

Visitors

Visitor design pattern is implemented in UML model structure. Every *com.nomagic.uml2.ext.magicdraw.classes.mdkernel.Element* has the *accept(AbstractVisitor)* method for visiting it.

The *com.nomagic.uml2.impl.ModelVisitor* has the *visit<element_metatype>* method for all types of model elements. This is very useful when you are working with a large collection of elements and need to perform actions, specific for every type of *Element* (for example, save/load, copy/paste, or a specific properties setting).

Just derive your class from *com.nomagic.magicdraw.uml.InheritanceVisitor* and override some *visit...* methods and call *accept(AbstractVisitor)* for *Elements*.

```
ModelVisitor myVisitor = new ModelVisitor()
{
    // override some visit methods ...
    public void visitClass(Class element, VisitorContext context)
    {
        //this is my UML Class
    }
};
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*