EntryMap.java

```
1 package st;
 3 import java.util.ArrayList;
 6 public class EntryMap {
 8
      private ArrayList<Entry> entries;
 9
10
      private HashSet<Entry> uniqueEntries;
11
12
      public EntryMap(){
          entries = new ArrayList<>();
13
14
          uniqueEntries = new HashSet<>();
15
      }
16
17
      public void store(String pattern, String value, Boolean caseSensitive)
  throws RuntimeException{
          if (caseSensitive == null){
18
              caseSensitive = Boolean.FALSE;
19
20
21
          Entry entry = new Entry(pattern, value, caseSensitive);
22
          if (!isEntryValid(entry)){
23
              throw new RuntimeException();
24
          }
25
          if (isEntryUnique(entry)){
26
27
               addEntry(entry);
28
          }
29
      }
30
31
      private Boolean isEntryValid(Entry entry){
32
          if (entry.getPattern()== null)
33
               return Boolean. FALSE;
34
          if (entry.getPattern().isEmpty())
35
               return Boolean.FALSE;
36
          if (entry.getValue() == null)
37
               return Boolean.FALSE;
38
          return Boolean.TRUE;
39
      }
40
41
      private Boolean isEntryUnique(Entry entry){
42
          return !uniqueEntries.contains(entry);
43
44
45
      private void addEntry(Entry entry){
46
          entries.add(entry);
47
          uniqueEntries.add(entry);
48
      }
49
50
      public ArrayList<Entry> getEntries() {
51
           return entries;
52
      }
53
```

EntryMap.java

```
class Entry {
54
55
           String pattern;
56
           String value;
57
           Boolean caseSensitive;
58
59
           public Entry(String pattern, String value, Boolean caseSensitive) {
60
               this.pattern = pattern;
61
               this.value = value;
62
               this.caseSensitive = caseSensitive;
63
           }
64
65
           public String getPattern() {
66
               return pattern;
67
           }
68
69
           public String getValue() {
70
               return value;
           }
71
72
73
           public Boolean getCaseSensitive() {
74
               return caseSensitive;
75
           }
76
77
           @Override
78
           public boolean equals(Object o) {
79
               if (this == 0) return true;
80
               if (o == null || getClass() != o.getClass()) return false;
81
82
               Entry entry = (Entry) o;
83
84
               if (!getPattern().equals(entry.getPattern())) return false;
85
               if (!getValue().equals(entry.getValue())) return false;
               return getCaseSensitive() != null ? getCaseSensitive().equals
86
  (entry.getCaseSensitive()) : entry.getCaseSensitive() == null;
87
           }
88
89
           @Override
90
           public int hashCode() {
               int result = getPattern().hashCode();
91
               result = 31 * result + getValue().hashCode();
result = 31 * result + (getCaseSensitive() != null ?
92
93
  getCaseSensitive().hashCode() : 0);
94
               return result;
95
           }
96
      }
97
98 }
99
```