Übungsblatt 8

1. Installiere den MultiJava Compiler und die JML Tools, verfügbar von:

   ```
   http://multijava.sourceforge.net/downloads.shtml
   http://www.cs.iastate.edu/~leavens/JML/download.shtml
   ```

   Entweder die aktuelle Release, oder besser die letzte CVS Version.
   Es wird eine Java 1.4 Umgebung benötigt!
   Nach erfolgreichem setzen der Umgebungsvariablen kann man mit einem Aufruf von:

   ```
   $ java org.jmlspecs.checker.Main --universes
         -w3 -forg.multijava.mjc.UniverseFilter XXX.java
   ```

   ein Programm mit Universe-Annotationen überprüfen.

2. Verwende das Universe Typsystem, um die ArrayListe aus der Übungsstunde zu annotieren.

3. Gegeben sind die folgenden Klassen für eine verlinkte Liste:

```
class Entry {
     Object element;
     Entry previous, next;

     Entry( Object o, Entry p, Entry n ) {
          element = o; previous = p; next = n; }
}

public class LinkedList {
     private Entry header;

     public LinkedList() {
          header = new Entry(null, null, null);
          header.next = header;
          header.previous = header;
     }

     public void add( Object o ) {
          Entry newE = new Entry(o,header,header.next);
          header.next.previous = newE;
          header.next = newE;
     }

     public Object get( int idx ) {
          Entry e = header.next;

          for( int i=0; i<idx; ++i ) { e = e.next; }

          return e.element;
```

```
        }

        public ReadIterator getReadIterator() {
             return new ReadIterator( header );
        }

        public DeleteIterator getDeleteIterator() {
             return new DeleteIterator( header );
        }


        public static void main( String[] args ) {

             LinkedList ll = new LinkedList();

             ll.add( new Integer(20) );
             ll.add( "xyz" );
             ll.add( new Float(2.2f) );
             ll.add( "Hello World" );
             ll.add( new Object() );

             ReadIterator itr = ll.getReadIterator();

             int i = 0;
             while( itr.hasNext() ) {
                  itr.moveNext();
                  System.out.println("Element[" + i + "]: "
                                     + itr.element() );
                  ++i;
             }

             DeleteIterator itd = ll.getDeleteIterator();
             itd.moveNext();
             itd.moveNext();
             itd.delete();
             System.out.println("Deleted second element.");

             itr = ll.getReadIterator();

             i = 0;
             while( itr.hasNext() ) {
                  itr.moveNext();
                  System.out.println("Element[" + i + "]: "
                                     + itr.element() );
                  ++i;
             }
        }
   }
```

Die Iteratoren sind wie folgt implementiert:

```
public class ReadIterator {

     public ReadIterator( Entry h ) {
          // the header is a dummy that is never null
          current = h;
          header = h;
     }

     public boolean hasNext() {
          return current.next != header;
     }

     public void moveNext() {
          current = current.next;
     }

     public Object element() {
          return current.element;
     }

     protected Entry current;
     protected Entry header;
}

public class DeleteIterator extends ReadIterator {

     public DeleteIterator( Entry h ) {
          super( h );
     }

     public void delete() {
          if( current.previous != null )
               current.previous.next = current.next;

          if( current.next != null )
               current.next.previous = current.previous;

          current = current.next;
     }
}
```

Der Quellcode zu dieser Übung ist auf der Website verfügbar!

a) Verwende das Universe Typsystem, um die Kapselung von LinkedList sicherzustellen.
b) Annotiere den ReadIterator.
c) Annotiere den DeleteIterator.
d) Lassen sich beide Iteratoren mit dem Universe Typsystem typisieren? Begründe deine Antwort und nimm gegebenenfalls Änderungen an der Implementierung vor, so dass eine Typisierung möglich ist.

4. Stelle Übung 3 von Übungsblatt 6 fertig.

5. **Wettbewerb:** Finde ein Programm, das

a) das Universe Typsystem verwendet,
b) sich mit dem CVS Compiler ohne Warnung übersetzen lässt,
c) aber trotzdem die Ownership-Invariante verletzt.