

Fortgeschrittene Programmiertechniken

Prof. Dr. J. Pauli, Dipl.-Inform. J. Hoefinghoff, J. Kapitza, M. Cherubim

Übungsblatt 2

Seite 1 von 3

In der Vorlesung haben Sie das Konzept der Persistenz in Java mittels Serialisierung kennengelernt. In den folgenden Aufgaben sollen einige Aspekte dieses Konzepts näher untersucht werden. Für die Implementierung der Klassen soll das **Strategie-Muster** verwendet werden. **Fehlerhafte Implementierungen führen zu Punktabzug. Sie haben alle nötigen Informationen in der Vorlesung erhalten.** Für die Bearbeitung der Aufgaben nutzen Sie bitte das Interface *fpt.com.SerializableStrategy*, welches von jeder Serialisierungsstrategie implementiert werden muss.

Aufgabe 1

HardwareShop

Bauen Sie die Serialisierung an beliebiger Stelle in das Shop System ein. Für die Auswahl der Strategie könnte beispielsweise ein Menü oder ein Dialog - beim Schließen bzw. Öffnen des Programms - verwendet werden. Sie haben bereits ein einfaches Produkt und eine dazugehörige Produktliste implementiert. Passen Sie ihre Implementierung an die unten genannten Kriterien an, sofern diese denen noch nicht entsprechen.

- a) Sorgen Sie dafür, dass jedes Produkt eine eindeutige ID bekommt.
Erstellen Sie dazu die Klasse *IDGenerator*, die Ihnen immer eine gültige ID gibt.
Diese darf nicht mehrfach existieren.
Der Generator startet mit dem Wert 1 und generiert einen maximalen Wert von 999999.
Wird der Maximalwert überschritten, so wird ein Exception-Objekt geworfen. (throws Exception)
Wird keine Exception geworfen, gibt es für den Generator keine Punkte.

