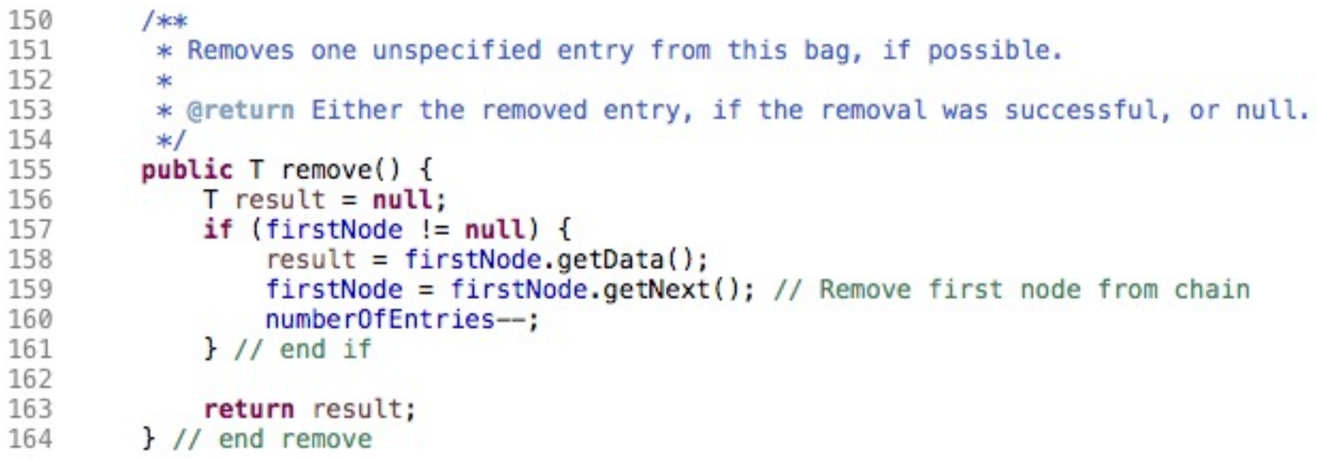
Removing

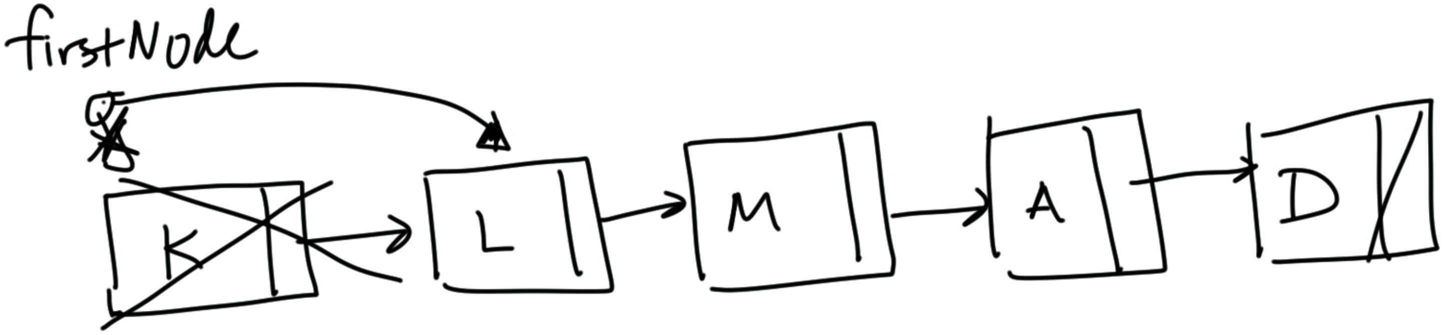
case 1: unspecified item
(easiest to just remove the fist entry)

case 2: specified item which may
be anywhere in the linked chain



```java
/**
 * Removes one unspecified entry from this bag, if possible.
 *
 * @return Either the removed entry, if the removal was successful, or null.
 */
public T remove() {
    T result = null;
    if (firstNode != null) {
        result = firstNode.getData();
        firstNode = firstNode.getNext(); // Remove first node from chain
        numberOfEntries--;
    } // end if

    return result;
} // end remove
```



