Федеральное государственное бюджетное образовательное учреждение   
высшего образования

|  |  |
| --- | --- |
| Gerb-BMSTU_01 | «Московский государственный технический университет  им. Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана) |

ФАКУЛЬТЕТ – Информатика и управления

КАФЕДРА – Информационные системы и телекоммуникации

РАСЧЁТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе на тему

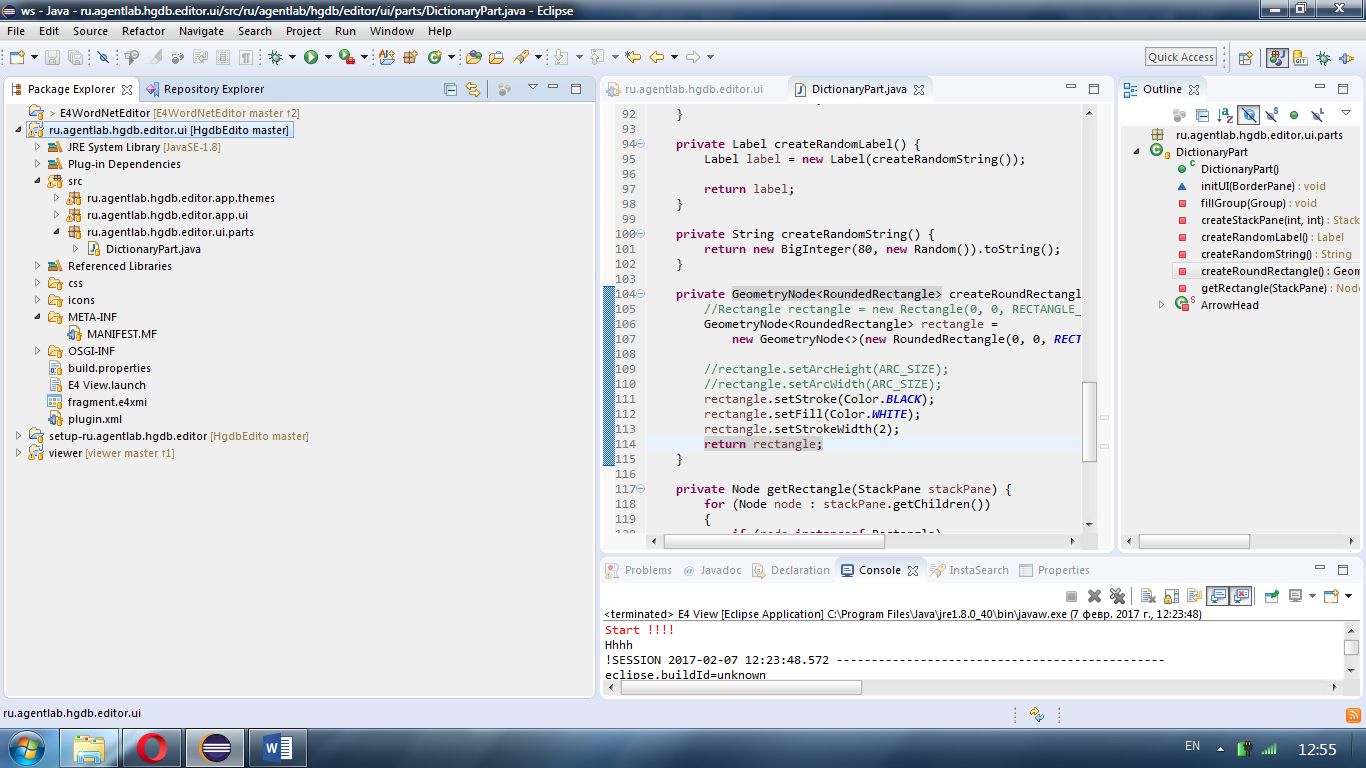
«Интеграция редактора HypergraphDB в Eclipse IDE»

Студент группы ИУ3-73 (подпись) 07.02.2017 М.К. Быковский

Руководитель курсовой работы (подпись) 07.02.2017 А. М. Иванов

Москва, 2017

Структура:



Код разработанной программы:

Класс рендеринга формы DictionaryPart.java:

**package** ru.agentlab.hgdb.editor.ui.parts;

**import** java.math.BigInteger;

**import** java.util.Random;

**import** javax.annotation.PostConstruct;

**import** org.eclipse.gef4.fx.anchors.DynamicAnchor;

**import** org.eclipse.gef4.fx.nodes.Connection;

**import** org.eclipse.gef4.fx.nodes.GeometryNode;

**import** org.eclipse.gef4.geometry.planar.RoundedRectangle;

**import** javafx.scene.Group;

**import** javafx.scene.Node;

**import** javafx.scene.control.Label;

**import** javafx.scene.layout.BorderPane;

**import** javafx.scene.layout.StackPane;

**import** javafx.scene.paint.Color;

**import** javafx.scene.shape.Polygon;

**import** javafx.scene.shape.Rectangle;

**import** javafx.scene.shape.StrokeLineJoin;

**public** **class** DictionaryPart {

**private** **static** **final** **int** ***GRAPH\_WIDTH*** = 2000;

**private** **static** **final** **int** ***GRAPH\_HEIGHT*** = 1000;

**private** **static** **final** **int** ***DR*** = 20;

**private** **static** **final** **int** ***SMALL\_RADIUS*** = 75;

**private** **static** **final** **int** ***BIG\_RADIUS*** = 250;

**private** **static** **final** **int** ***GRAPH\_CENTER\_X*** = 300;

**private** **static** **final** **int** ***GRAPH\_CENTER\_Y*** = 300;

**private** **static** **final** **int** ***RECTANGLE\_SIZE*** = 40;

**private** **static** **final** **int** ***ARC\_SIZE*** = 10;

**public** DictionaryPart() {

System.***out***.println("Hhhh"); //$NON-NLS-1$

}

@PostConstruct

**void** initUI(BorderPane pane) {

Group group = **new** Group();

fillGroup(group);

pane.setCenter(group);

