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 №3**

**по курсу «Разработка ПО»**

Выполнил:

Малых Д.А.

Группа:

ИУ3-61

Проверил:

Иванов А. М.

Москва 2014

## Задание

Создать тестовый Eclipse-проект, в котором происходит формирование информационной модели данных диаграммы и работа с данными в соответствии со своим вариантом задания и требованиями.

Добавить созданные проект в репозитарий системы контроля версий.

## Цель работы

* Углубление навыков работы с системой контроля версий
* Ознакомление на практике с основами графовых БД и технологий Semantic Web

## Картинка - задание

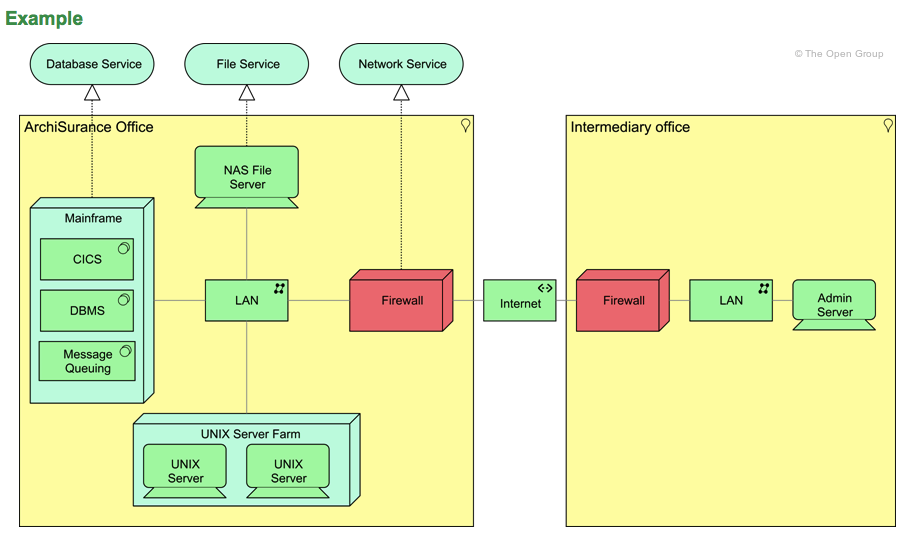


Рисунок 1. Картинка-задание

1. **Исходный код программы**
2. **package JenaPruebas;**
3. **import java.io.FileWriter;**
4. **import java.io.IOException;**
5. **import com.hp.hpl.jena.ontology.Individual;**
6. **import com.hp.hpl.jena.ontology.ObjectProperty;**
7. **import com.hp.hpl.jena.ontology.OntClass;**
8. **import com.hp.hpl.jena.ontology.OntModel;**
9. **import com.hp.hpl.jena.rdf.model.ModelFactory;**
10. **import com.hp.hpl.jena.rdf.model.SimpleSelector;**
11. **import com.hp.hpl.jena.rdf.model.StmtIterator;**

14. **public class LayeredViewPoint {**
15. **static final String SCHEMA = "http://www.agentlab.ru/jfxed/onto/strategicrationale";**
16. **static final String NS = SCHEMA + "#";**
18. **public static void main (String[] args) {**
19. **OntModel m = ModelFactory.createOntologyModel();**
21. **OntClass concept = m.createClass(NS + "Concept");**
22. **OntClass relation = m.createClass(NS + "Relation");**


26. **//объекты**
27. **OntClass redRectClass = m.createClass(NS + "redRect");**
28. **redRectClass.addSuperClass(concept);**
29. **OntClass greenComputer = m.createClass(NS + "greenComputer");**
30. **greenComputer.addSuperClass(concept);**
31. **OntClass greenRectClass = m.createClass(NS + "greenRect");**
32. **greenRectClass.addSuperClass(concept);**
33. **OntClass greenOvalClass = m.createClass(NS + "greenOval");**
34. **greenOvalClass.addSuperClass(concept);**
35. **OntClass bigGreenRectClass = m.createClass(NS + "bigGreenRect");**
36. **bigGreenRectClass.addSuperClass(concept);**
38. **//связи**
39. **OntClass oneWayArrowClass = m.createClass(NS + "oneWayArrow");**
40. **oneWayArrowClass.addSuperClass(relation);**
41. **OntClass oneWayLinkClass = m.createClass(NS + "oneWayLink");**
42. **oneWayLinkClass.addSuperClass(relation);**


