Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

***«Московский государственный технический университет имени Н.Э. Баумана»   
(МГТУ им. Н.Э. Баумана)***

***Кафедра ИУ3, 3 курс, 6 семестр.***

**Отчет по лабораторной работе №5**

**“** [**Расширение Eclipse своими плагинами**](http://www.agentlab.ru/confluence/pages/viewpage.action?pageId=55771143)**”**

**по курсу «Разработка ПО»**

Выполнил: Шорох Д.К.

Группа: ИУ3-62

Преподаватель: Иванов А.М.

Москва 2014

# Цель задания

Цель задания: познакомит студентов с технологией расширения Eclipse своими плагинами.

# Задание

1. 1.Создайте расширение для Eclipse используя фигурку полученную в процессе лабораторной работы №2.
2. Откройте Eclipse и обновите локальную копию репозитария
3. В перспективе Git, надо сделать Pull
4. Переключитесь на стандартную целевую платформу Eclipse
5. Настройки Eclipse > Plug-in Development > Target Platform, установить **Running Platform**
6. Импортируйте в Workspace проекты:

* **ru.agentlab.jfxed**
* **ru.agentlab.jfxed.figures**
* **ru.agentlab.jfxed.figures.class**

1. Запустите редактор

## Реализация:

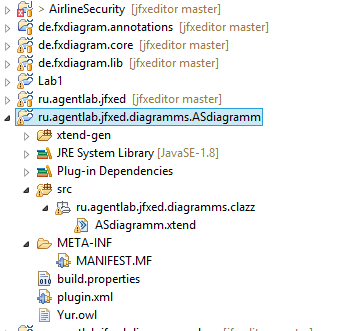


Рис.1. Структура папок.

Файл ASdiagramm.xtend

**package** ru.agentlab.jfxed.diagramms.clazz

**import** com.hp.hpl.jena.ontology.OntModel

**import** com.hp.hpl.jena.query.QueryExecutionFactory

**import** com.hp.hpl.jena.query.QueryFactory

**import** com.hp.hpl.jena.query.QuerySolution

**import** com.hp.hpl.jena.rdf.model.Resource

**import** de.fxdiagram.core.XDiagram

**import** ru.agentlab.jfxed.IDiagram

**import** ru.agentlab.jfxed.figures.requirement.ClassFigure

**public** **class** ASdiagramm **implements** IDiagram {

**static** String *SOURCE* = "http://www.agentlab.ru/jfxed/onto/strategicrationale"

**static** String *NS* = *SOURCE* + "#"

**override** createJfx(OntModel jenaModel, XDiagram jfxDiagram) {

**val** queryString ='''

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

select ?uri

where {

?uri rdf:type <«*SOURCE*»#Concept>

}

'''

**val** query = QueryFactory.*create*(queryString)

// Execute the query and obtain results

**val** qe = QueryExecutionFactory.*create*(query, jenaModel)

**val** results = qe.execSelect()

**for** ( ; results.hasNext() ; )

{

**val** QuerySolution soln = results.nextSolution()

**val** Resource x = soln.getResource("uri") // Get a result variable by name.

**val** target = **new** ClassFigure() => [

layoutX = 280

layoutY = 280

name = x.localName

]

jfxDiagram => [

nodes += target

]

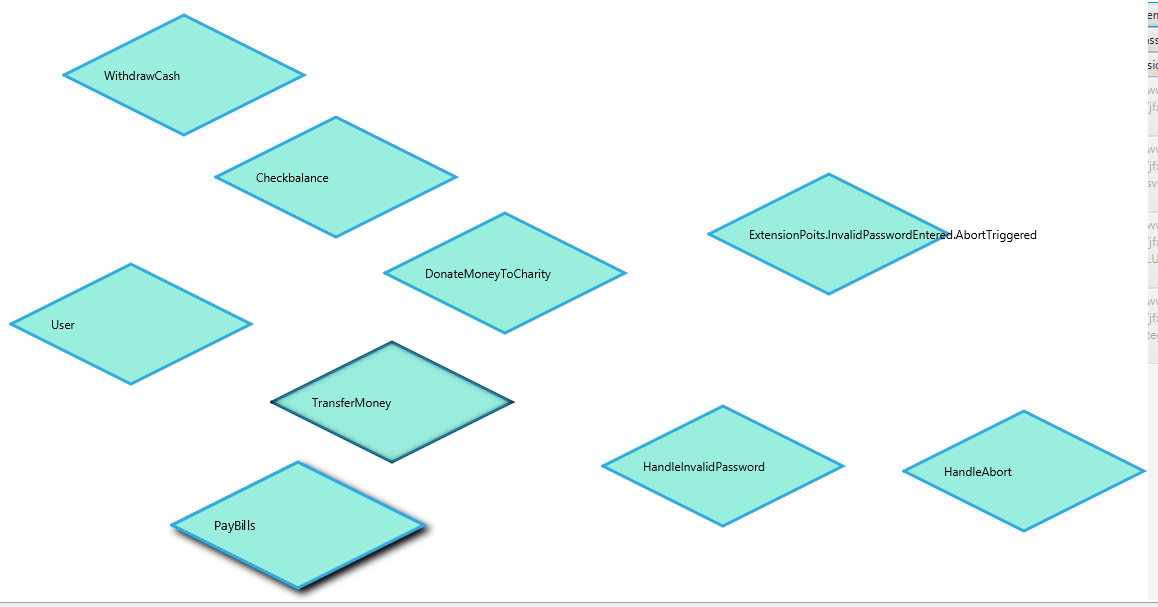
}

qe.close()

}

}

## Результат тестирования:

****

**Рис.2 Диаграмма**