

ModelHierarchyVisitor

The `com.nomagic.uml2.impl.ModelHierarchyVisitor` is an enhanced implementation of the *Visitor* pattern and a subclass of `com.nomagic.uml2.impl.ModelVisitor`, used for visiting model elements.

Every `visit<element_metatype>` method calls a `visit<element_metatype>` method for a super type.

Model elements `visit...` methods have an additional context parameter of the `com.nomagic.uml2.ext.jmi.reflect.VisitorContext` type. The visitor context is used to track what super type `visit<element_metatype>` methods were already visited (to avoid multiple visits because some model elements have a multiple inheritance). Open API users should not work with the visitor context. All tracking is done in internal implementation.

For example, you can put some code into `com.nomagic.uml2.impl.ModelHierarchyVisitor.visitClassifier(Classifier)` and it will be executed for all subclasses of a `com.nomagic.uml2.ext.magicdraw.classes.mdkernel.Classifier`.

```
ModelVisitor myVisitor = new ModelHierarchyVisitor()
{
    public void visitClassifier(Classifier element, VisitorContext
context)
    {
        //this is my classifier
    }
};
Project project = ...;
Package root = project.getPrimaryModel();
Iterator<Element> it = root.getOwnedElement().iterator();
while (it.hasNext())
{
    it.next().accept(myVisitor);
}
```



You can find the code examples in `<installation_directory>\openapi\examples\statistics`