46. **//свойства стрелки**
47. **ObjectProperty propOneWayArrow = m.createObjectProperty(NS + "OneWayArrow");**
48. **propOneWayArrow.addDomain(bigGreenRectClass);**
49. **propOneWayArrow.addRange(greenOvalClass);**
50. **ObjectProperty propOneWayArrow1 = m.createObjectProperty(NS + "OneWayArrow1");**
51. **propOneWayArrow1.addDomain(greenComputer);**
52. **propOneWayArrow1.addRange(greenOvalClass);**
53. **ObjectProperty propOneWayArrow2 = m.createObjectProperty(NS + "OneWayArrow2");**
54. **propOneWayArrow2.addDomain(redRectClass);**
55. **propOneWayArrow2.addRange(greenOvalClass);**
57. **//свойства линии**
58. **ObjectProperty propOneWayLink = m.createObjectProperty(NS + "OneWayLink");**
59. **propOneWayLink.addDomain(bigGreenRectClass);**
60. **propOneWayLink.addRange(greenRectClass);**
61. **ObjectProperty propOneWayLink1 = m.createObjectProperty(NS + "OneWayLink");**
62. **propOneWayLink.addDomain(greenRectClass);**
63. **propOneWayLink.addRange(greenComputer);**
64. **ObjectProperty propOneWayLink2 = m.createObjectProperty(NS + "OneWayLink");**
65. **propOneWayLink.addDomain(greenRectClass);**
66. **propOneWayLink.addRange(redRectClass);**



71. **Individual mainframe = m.createIndividual(NS + "mainframe", bigGreenRectClass);**
72. **Individual unixServer = m.createIndividual(NS + "UNIXServerfarm", bigGreenRectClass);**
73. **Individual lan1 = m.createIndividual(NS + "LAN", greenRectClass);**
74. **Individual lan2 = m.createIndividual(NS + "LAN", greenRectClass);**
75. **Individual internet = m.createIndividual(NS + "internet", greenRectClass);**
76. **Individual databaseServer = m.createIndividual(NS + "DatabaseServer", greenOvalClass);**
77. **Individual fileService = m.createIndividual(NS + "FileService", greenOvalClass);**
78. **Individual networkService = m.createIndividual(NS + "NetworkService", greenOvalClass);**
79. **Individual firewall1 = m.createIndividual(NS + "firewall", redRectClass);**
80. **Individual firewall2 = m.createIndividual(NS + "firewall", redRectClass);**
81. **Individual nasFileServer = m.createIndividual(NS + "NASFileServer", greenComputer);**
82. **Individual adminServer = m.createIndividual(NS + "AdminServer", greenComputer);**

85. **Individual link1 = m.createIndividual(NS + "mainframe", oneWayArrowClass);**
86. **link1.addProperty(propOneWayArrow, databaseServer);**
88. **Individual link2 = m.createIndividual(NS + "NASFileServer", oneWayArrowClass);**
89. **link1.addProperty(propOneWayArrow1, fileService);**
91. **Individual link3 = m.createIndividual(NS + "firewall1", oneWayArrowClass);**
92. **link1.addProperty(propOneWayArrow2, networkService);**
94. **Individual lane1 = m.createIndividual(NS + "mainframe1", oneWayLinkClass);**
95. **link1.addProperty(propOneWayArrow, lan1);**
97. **Individual lane2 = m.createIndividual(NS + "lan1", oneWayLinkClass);**
98. **link1.addProperty(propOneWayArrow1, nasFileServer);**
100. **Individual lane3 = m.createIndividual(NS + "unixServerFarm", oneWayLinkClass);**
101. **link1.addProperty(propOneWayArrow, nasFileServer);**
103. **Individual lane4 = m.createIndividual(NS + "unixServerFarm1", oneWayLinkClass);**
104. **link1.addProperty(propOneWayArrow2, firewall1);**
106. **Individual lane5 = m.createIndividual(NS + "unixServerFarm2", oneWayLinkClass);**
107. **link1.addProperty(propOneWayArrow2, internet);**
109. **Individual lane6 = m.createIndividual(NS + "unixServerFarm3", oneWayLinkClass);**
110. **link1.addProperty(propOneWayArrow2, firewall2);**
112. **Individual lane7 = m.createIndividual(NS + "unixServerFarm4", oneWayLinkClass);**
113. **link1.addProperty(propOneWayArrow1, adminServer);**
115. **System.out.println("Request1\n");**
116. **for(StmtIterator it = link1.listProperties(); it.hasNext();)**
117. **{**
118. **System.out.println(it.nextStatement());**
119. **}**
120. **System.out.println("Request2\n");**
121. **for(StmtIterator it = lan1.listProperties(); it.hasNext();)**
122. **{**
123. **System.out.println(it.nextStatement());**
124. **}**
125. **System.out.println("Request3\n");**
126. **//вывод всего, что св€зано с insurer дл€ всей модели**
127. **OntClass c = mainframe.getOntClass();**
128. **SimpleSelector s = new SimpleSelector(null, null, c);**
129. **for(StmtIterator it = m.listStatements(s); it.hasNext();)**
130. **{**
131. **System.out.println(it.nextStatement());**
132. **}**

135. **try {**
136. **m.write(new FileWriter("Dm.owl"), "RDF/XML");**
137. **} catch (IOException e) {**
138. **e.printStackTrace();**
139. **}**
141. **}**
142. **}**

**Результат**

