Erste Schritte mit db4o

Die Übungen mit DB4O können Sie wahlweise in Java, Visual Basic oder auch in C# durchführen.

1. **Installieren von db4o**
   Unter `www.db4o.com` (oder vom Handoutserver `Hoffmann/Konzepte der Datenbanktechnologie`) kann `db4o` in der aktuellen Release-Version 8.0 geladen werden.

2. **Eclipse– oder Visual-Studio-2013-Projekt**
   Legen Sie bitte in Eclipse ein Java–Projekt oder aber in Visual Studio ein Visual-Basic-Konsolen-Projekt an.

   Java

   Erzeugen Sie in dem Projekt einen Ordner `lib`, kopieren Sie `db4o-8.0.249.16098-all-java5.jar` dorthin und nehmen Sie es in die Liste der verwendeten Bibliotheken auf (Java Build Path). `db4o-8.0.249.16098-all-java5.jar` enthält die `db4o`–Engine für das Java-5- und Java-6-SDK. Für unsere Programme in dieser Übung ist das ausreichend.

   Visual Basic

   Fügen Sie bitte den Verweis auf `DB4objects.Db4o.dll` dem Projekt hinzu (Projekt>Verweis hinzufügen). Für unsere Übung ist das ausreichend. Damit steht `db4o` Visual-Basic-Programmen zur Verfügung.

3. **Die Objekte**
   Die Objekte, die wir in dieser Übung verwalten wollen, beschreiben in Anlehnung an das `db4o`-Tutorial Formel-1-Fahrer (Pilots). Sie sind in Java wie folgt definiert (dieses und das Visual-Basic-File finden sich auch auf dem Handout–Server):

```
package de.fhwedel.kdbt.ueb01;

public class Pilot {
    private String m_name;
    private int m_points;

    public Pilot(final String name, final int points) {
        m_name = name;
        m_points = points;
    }

    public int getPoints() { return m_points; }

    public void addPoints(final int points) { m_points += points; }

    public String getName() { return m_name; }

    public String toString() { return m_name + "/" + m_points; }
}
```

4. **Der Rahmen**
   Um mit einer `db4o`-Datenbank zu arbeiten, muss sie geöffnet und ein `ObjectContainer`–Objekt (Java) bzw. ein Visual-Basic-`IEmbeddedObjectContainer` erzeugt werden. Wir verwenden zunächst lokale Datenbanken in Files, die mit `Db4o.openFile` (Java) bzw. `DB4oEmbedded.OpenFile` (Visual Basic) geöffnet werden können. Nach getaner Arbeit, muss die Datenbank mit `close()` wieder geschlossen werden.

   Schreiben Sie bitte ein Rahmenprogramm, das eine Datenbank öffnet, Platz für die Datenbankoperationen der folgenden Aufgaben lässt und die Datenbank in einem `finally`–Block wieder schließt.

Java:

```
new File( <name-der-datenbank> ).delete();
ObjectContainer db = Db4o.openFile( <name-der-datenbank> );
try {
    ...
} finally {
    db.close();
}
```

Visual Basic:

```
Imports EB4objects.Db4o
...
Dim db As IEmbeddedObjectContainer = DB4oEmbedded.Openfile( <name-der-datenbank> )
Try
    ...
Finally
    db.Close()
End Try
```

5. db4o **Create, Read, Update, Delete**

   (a) Das C in CRUD
   Mit Hilfe der `ObjectContainer`–Methode `store()` werden Objekte in die Datenbank geschrieben. Befindet sich ein Objekt noch nicht in der Datenbank, wird es dort angelegt.
   Schreiben Sie bitte ein einfaches Programm, dass Pilot–Objekte (Etwa mit Namen *Michael Schumacher*, *Rubens Barricello*, *Jenson Button* oder *Nick Heidfeld*) erzeugt und in die Datenbank schreibt.

   (b) Das R in CRUD
   `db4o` verwendet unterschiedliche Datenbank–Abfragetechniken. Wir wollen im Moment *Query by Example* einsetzen. Dabei wird ein Prototyp–Objekt erzeugt, das an einigen Attributen gewünschte konkrete Werte enthält. Attribute deren Wert bei der Abfrage beliebig sein sollen, bleiben uninitialisiert (oder bekommen die default-Werte `null`, `0`, ...). Mit Hilfe der `ObjectContainer`–Methode `queryByExample()` werden dann alle Objekte der Datenbank ermittelt, die auf den Prototypen passen. Ergebnis ist ein `ObjectSet`, über das iteriert werden kann.
   Erweitern Sie bitte das obige Programm, so dass Pilot–Objekte aus der Datenbank ausgelesen und auf der Konsole ausgegeben werden.
   Wie geht man sinnvollerweise vor, wenn man bei einer Abfrage nur an einem einzelnen Objekt interessiert ist?

   (c) Das U in CRUD
   Für das Aktualisieren eines Objekts wird ebenfalls die `ObjectContainer`–Methode `store()` verwendet.
   Schreiben Sie bitte eine Methode, die ein gegebenes Objekt in der Datenbank aktualisiert (beispielsweise die Formel-1-Punkte eines Pilot-Objekts erhöht und es dann wieder in die Datenbank schreibt). Überprüfen Sie bitte das Resultat.
   Wie unterscheidet `store()` zwischen neuen und veränderten Objekten?

   (d) Das D in CRUD
   Die `ObjectContainer`–Methode `delete()` ist für das Löschen von Objekten in der Datenbank verantwortlich.
   Ergänzen Sie bitte Ihr Programm so, dass am Ende alle Pilot-Objekte aus der Datenbank gelöscht werden. Überprüfen Sie bitte das Resultat.