

# Towards Variability-aware Smells

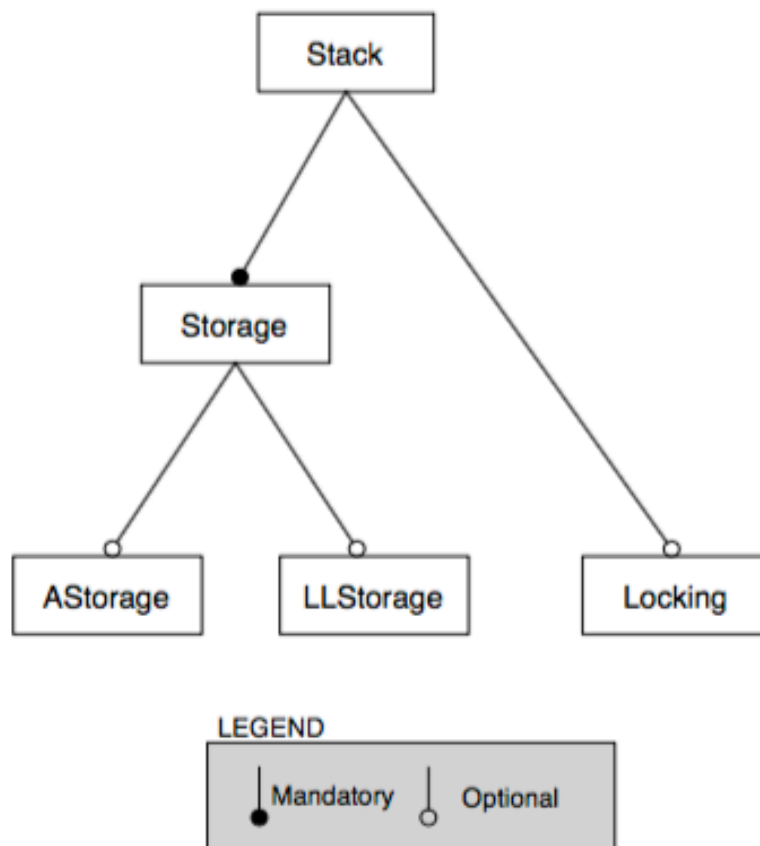
## Bad Annotation Name

### DEFINITION:

Bad Annotation name smell characterizes when the annotation tag name in source code does not correspond exactly to the feature name in the feature model.

### EXAMPLE:

Considering the feature model and the optional features *AStorage* and *LLStorage* and *Locking* in Figure 7.2, we can observe different feature-names in the annotation tag name in Listing 7.3, such as *AStoraging*, *LLStoraging*, and *Locker*.



**Figure 7.2** Bad annotation example.

**Listing 7.3** Bad annotation example.

```
1 class Stack <E> {
2   // #if ${AStoraging} == "T"
3   List<E> store = new ArrayList<E>();
4   // #endif
5   // #if ${LLStoraging} == "T"
6   List<E> store = new LinkedList<E>();
7   // #endif
8   // #if ${Locker} == "T"
9   public void push(E e, Lock lock) {
10    lock.lock() ;
11    store.add(e);
12
13    lock.unlock();
14  }
15  E pop(Lock lock) {
16    lock.lock() ;
17    try { return store.remove(store.size()-1); }
18    finally { lock.unlock(); }
19  }
20 }
21 // #endif
```

### **PROBLEM:**

Bad Annotation Name smell impacts negatively on product derivation and program comprehension because this smell makes difficult to identify features implementations (feature's tag) in the variability implementation source code.