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:

Учанев И.Л.

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

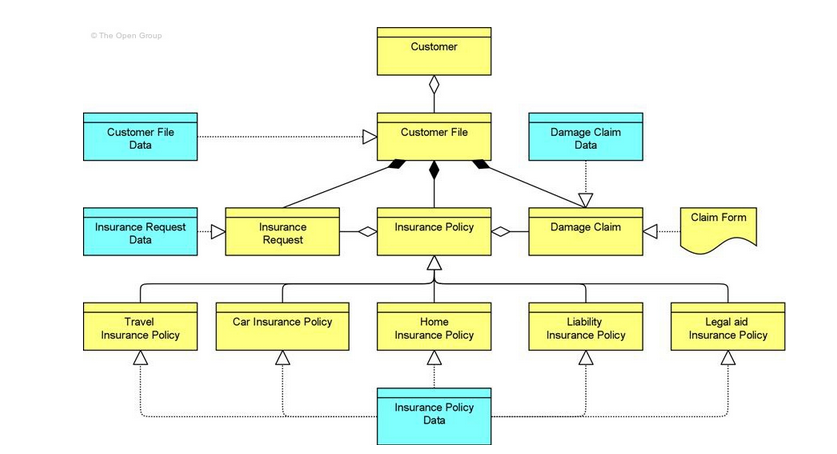


Рисунок 1. Картинка-задание

1. **Исходный код программы**

**package** ru.agentlab.jfxed.diagramms.InformationStructureViewpoint;

**import** java.io.FileWriter;

**import** java.io.IOException;

**import** com.hp.hpl.jena.ontology.Individual;

**import** com.hp.hpl.jena.ontology.ObjectProperty;

**import** com.hp.hpl.jena.ontology.OntClass;

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

**import** com.hp.hpl.jena.rdf.model.ModelFactory;

**import** com.hp.hpl.jena.rdf.model.SimpleSelector;

**import** com.hp.hpl.jena.rdf.model.StmtIterator;

**public** **class** lab {

**static** **final** String ***SCHEMA*** = "http://www.agentlab.ru/jfxed/onto/strategicrationale";

**static** **final** String ***NS*** = ***SCHEMA*** + "#";

**public** **static** **void** main (String[] args) {

OntModel m = ModelFactory.*createOntologyModel*();

//create superclasses

OntClass concept = m.createClass(***NS*** + "Concept");

OntClass relation = m.createClass(***NS*** + "Relation");

//create objects

OntClass RepresentationClass = m.createClass(***NS*** + "Representation");

RepresentationClass.addSuperClass(concept);

OntClass BusinessObjectClass = m.createClass(***NS*** + "BusinessObject");

BusinessObjectClass.addSuperClass(concept);

OntClass DataObjectClass = m.createClass(***NS*** + "DataObject");

DataObjectClass.addSuperClass(concept);

//create arrows

OntClass AggregationClass = m.createClass(***NS*** + "Aggregation");

AggregationClass.addSuperClass(relation);

OntClass CompositionClass = m.createClass(***NS*** + "Composition");

CompositionClass.addSuperClass(relation);

OntClass SpecializationClass = m.createClass(***NS*** + "Specialization");

SpecializationClass.addSuperClass(relation);

OntClass RealizationClass = m.createClass(***NS*** + "Realization");

RealizationClass.addSuperClass(relation);

//create properties

//whiteRhombus

ObjectProperty propAggregation = m.createObjectProperty(***NS*** + "Aggregation");

propAggregation.addDomain(BusinessObjectClass);

propAggregation.addRange(BusinessObjectClass);

//blackRhombus

ObjectProperty propComposition = m.createObjectProperty(***NS*** + "Composition");

propComposition.addDomain(BusinessObjectClass);

propComposition.addRange(BusinessObjectClass);

//Specialization

ObjectProperty propSpecialization = m.createObjectProperty(***NS*** + "Specialization");

propSpecialization.addDomain(BusinessObjectClass);

propSpecialization.addRange(BusinessObjectClass);

//Realization

ObjectProperty propRealization = m.createObjectProperty(***NS*** + "Realization");

propRealization.addDomain(DataObjectClass);

propRealization.addRange(BusinessObjectClass);

ObjectProperty propRealization1 = m.createObjectProperty(***NS*** + "Realization");

propRealization.addDomain(RepresentationClass);

propRealization.addRange(BusinessObjectClass);

//create individuals

Individual customer = m.createIndividual(***NS*** + "customer", BusinessObjectClass);

Individual customerFile = m.createIndividual(***NS*** + "customerFile", BusinessObjectClass);

Individual customerFileData = m.createIndividual(***NS*** + "customerFileData", DataObjectClass);

Individual damageClaimData = m.createIndividual(***NS*** + "damageClaimData", DataObjectClass);

Individual insuranceRequestData = m.createIndividual(***NS*** + "insuranceRequestData", DataObjectClass);

Individual insuranceRequest = m.createIndividual(***NS*** + "insuranceRequest", BusinessObjectClass);

Individual insurancePolicy = m.createIndividual(***NS*** + "insurancePolicy", BusinessObjectClass);

Individual damageClaim = m.createIndividual(***NS*** + "damageClaim", BusinessObjectClass);

Individual claimForm = m.createIndividual(***NS*** + "claimForm", RepresentationClass);

Individual travelInsurancePolicy = m.createIndividual(***NS*** + "travelInsurancePolicy", BusinessObjectClass);