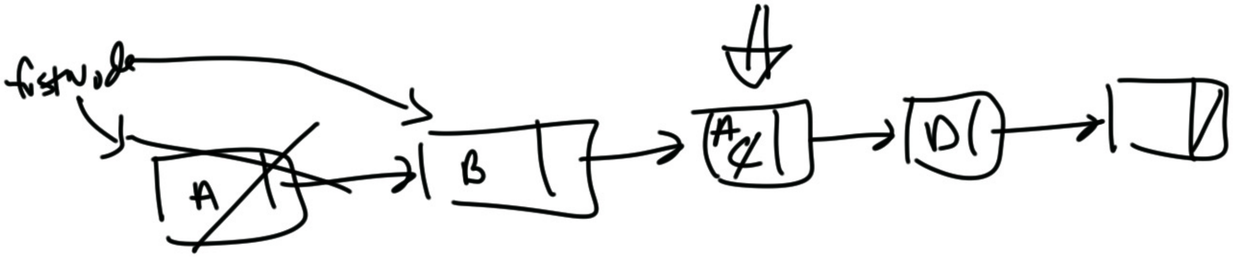
numberOfEntries	result
~~5~~	~~null~~
4	K

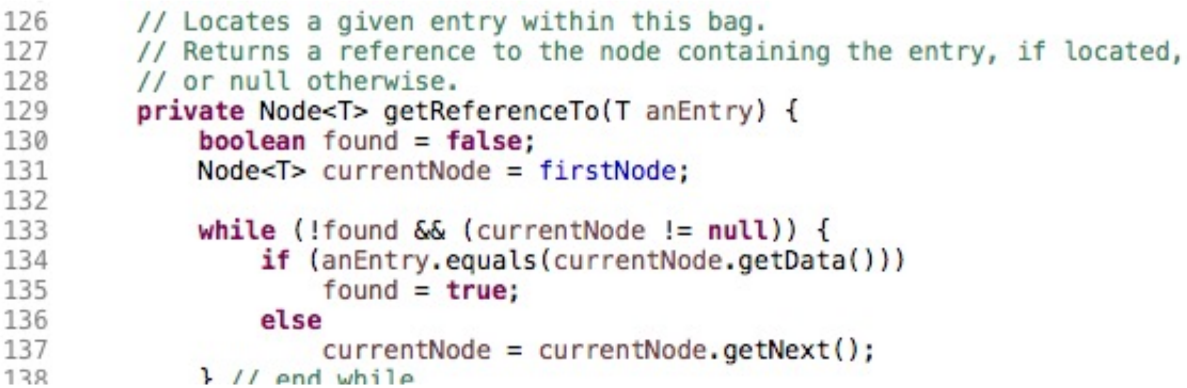
Removing Specified Item

1. Locate item and refernce its Node

2. Overwrite this data with data from the first node of the chain

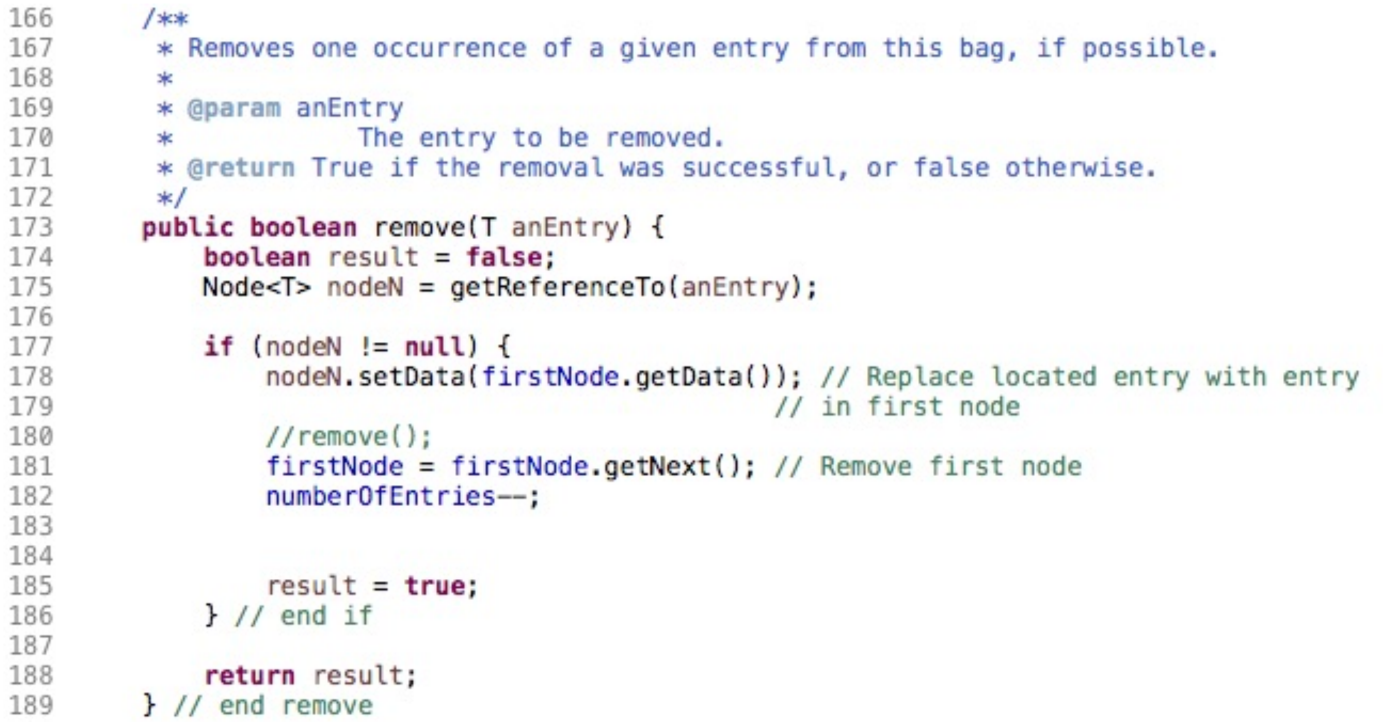
3. Remove the first node in the chain





```java
    // Locates a given entry within this bag.
    // Returns a reference to the node containing the entry, if located,
    // or null otherwise.
    private Node<T> getReferenceTo(T anEntry) {
        boolean found = false;
        Node<T> currentNode = firstNode;

        while (!found && (currentNode != null)) {
            if (anEntry.equals(currentNode.getData()))
                found = true;
            else
