

## Project 7: Undoing a Visitor using R3

Allyson Walpole (amw4363) and Hannah Mateja (hnm466)

R3 Script:

```
RPackage pkg =
RProject.getPackage("RefactoringCrawlerWithVisitor",
"edu.uiuc.detectRefactorings.detection");
RClassOrInterface vis = pkg.getClass("Visitor");
ArrayList<RMethod> visits = vis.getAllMethods();
for(RMethod v : visits)
{
    RParameter del = v.getParameter(0);
    v.move(del);
}
RClassOrInterface fd = pkg.getClass("FieldDetection");
String[] comp = {"visit",
"edu.uiuc.detectRefactorings.detection.Visitor"};
RMethod newVisit = fd.getMethod(comp);
ArrayList<RMethod> newVisits = newVisit.getRelatives();
for(RMethod nv : newVisits)
{
    nv.replace("accept");
}
String[] comp2 = {"accept",
"edu.uiuc.detectRefactorings.detection.Visitor"};
RMethod accept = fd.getMethod(comp2);
ArrayList<RMethod> accepts = accept.getRelatives();
for(RMethod acc : accepts)
{
    RParameter visParam = acc.getParameter(0);
    acc.removeParameter(visParam);
}
String[] comp3 = {"accept"};
accept = fd.getMethod(comp3);
accepts = accept.getRelatives();
for(RMethod acc : accepts)
{
    acc.rename("isRename");
}
vis.delete();
```

Explanation of script:

First, we get the Visitor class and all of its visit() methods, and we move those visit() methods to the detection class it belongs to. This move does not need a delegate, as the Visitor is going to be deleted

anyway. Then, we get the newly placed visit() functions with the detection classes, and in-lined the visit() method into the accept() methods using replace(). Afterwards, we get all of the accept() functions and remove the Visitor parameter. Then, we rename the accept() methods to isRename(). Lastly, we remove the Visitor class.