Individual carInsurancePolicy = m.createIndividual(***NS*** + "carInsurancePolicy", BusinessObjectClass);

Individual homeInsurancePolicy = m.createIndividual(***NS*** + "homeInsurancePolicy", BusinessObjectClass);

Individual liabilityInsurancePolicy = m.createIndividual(***NS*** + "liabilityInsurancePolicy", BusinessObjectClass);

Individual legalAidInsurancePolicy = m.createIndividual(***NS*** + "legalAidInsurancePolicy", BusinessObjectClass);

Individual insurancePolicyData = m.createIndividual(***NS*** + "insurancePolicyData", DataObjectClass);

//create ind links

Individual link11 = m.createIndividual(***NS*** + "customerFile", AggregationClass);

link11.addProperty(propAggregation, customer);

Individual link12 = m.createIndividual(***NS*** + "customerFileData", RealizationClass);

link12.addProperty(propRealization, customerFile);

Individual link13 = m.createIndividual(***NS*** + "damageClaimData", RealizationClass);

link13.addProperty(propRealization, damageClaim);

Individual link21 = m.createIndividual(***NS*** + "insuranceRequestData", RealizationClass);

link21.addProperty(propRealization, insuranceRequest);

Individual link22 = m.createIndividual(***NS*** + "insuranceRequest", AggregationClass);

link22.addProperty(propAggregation, insurancePolicy);

Individual link23 = m.createIndividual(***NS*** + "premiumPayment", CompositionClass);

link23.addProperty(propComposition, customerFile);

Individual link24 = m.createIndividual(***NS*** + "insurancePolicy", CompositionClass);

link24.addProperty(propComposition, customerFile);

Individual link25 = m.createIndividual(***NS*** + "damageClaim", CompositionClass);

link25.addProperty(propComposition, customerFile);

Individual link26 = m.createIndividual(***NS*** + "damageClaim", AggregationClass);

link22.addProperty(propAggregation, insurancePolicy);

Individual link27 = m.createIndividual(***NS*** + "claimForm", RealizationClass);

link27.addProperty(propRealization1, damageClaim);

Individual link31 = m.createIndividual(***NS*** + "travelInsurancePolicy", SpecializationClass);

link31.addProperty(propSpecialization, insurancePolicy);

Individual link32 = m.createIndividual(***NS*** + "carInsurancePolicy", SpecializationClass);

link32.addProperty(propSpecialization, insurancePolicy);

Individual link33 = m.createIndividual(***NS*** + "homeInsurancePolicy", SpecializationClass);

link33.addProperty(propSpecialization, insurancePolicy);

Individual link34 = m.createIndividual(***NS*** + "liabilityInsurancePolicy", SpecializationClass);

link34.addProperty(propSpecialization, insurancePolicy);

Individual link35 = m.createIndividual(***NS*** + "legalAidInsurancePolicy", SpecializationClass);

link35.addProperty(propSpecialization, insurancePolicy);

Individual link41 = m.createIndividual(***NS*** + "insurancePolicyData", RealizationClass);

link41.addProperty(propRealization, travelInsurancePolicy);

Individual link42 = m.createIndividual(***NS*** + "insurancePolicyData", RealizationClass);

link42.addProperty(propRealization, carInsurancePolicy);

Individual link43 = m.createIndividual(***NS*** + "insurancePolicyData", RealizationClass);

link43.addProperty(propRealization, homeInsurancePolicy);

Individual link44 = m.createIndividual(***NS*** + "insurancePolicyData", RealizationClass);

link44.addProperty(propRealization, liabilityInsurancePolicy);

Individual link45 = m.createIndividual(***NS*** + "insurancePolicyData", RealizationClass);

link45.addProperty(propRealization, legalAidInsurancePolicy);

System.***out***.println("Request1\n");

**for**(StmtIterator it = link11.listProperties(); it.hasNext();)

{

System.***out***.println(it.nextStatement());

}

System.***out***.println("Request2\n");

**for**(StmtIterator it = damageClaimData.listProperties(); it.hasNext();)

{

System.***out***.println(it.nextStatement());

}

System.***out***.println("Request3\n");

OntClass c = claimForm.getOntClass();

SimpleSelector s = **new** SimpleSelector(**null**, **null**, c);

**for**(StmtIterator it = m.listStatements(s); it.hasNext();)

{

System.***out***.println(it.nextStatement());

}

**try** {

m.write(**new** FileWriter("Igor.owl"), "RDF/XML");

} **catch** (IOException e) {

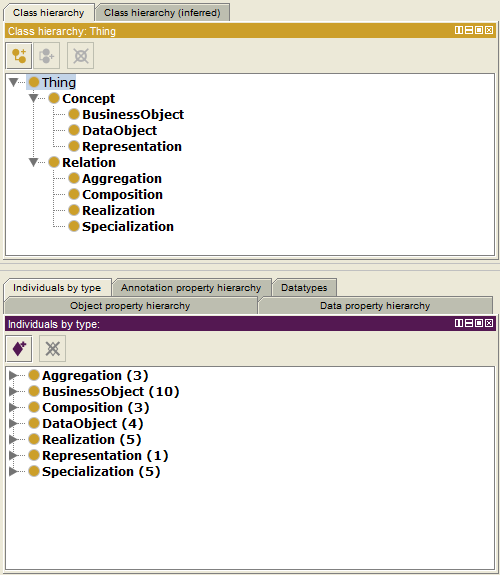
e.printStackTrace();

}

}

}

1. **Результат**

****

