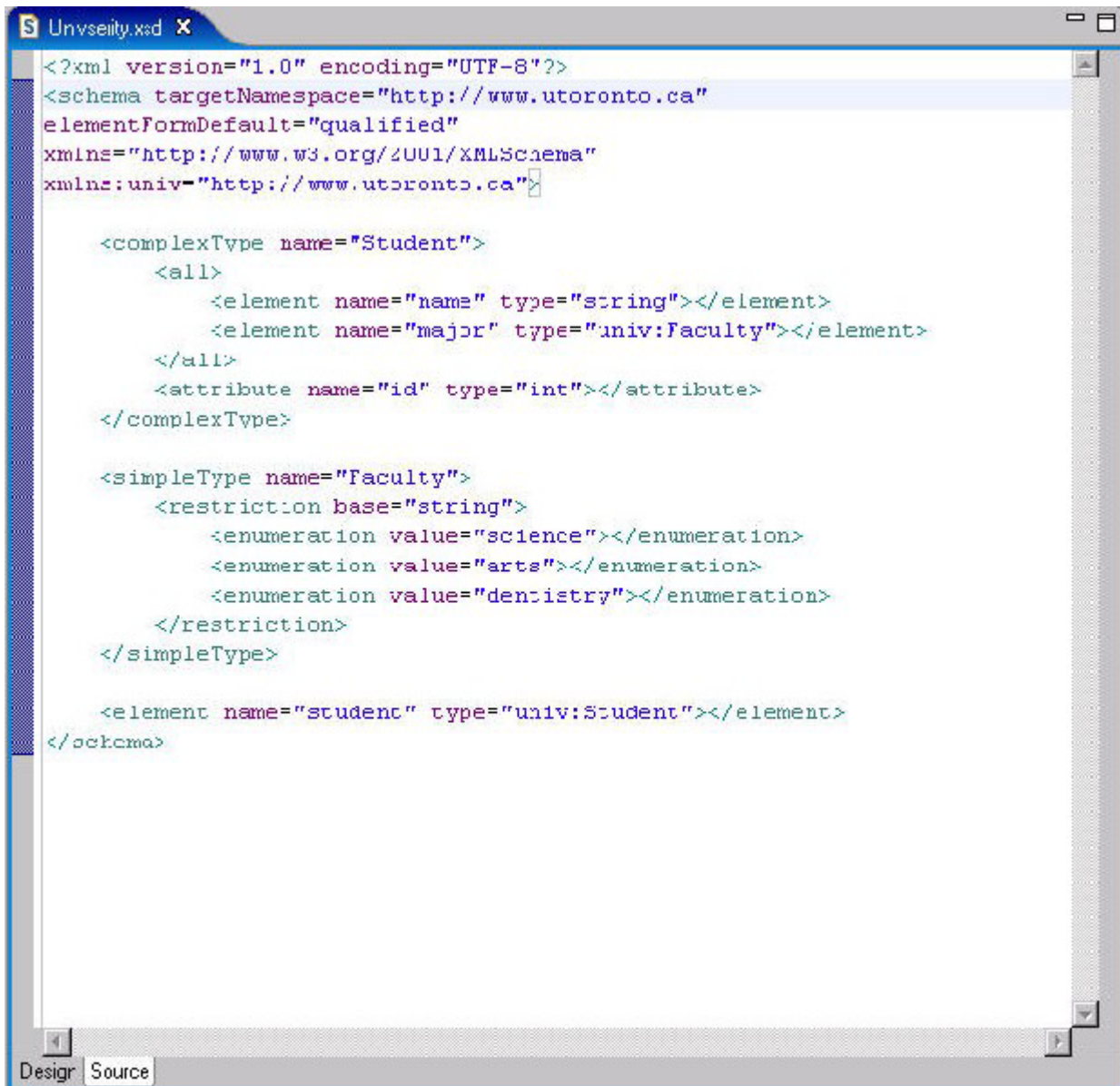
このスキーマで定義されている型を参照するには、この定義済み接頭部を使用する必要があります。「ソース」ビューを見て、`major` 要素と `student` 要素が次のようにして型を参照することに注意してください。

```
<element name="major" type="tns:Faculty"/>
<element name="student" type="tns:Student"/>
```

