

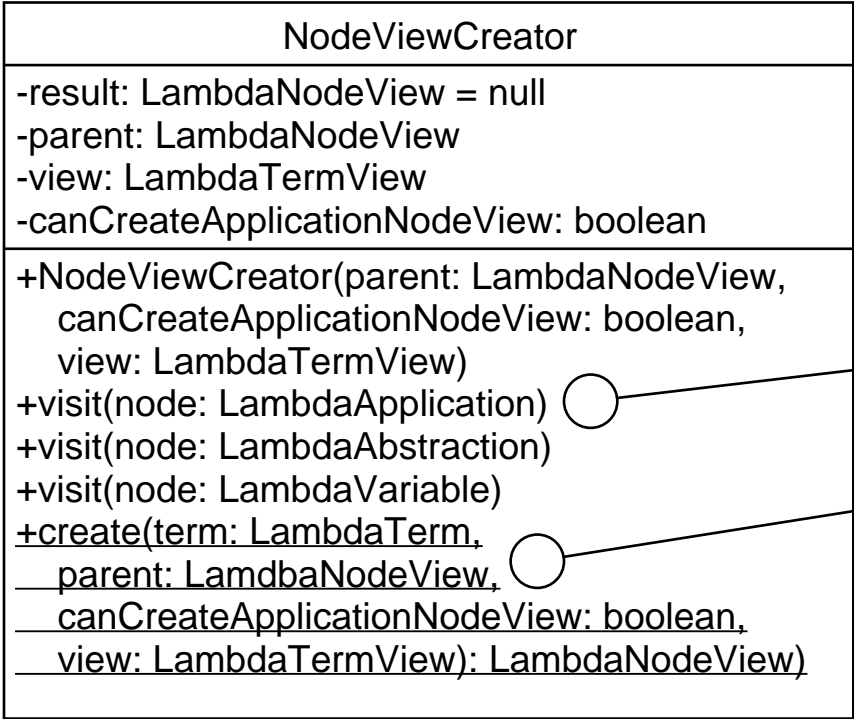
```
...  
lastVisited = inserted;
```

```
if (rightSibling == null && lastVisited == node.getFirstChild()) {  
    rightSibling = node.getSecondChild();  
}  
if (inserted == node.getSecondChild()) { // Parenthesis  
    isSecondApplicationChild = true;  
}  
if (view.hasNodeView(node))  
    insertChild(view.getActor(node));  
} else {  
    lastVisited = node;  
    node.getParent().accept(this);  
}
```

```
insertChild(view.getActor(node));
```

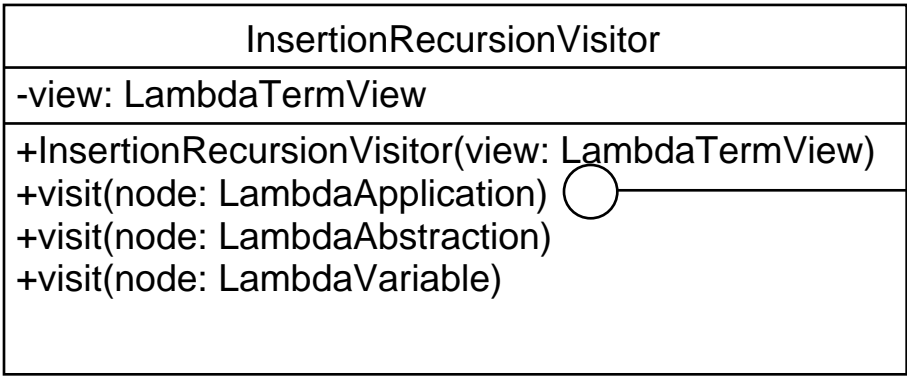
```
assert(false);
```

```
parent.insertChild(NodeViewCreator.create(inserted, parent, isSecondApplicationChild, view), rightSibling);  
inserted.accept(new InsertionRecursionVisitor(view));
```



```
if (canCreateApplicationNodeView) {  
    result = new LambdaParenthesisView(node, parent, view);  
}
```

```
NodeViewCreator visitor = new NodeViewCreator(parent, canCreateApplicationNodeView, view);  
term.accept(visitor);  
return visitor.result;
```



```
if (node.getFirstChild() != null) {  
    node.accept(new InsertionVisitor(node.getFirstChild(), view));  
}  
if (node.getSecondChild() != null) {  
    node.accept(new InsertionVisitor(node.getSecondChild(), view));  
}
```