Project 7: Making a Visitor using R3

Allyson Walpole (amw4363) and Hannah Mateja (hnm466)

R3 Script:

```
RPackage pkg = RProject.getPackage("RefactoringCrawler",
"edu.uiuc.detectRefactorings.detection");
RClassOrInterface vis = pkg.newClass("LikelinessVisitor");
RField instance = vis.addSingleton();
RClassOrInterface fd =  pkg.getClass("FieldDetection");
String[] sig = {"computeLikeliness", "edu.uiuc.detectRefactorings.util.Node",
"edu.uiuc.detectRefactorings.util.Node"};
RMethod comp = fd.getMethod(sig);
ArrayList<RMethod> relatives = comp.getRelatives();
for(RMethod rel : relatives)
{
        rel.addParameter(vis, "LikelinessVisitor.instance");
}
for(RMethod rel : relatives)
{
        RParameter del = rel.getParameter(2);
        rel.moveAndDelegate(del);
}
String[] sig2 = {"computeLikeliness", "edu.uiuc.detectRefactorings.util.Node",
"edu.uiuc.detectRefactorings.util.Node","edu.uiuc.detectRefactorings.detection.
LikelinessVisitor" };
comp = fd.getMethod(sig2);
relatives = comp.getRelatives();
for(RMethod rel : relatives)
{
        rel.rename("accept");
}
ArrayList<RMethod> visitingMethods = vis.getAllMethods();
for(RMethod visitM : visitingMethods)
{
        visitM.rename("visit");
}
```

Explanation of the script:

Firstly, we created a new class to be the Visitor, and named it LikelinessVisitor. Then, we got all the relatives for the computeLikeliness(Node, Node) function. We did this by getting the method from FieldDetection and calling getRealtives on that RMethod and putting th result into an ArrayList of RMethod. Then, for each relative, we added a LikelinessVisitor parameter to the method, then moved the method to the Visitor via the parameter delegate. After that, we had to get the new relatives, since the signature had changed. With this list of relatives, we renamed the computeLikeliness method to accept. Finally, we renamed all of the functions within LikelinessVisitor to visit().