スキーマの名前空間接頭部またはターゲット名前空間を変更する必要がある場合は、「設計」ビューを使用します。以下のステップに従ってください。

1. 「設計」ビューで、`University` スキーマを選択します。
2. 「プロパティ」ビューに進んで、「接頭部」を `univ` に変更します。
3. 「ターゲット名前空間」を `http://www.utoronto.ca` に変更します。
4. 「ソース」ビューを見てください。スキーマ要素の属性と型のすべての接頭部が自動的にユーザーの指定どおりに変更されていることに注意してください。

コードは、次のようになっているはずです。



スキーマの妥当性検査

XML スキーマ・エディターのもう 1 つの便利な機能はインクリメンタル妥当性検査機能です。XML スキーマ・ファイルを保管した後に、それを「ナビゲーター」ビューで右クリックし、「妥当性検査」をクリックします。検証エラーはいずれも「問題」ビュー内に報告されますが、「ソース」ビューの該当行に赤いマーカーがつけられています。

演習のチェックポイント

XML スキーマ・エディターは、単一ステップで、複数の依存関係に影響する変更を簡単に行うツールを提供します。さらに、このエディターは、名前空間の詳細を指定し、ファイル構文を検査するシンプルなメソッドを提供します。

この演習では、次のことを学習しました。

- 変更を加え、リファクタリングの動作方法を確認する。

- 名前空間接頭部およびターゲット名前空間を変更する。
- XML スキーマの妥当性検査を行う。

演習 1.4: XML スキーマ用 Java Bean の生成

XML スキーマ・エディターは、開発者が XML アプリケーションを迅速に作成できるように、XML スキーマから Java Bean を生成する機能をサポートしています。これらの Java Bean を使用することによって、インスタンス文書を素早く作成したり、XML スキーマ準拠の文書をコーディングなしで、直接 Document Object Model (DOM) API にロードすることができます。

この演習を始める前に、4 ページの『演習 1.3: XML スキーマの編集および妥当性検査』を終了しておく必要があります。

Bean を含む Java プロジェクトの作成

University.xsd ファイルから生成する Bean は、Java ソース・コードと一緒に処理されるように構成されるプロジェクトに含める必要があります。

1. 「ファイル」→「新規」→「その他」の順にクリックします。「新規」ウィンドウで、「Java」→「Java プロジェクト」と選択します。「次へ」をクリックします。
2. 「プロジェクト名」フィールドに UniversityJava と入力し、「終了」をクリックします。
3. Java パースペクティブに切り替えるようにプロンプトが出されます。「はい」をクリックします。

Bean の生成

1. 「ナビゲーター」ビューで、University.xsd を右クリックし、「生成」→「Java」と選択します。
2. 「生成 (Generator)」リストで「SDO Beans 生成 (SDO Beans Generator)」が選択されており、作成した「UniversityJava」プロジェクトがコンテナーとして選択されていることを確認した後、「終了」をクリックします。

UniversityJava プロジェクト内の「ナビゲーター」ビューに Bean が表示されます。「問題」ビューに表示される警告はすべて無視できます。

演習のチェックポイント

XML スキーマ・エディターは、DOM API を必ずしも十分に理解していなくても Java Bean を使用するのに必要なツールを提供します。

この演習により、Java Bean の基本を理解できます。以上で、この演習は終了です。この演習では、次のことを学習しました。

1. Bean を含む Java プロジェクトを作成する。
2. Java Bean を生成する。

要約

お疲れ様でした。XML スキーマの作成と変更、および変更済みスキーマを基にした Java Bean の生成は、正常に行われました。

達成した学習目標

すべての演習を完了すると、以下の方法を理解できるようになります。

- XML スキーマを作成する。
- 作成した XML スキーマにコンポーネントを追加する。
- リファクタリングの概念を理解する。
- XML スキーマの名前空間接頭部とターゲット名前空間を変更する。
- XML スキーマから Java Bean を生成する。

参考情報

このチュートリアルで扱ったトピックについて、さらに学習したい場合は、次の情報源を参照してください。

- w3.org Web サイト: XML Schema Part 0: Primer は、XML スキーマ言語を使用して、スキーマを作成する方法を短期間で理解することに重点が置かれています。
- オンライン・ヘルプ (「ヘルプ」>「ヘルプ目次」) には、XML スキーマ・エディターのドキュメンテーションが含まれています。