**Designing to Interfaces**

Java provides collection interfaces such as *java.util.Collection* (which is the parent to all the collection interfaces except the map interface), *java.util.List* (the interface for list data structure), *java.util.Set* (for set data structures), *java.util.Map* (for key/value map data structures), and more.

The *List* implementation is intended to hold ordered list of elements. We use concrete implementations, such as ArrayList or LinkedList, but point here is that we must design to the *List* interface in our Hibernate rather than to its concrete implementation.

Always use interfaces when you are defining your collection variables. Hibernate does not like it when we use concrete classes as the variable types:

AcroRd32_2017-01-13_14-55-46.png

Instead, we should define the types using interfaces, like so:

AcroRd32_2017-01-13_14-55-46.png

The reason is that behind the scenes, Hibernate use its own implementation of List!