

Kata "Testdaten"

Nach wie vor spielt die klassische „Forms over Data“-Anwendung eine grosse Rolle. Daten aus einer Datenbank sollen per Formular bearbeitet werden. Wenn diese Applikationen getestet werden, spielen Testdaten eine zentrale Rolle. Möglichst viele sollten es sein und möglichst realistisch geformt noch dazu.

Immer wieder begegnet man der Anforderung, Daten aus einer Datenbank in einem Formular zu visualisieren. Oft sind die Datenmengen dabei so gross, dass man nicht einfach alle Daten in einem Rutsch laden sollte. Stattdessen müssen die Daten seitenweise abgerufen und visualisiert werden. Suchen und Filtern kommen meistens hinzu, und schon stellt sich die Frage, ob der gewählte Ansatz auch noch funktioniert, wenn mehr als nur eine Handvoll Testdaten in der Datenbank liegen.

Aufgabe

Erstellen Sie eine Bibliothek zum Erzeugen von Testdaten.

Anforderungen

Die generierten Testdaten sollen eine Tabellenstruktur haben. Für jede Spalte wird definiert, von welchem Typ die Werte sind und wie sie erzeugt werden. Anschliessend gibt man an, wie viele Zeilen generiert werden sollen, und die Testdaten werden generiert.

Die Anforderungen an die Daten können sehr vielfältig sein. Um hier ausreichend flexibel zu sein, sollen die Daten nach verschiedenen Strategien erzeugt werden können. Reine Zufallsdaten sind ein erster Schritt, dürften aber in vielen Fällen nicht ausreichen. Zumindest eine Beschränkung innerhalb vorgegebener Minimum- und Maximumwerte erscheint sinnvoll. Eine weitere Strategie könnte darin bestehen, eine Liste von möglichen Werten vorzugeben, aus denen dann zufällig ausgewählt wird. So könnten beispielsweise Strassennamen generiert werden, die in den Formularen dann auch wie Strassennamen aussehen statt wie zufällig zusammengewürfelte Zeichenfolgen. Es müssen lediglich einige Strassennamen vorgegeben werden. Das Gleiche bietet sich für die Namen von Personen an. Auch hier kann gut mit einer Liste von Namen gearbeitet werden, aus der dann zufällig Werte ausgewählt werden. Die Strategie für die Testdatenerzeugung soll möglichst flexibel sein. Ein Entwickler sollte mit wenig Aufwand einen eigenen Generator ergänzen können. Endergebnis der Datenerzeugung soll eine Aufzählung von Zeilen sein.

Die generierten Zeilen können dann beliebig verwendet werden. Sie können direkt in Tests einfließen oder auch zuerst als Datei gespeichert werden. Hier bietet sich beispielsweise die Speicherung als CSV-Datei an. Auch das Speichern in einer Datenbank ist natürlich ein typisches Szenario. Das konkrete Speichern der Daten sollte unabhängig sein vom Erzeugen. Es lohnt sich also wieder, sich vor der Implementierung ein paar Gedanken zur Architektur zu machen.

Bei dieser Übung geht es wieder primär um eine Bibliothek und weniger um eine Benutzerschnittstelle. Wer mag, kann sich aber auch um eine Benutzerschnittstelle kümmern, denn die dürfte hier etwas anspruchsvoller sein. Schliesslich benötigen die verschiedenen Generatoren unterschiedliche Eingabedaten. Genügen bei einem Zufallsgenerator vielleicht Minimum und Maximum, müssen bei einem anderen Generator Wertelisten eingegeben werden. Hinzu kommt, dass die Eingabedaten von unterschiedlichem Typ sein können, wofür unterschiedliche Eingabvalidierungen nötig sind.

Ziele

- Test getriebe Entwicklung (TDD)
- Fluent-Interface