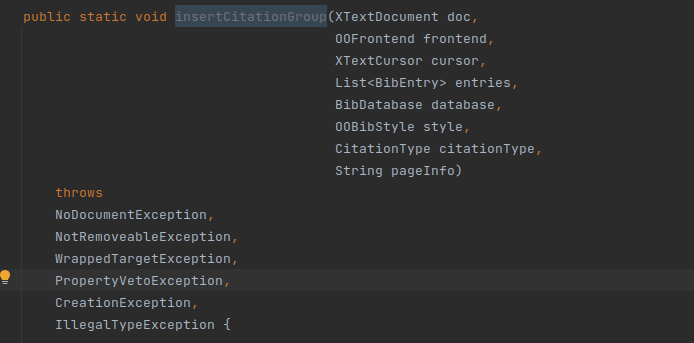
### **Code Smells**

src/main/java/org/jabref/logic/openoffice/action/EditInsert.java

line 59

8 parameters



src/main/java/org/jabref/logic/openoffice/action/EditMerge.java

line 156

Long Method

private static boolean checkAddToGroup(ScanState state, CitationGroup group, XTextRange currentRange) {  
  
 if (state.currentGroup.isEmpty()) {  
 return false;  
 }  
  
 Objects.*requireNonNull*(state.currentGroupCursor);  
 Objects.*requireNonNull*(state.cursorBetween);  
 Objects.*requireNonNull*(state.prev);  
 Objects.*requireNonNull*(state.prevRange);  
  
 // Only combine (Author 2000) type citations  
 if (group.citationType != CitationType.*AUTHORYEAR\_PAR*) {  
 return false;  
 }  
  
 if (state.prev != null) {  
  
 // Even if we combine AUTHORYEAR\_INTEXT citations, we would not mix them with AUTHORYEAR\_PAR  
 if (group.citationType != state.prev.citationType) {  
 return false;  
 }  
  
 if (!UnoTextRange.*comparables*(state.prevRange, currentRange)) {  
 return false;  
 }  
  
 // Sanity check: the current range should start later than the previous.  
 int textOrder = UnoTextRange.*compareStarts*(state.prevRange, currentRange);  
 if (textOrder != (-1)) {  
 String msg =  
 String.*format*("MergeCitationGroups:"  
 + " \"%s\" supposed to be followed by \"%s\","  
 + " but %s",  
 state.prevRange.getString(),  
 currentRange.getString(),  
 ((textOrder == 0)  
 ? "they start at the same position"  
 : "the start of the latter precedes the start of the first"));  
 *LOGGER*.warn(msg);  
 return false;  
 }  
 }  
  
 if (state.cursorBetween == null) {  
 return false;  
 }  
  
 Objects.*requireNonNull*(state.cursorBetween);  
 Objects.*requireNonNull*(state.currentGroupCursor);  
  
 // assume: currentGroupCursor.getEnd() == cursorBetween.getEnd()  
 if (UnoTextRange.*compareEnds*(state.cursorBetween, state.currentGroupCursor) != 0) {  
 *LOGGER*.warn("MergeCitationGroups: cursorBetween.end != currentGroupCursor.end");  
 throw new IllegalStateException("MergeCitationGroups failed");  
 }  
  
 /\*  
 \* Try to expand state.currentGroupCursor and state.cursorBetween by going right to reach  
 \* rangeStart.  
 \*/  
 XTextRange rangeStart = currentRange.getStart();  
 boolean couldExpand = true;  
 XTextCursor thisCharCursor =  
 (currentRange.getText().createTextCursorByRange(state.cursorBetween.getEnd()));  
  
 while (couldExpand && (UnoTextRange.*compareEnds*(state.cursorBetween, rangeStart) < 0)) {  
 //  
 // Check that we only walk through inline whitespace.  
 //  
 couldExpand = thisCharCursor.goRight((short) 1, true);  
 String thisChar = thisCharCursor.getString();  
 thisCharCursor.collapseToEnd();  
 if (thisChar.isEmpty() || "\n".equals(thisChar) || !thisChar.trim().isEmpty()) {  
 couldExpand = false;  
 if (!thisChar.isEmpty()) {  
 thisCharCursor.goLeft((short) 1, false);  
 }  
 break;  
 }  
 state.cursorBetween.goRight((short) 1, true);  
 state.currentGroupCursor.goRight((short) 1, true);  
  
 // These two should move in sync:  
 if (UnoTextRange.*compareEnds*(state.cursorBetween, state.currentGroupCursor) != 0) {  
 *LOGGER*.warn("MergeCitationGroups: cursorBetween.end != currentGroupCursor.end (during expand)");  
 throw new IllegalStateException("MergeCitationGroups failed");  
 }  
 }  
  
 return couldExpand;  
}

src/main/java/org/jabref/logic/openoffice/action/EditMerge.java

line 210

line 242

line 282

Shotgun Surgery

