Pattern Book Week 6

# Looping list pattern

# Copying list patterns(2 solutions)

LinkedList<String> copy = new LinkedList<String> ();  
copy.addAll (original);

# Lookup pattern

Private Account account (String type) {  
 for (Account account : accounts)  
 if (type.equals (account.getType()))  
 return account;  
 return null;  
}

# Match function

**Client**public class Customer {  
 private Account account (String type) {  
 for (Account account : accounts)  
 if (account.hadType (type))  
 return account;  
 return null;  
 }  
}

**Supplier**public class Account {  
 private String type;  
   
 publiv boolean hasType (String type) {  
 return type.equals (this.type);  
 }  
}

# Find all matches function

# Remove all matches function(two solutions)

**Solution 1:**LinkedList<String> zWords = zWords(list);  
list.removeAll (zWords);

**Solution 2:**for (Iterator<String> it = list.iterator(); it.hasNext() ;)  
 if (it,next ().contains(“z”))  
 it.remove();

# Java’s ConcurrentModificationException

# Remove one match function(two solutions)

**Solution 1:**for (String word : list)  
 if (word.contains (“z”)) {  
 list.remove (word);  
 break;  
 }

**Solution 2:**for (Iterator<String> it = list.iterator() ; it.hasNext() ; )  
 if (it.next().contains(“z”)) {  
 it.remove();  
 break;  
 }