                currentNode = currentNode.getNext();
        } // end while
```

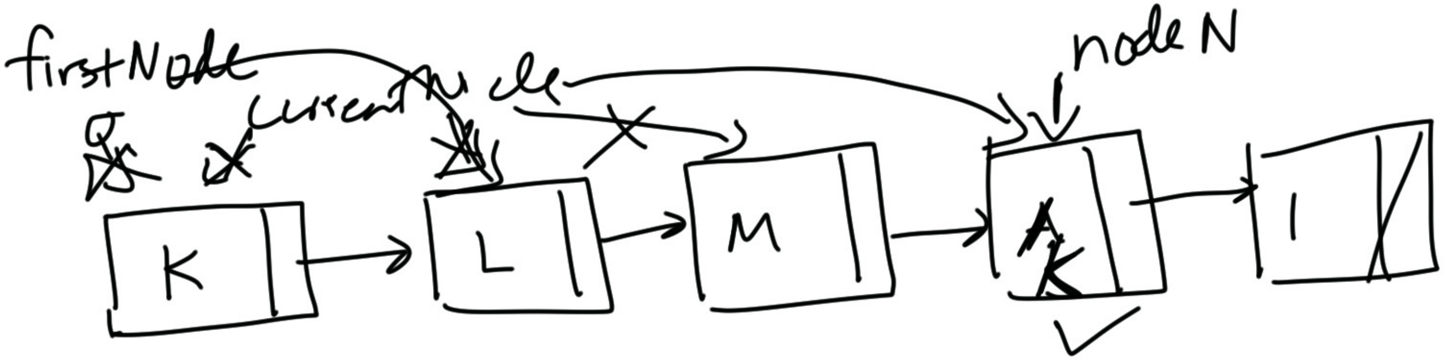


```java
/**
 * Removes one occurrence of a given entry from this bag, if possible.
 *
 * @param anEntry
 *            The entry to be removed.
 * @return True if the removal was successful, or false otherwise.
 */
public boolean remove(T anEntry) {
    boolean result = false;
    Node<T> nodeN = getReferenceTo(anEntry);

    if (nodeN != null) {
        nodeN.setData(firstNode.getData()); // Replace located entry with entry
                                         // in first node
        //remove();
        firstNode = firstNode.getNext(); // Remove first node
        numberOfEntries--;


        result = true;
    } // end if

    return result;
} // end remove
```



