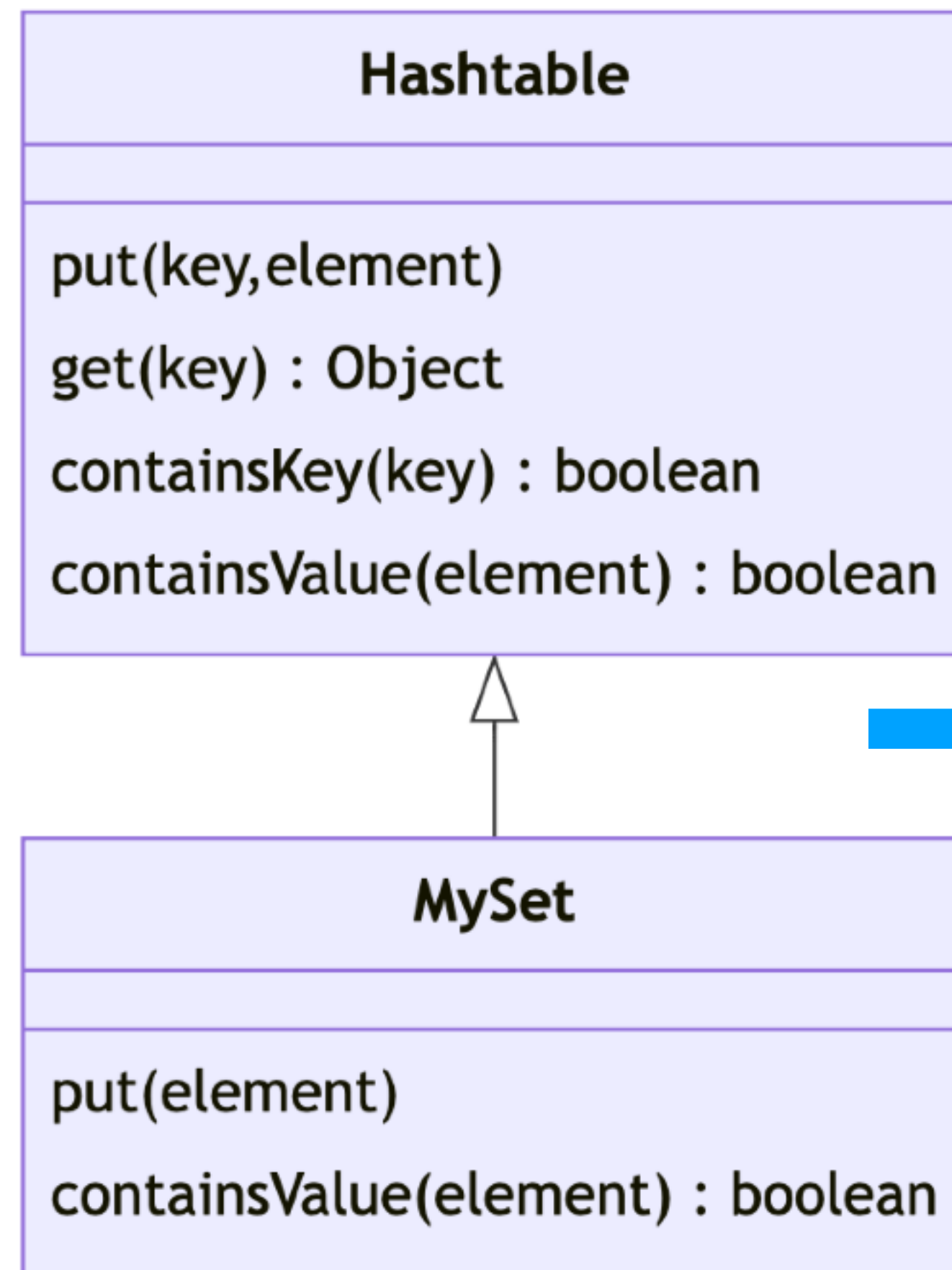
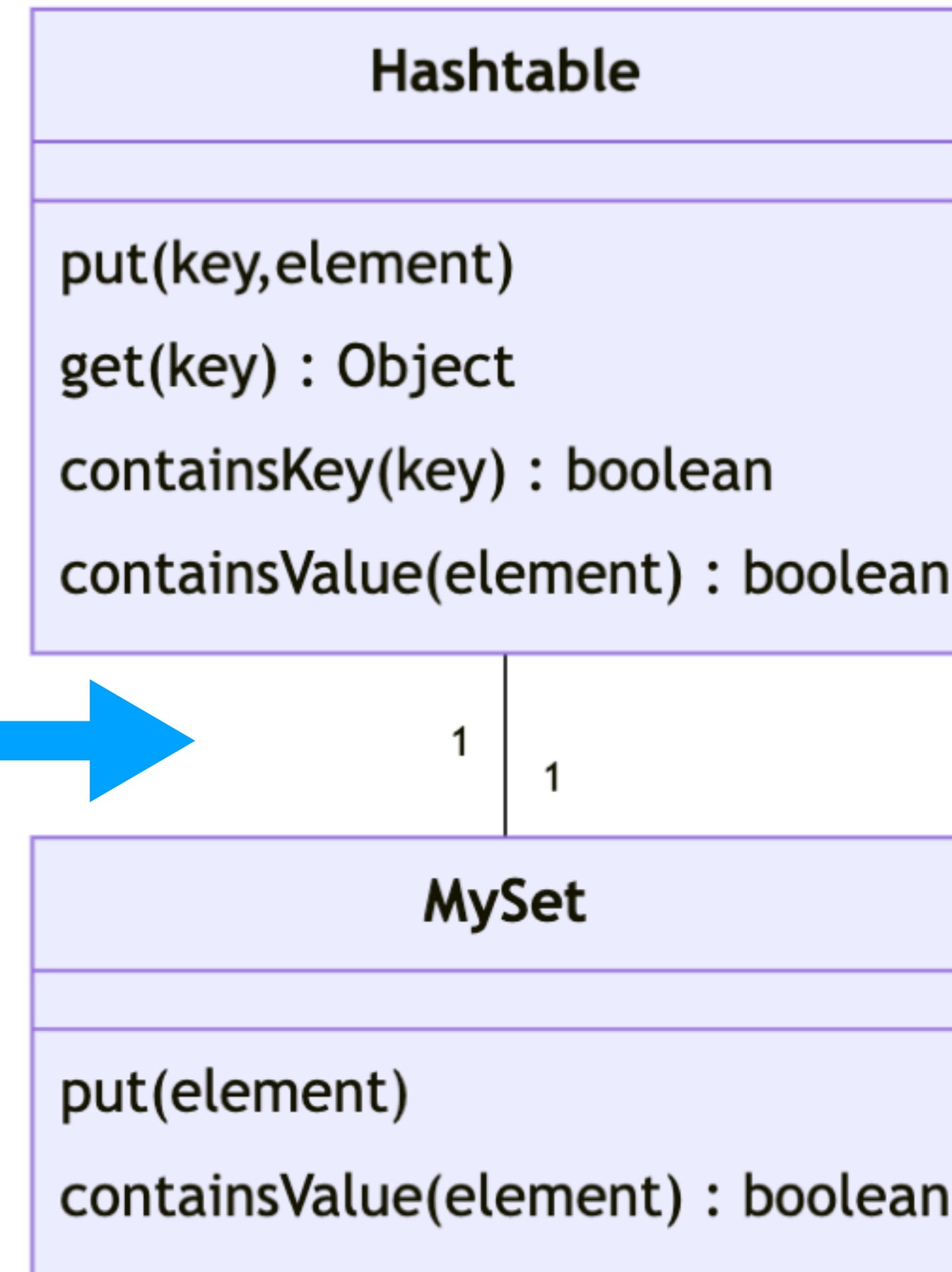


# Delegation and Implementation Inheritance

Object design model before transformation



Object design model after transformation



# Delegation and Implementation Inheritance

```
1 /* Implementation of MySet using inheritance */
2 class MySet extends Hashtable {
3     /* Constructor omitted */
4     MySet() {
5     }
6     void put(Object element) {
7         if (!containsKey(element)) {
8             put(element, this);
9         }
10    }
11    boolean containsValue (Object element){
12        return containsKey(element);
13    }
14    /* Other methods omitted */
15 }
```

```
1 /* Implementation of MySet using delegation */
2 class MySet {
3     private Hashtable table;
4     MySet() {
5         table = Hashtable();
6     }
7     void put(Object element) {
8         if (!containsValue(element)){
9             table.put(element, this);
10        }
11    }
12    boolean containsValue(Object element) {
13        return (table.containsKey(element));
14    }
15    /* Other methods omitted */
16 }
```

**Figure 8-3** An example of implementation inheritance. The left column depicts a questionable implementation of MySet using implementation inheritance. The right column depicts an improved implementation using delegation (UML class diagram and Java).