

```
1 import components.set.Set;
2 import components.set.Set1L;
3
4 /**
5  * Layered implementations of secondary methods {@code add} and {@code remove}
6  * for {@code Set}.
7  *
8  * @param <T>
9  *         type of {@code Set} elements
10 */
11 public final class SetSecondary1L<T> extends Set1L<T> {
12
13     /**
14      * No-argument constructor.
15      */
16     public SetSecondary1L() {
17         super();
18     }
19
20     @Override
21     public Set<T> remove(Set<T> s) {
22         assert s != null : "Violation of: s is not null";
23         assert s != this : "Violation of: s is not this";
24
25         Set<T> temp = s.newInstance();
26
27         for (T x : s) {
28             if (this.contains(x)) {
29                 temp.add(this.remove(x));
30             }
31         }
32         return temp;
33     }
34
35     @Override
36     public void add(Set<T> s) {
37         assert s != null : "Violation of: s is not null";
38         assert s != this : "Violation of: s is not this";
39
40         int i = 0;
41         while (i < s.size()) {
42             T val = s.removeAny();
43
44             if (!this.contains(val)) {
45                 this.add(val);
46             } else {
47                 s.add(val);
48             }
49             i++;
50         }
51     }
52
53 }
```