numberOfEntries

~~5~~ 4

anEntry

A

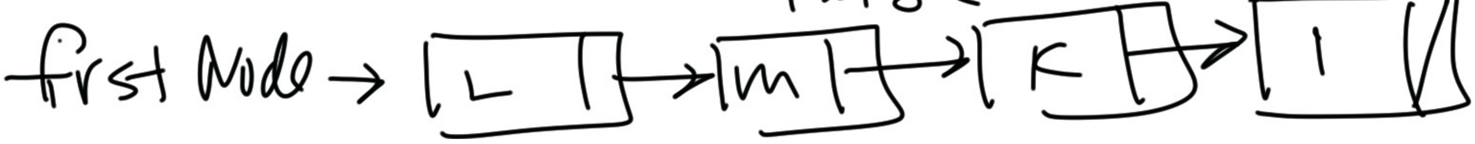
result

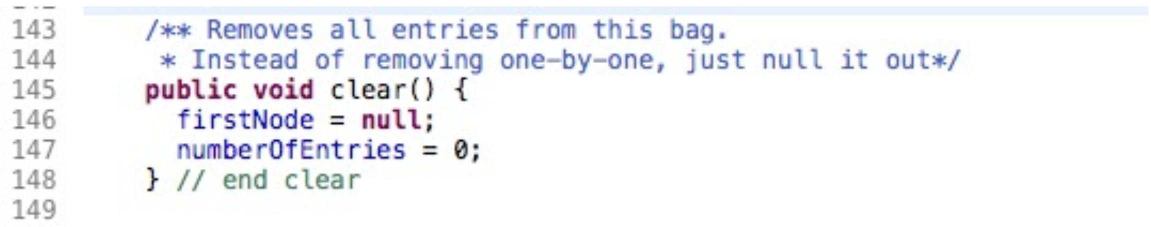
~~false~~
true

getReferenceTo

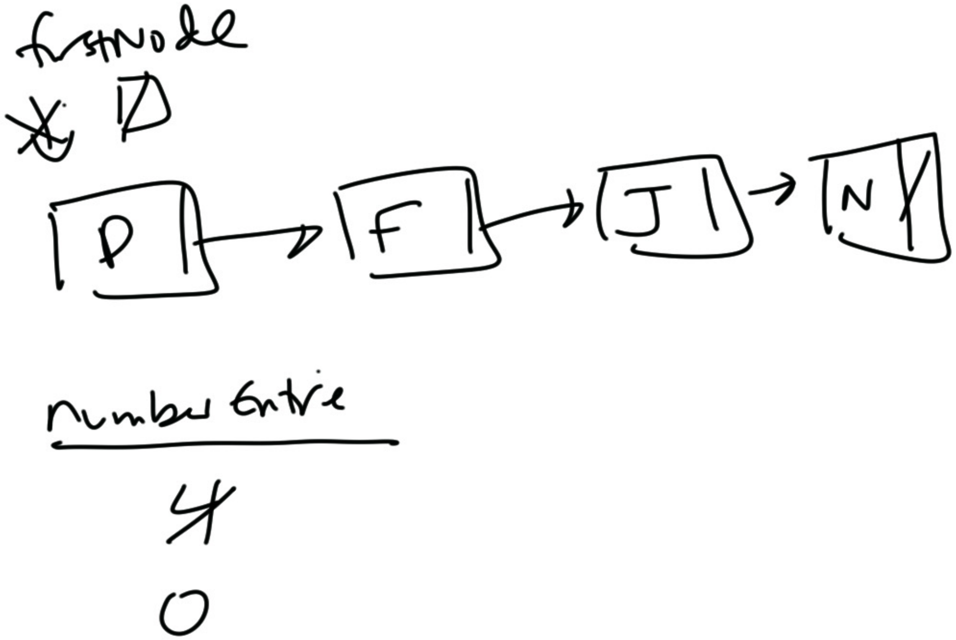
found

false





```
    /** Removes all entries from this bag.
     * Instead of removing one-by-one, just null it out*/
    public void clear() {
      firstNode = null;
      numberOfEntries = 0;
    } // end clear
```


firstNode
D
F
J
N
numberEntrie
~~4~~
0