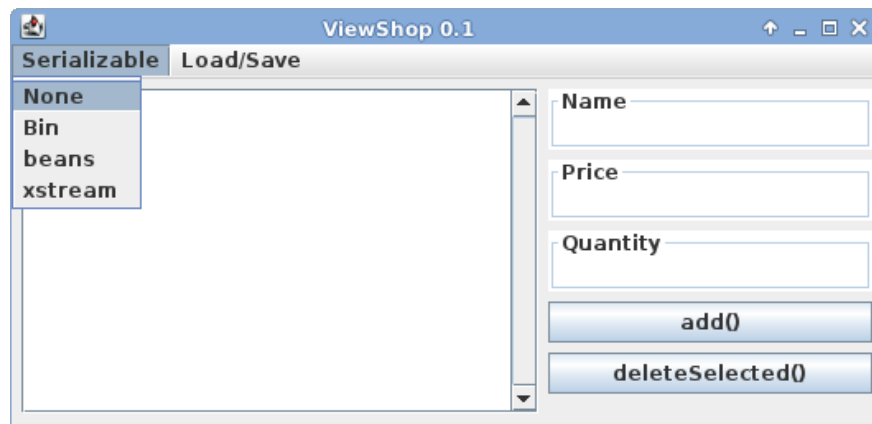


Abbildung 1: Menübeispiel Auswahl der Serialisierung

Fortgeschrittene Programmiertechniken

Prof. Dr. J. Pauli, Dipl.-Inform. J. Hoefinghoff, J. Kapitza, M. Cherubim

Übungsblatt 2

Seite 2 von 3

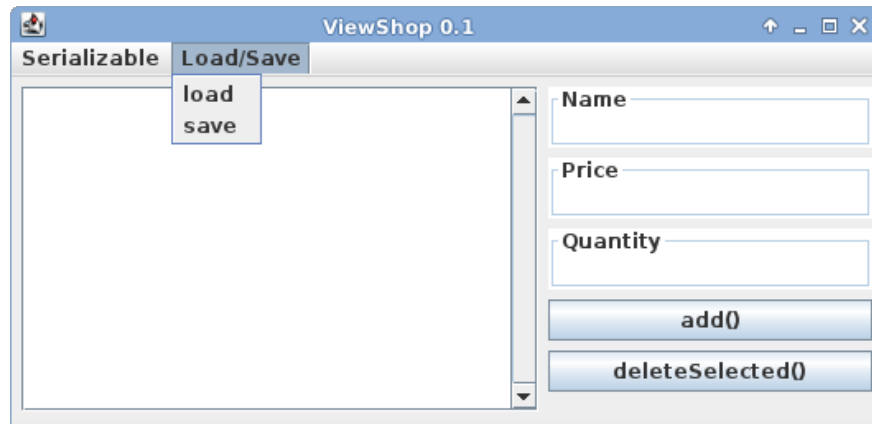


Abbildung 2: Menübeispiel Serialisierung nutzen

Aufgabe 2

Binary-Serialisierung

Bei der Objektserialisierung ist zu beachten, dass nicht nur das eigentliche Objekt serialisiert wird, sondern auch alle Objekte, die dieses Objekt als Attribut enthält. Der Fachausdruck dafür lautet 'Persistenz durch Erreichbarkeit'. Ihre Warenliste sollte mindestens 5 Objekte enthalten.

- Welche Kriterien muss eine Klasse erfüllen, damit diese serialisiert werden kann?
- Entwerfen Sie die Klasse *BinaryStrategy*, die das Interface *fpt.com.SerializableStrategy* implementiert.
- Serialisieren und deserialisieren Sie die Objekte in eine Datei *products.ser*.
- Welche Probleme können entstehen, wenn die Klassen nach der Serialisierung verändert werden?
Provozieren Sie einen Fehler, der durch die Deserialisierung einer nach der Serialisierung geänderten Klasse entsteht. Welche Möglichkeiten gibt es, um diesen Fehler zu verhindern?
- Wie hängen *serialVersionUID* und die Klasse zusammen?
- Wie kann man durch die Serialisierung ein Objekt kopieren?

Fortgeschrittene Programmiertechniken

Prof. Dr. J. Pauli, Dipl.-Inform. J. Hoefinghoff, J. Kapitza, M. Cherubim

Übungsblatt 2

Seite 3 von 3

Aufgabe 3

Java Beans XML-Serialisierung

Nutzen Sie die XML-Serialisierung, um flexibler in der Datenhaltung zu werden. Verwenden Sie dafür Javas eingebaute *java.beans.XMLEncoder* bzw. *java.beans.XMLDecoder* Klassen.

- Schreiben Sie die Klasse *XMLStrategy*, die das Interface *fpt.com.SerializableStrategy* implementiert.
- Serialisieren und deserialisieren Sie die Objekte in eine Datei *products.xml*
- Wie unterscheiden sich die Arten der Serialisierung?
- Nennen Sie Nachteile bzw. Vorteile der Ihnen bekannten Serialisierungsmechanismen.

Aufgabe 4

XStream XML-Serialisierung

Ihnen liegt nun folgendes XML Format vor.

```
<waren>
  <ware id="000001">
    <name>Das ist Ware eins.</name>
    <preis>12.50</preis>
  </ware>
  <ware id="000002">
    <name>Das ist Ware zwei.</name>
    <preis>12.50</preis>
    <anzahl>41</anzahl>
  </ware>
  <!--...-->
</waren>
```

Sie sollen das Format lesen und schreiben können. Nutzen Sie dazu [xstream](#). Bitte denken Sie daran, alle Attribute der Objekte zu speichern.

- Klären Sie, wie Attribute erstellt werden und wie man Eigenschaften eines Objektes umbenennen kann.
- Klären Sie, wie man ein Objekte konvertiert bzw. [dekorieren](#) kann.
- Erstellen Sie die Klasse *XStreamStrategy*, die das Interface *fpt.com.SerializableStrategy* implementiert.
- Die ID soll sechsstellig als Attribute im waren-Tag stehen. Wenn nötig mit führenden **0** auffüllen.
- Der Preis ist mit zwei Nachkommastellen anzugeben und verwendet einen Punkt als Trenner.
- Für beide Transformationen können sie einen *SingleValueConverter* verwenden.