

## Performance test 3

Method *containsAll(Collection c)* of class *java.util.AbstractCollection* called for values of a *org.-apache.commons.collections.map.MultiValueMap* is pretty slow! Imagine we did get a hint that this method runs way too slow!

### Task

Find the performance bug in method *containsAll(...)* concerning *MultiValueMap* and try to understand it.

### Problem is

The inner class *Values* of class *MultiValueMap* doesn't override the method *containsAll(...)* so that *AbstractCollection.containsAll(...)* will be called. This method iterates through all elements of the collection and calls *contains* for each element. The *contains(...)* method creates an iterator where in turn an iterator chain is being created. The creation of the iterator chain itself is also linear and spends a lot of time with constructing instances of class *ValuesIterator*. What actually happens is linear search with the iterator chain. Thus total asymptotic runtime is  $\mathcal{O}(n^3)$ .

### Solution

Override method *containsAll(...)* in class *MultiValueMap.Values*.

### Hints

1. The dynamic type of variable *values* in the performance test is not *AbstractCollection*.
2. The method *AbstractCollection.containsAll(...)* gets called because the class *MultiValueMap.Values* doesn't override the method *containsAll(...)*.
3. Try to override *containsAll(...)* with the help of a *HashSet*.