}

**private** **void** fillGroup(Group group) {

//org.eclipse.gef4.geometry.planar.Rectangle

Rectangle background = **new** Rectangle(0, 0, ***GRAPH\_WIDTH***, ***GRAPH\_HEIGHT***);

background.setFill(Color.***BLUE***);

group.getChildren().add(background);

**int** dAlpha = -1;

**for** (**int** cycleIndex = 0; cycleIndex < 3; cycleIndex++)

{

**int** radius = dAlpha == 0 ? ***BIG\_RADIUS*** : ***BIG\_RADIUS*** - ***DR***;

**for** (**int** rectIndex = 0; rectIndex < 20; rectIndex++)

{

**double** alpha = 2 \* Math.***PI*** / 20 \* rectIndex + dAlpha;

StackPane stackPane =

createStackPane(***GRAPH\_CENTER\_X*** + Double.*valueOf*(radius \* Math.*sin*(alpha)).intValue(),

***GRAPH\_CENTER\_Y*** + Double.*valueOf*(radius \* Math.*cos*(alpha)).intValue());

StackPane otherStackPane =

createStackPane(***GRAPH\_CENTER\_X*** + Double.*valueOf*(***SMALL\_RADIUS*** \* Math.*sin*(alpha)).intValue(),

***GRAPH\_CENTER\_Y*** + Double.*valueOf*(***SMALL\_RADIUS*** \* Math.*cos*(alpha)).intValue());

Connection connection = **new** Connection();

connection.setStartAnchor(**new** DynamicAnchor(getRectangle(stackPane)));

connection.setEndAnchor(**new** DynamicAnchor(getRectangle(otherStackPane)));

connection.setStartDecoration(**new** ArrowHead());

connection.setEndDecoration(**new** ArrowHead());

group.getChildren().add(stackPane);

group.getChildren().add(otherStackPane);

group.getChildren().add(connection);

}

dAlpha++;

}

}

**private** StackPane createStackPane(**int** x, **int** y) {

StackPane stackPane = **new** StackPane();

stackPane.setLayoutX(x);

stackPane.setLayoutY(y);

stackPane.getChildren().addAll(createRoundRectangle(), createRandomLabel());

**return** stackPane;

}

**private** Label createRandomLabel() {

Label label = **new** Label(createRandomString());

**return** label;

}

**private** String createRandomString() {

**return** **new** BigInteger(80, **new** Random()).toString();

}

**private** GeometryNode<RoundedRectangle> createRoundRectangle() {

//Rectangle rectangle = new Rectangle(0, 0, RECTANGLE\_SIZE, RECTANGLE\_SIZE);

GeometryNode<RoundedRectangle> rectangle =

**new** GeometryNode<>(**new** RoundedRectangle(0, 0, ***RECTANGLE\_SIZE***, ***RECTANGLE\_SIZE***, ***ARC\_SIZE***, ***ARC\_SIZE***));

//rectangle.setArcHeight(ARC\_SIZE);

//rectangle.setArcWidth(ARC\_SIZE);

rectangle.setStroke(Color.***BLACK***);

rectangle.setFill(Color.***WHITE***);

rectangle.setStrokeWidth(2);

**return** rectangle;

}

**private** Node getRectangle(StackPane stackPane) {

**for** (Node node : stackPane.getChildren())

{

**if** (node **instanceof** Rectangle)

{

**return** node;

}

}

**return** stackPane;

}

**private** **static** **class** ArrowHead

**extends** Polygon {

**public** ArrowHead() {

**super**(0, 0, 10, 3, 10, -3);

setFill(Color.***BLACK***);

setStroke(Color.***BLACK***);

setStrokeLineJoin(StrokeLineJoin.***ROUND***);

}

}

}

Код MANIFEST.MF:

Manifest-Version: 1.0

**Bundle-ManifestVersion**: 2

**Bundle-Name**: Wordnet Editor UI

**Bundle-SymbolicName**: ru.agentlab.hgdb.editor.ui;*singleton*:=true

**Bundle-Version**: 1.0.0.qualifier

**Require-Bundle**: javax.inject,

org.eclipse.osgi,

org.eclipse.e4.ui.di,

org.eclipse.e4.ui.services,

org.eclipse.e4.core.di.annotations,

org.eclipse.e4.core.services,

org.eclipse.e4.core.di,

org.eclipse.e4.ui.di,

org.eclipse.e4.core.di.extensions,

org.eclipse.e4.core.contexts,

org.eclipse.e4.ui.workbench,

org.eclipse.e4.ui.services,

org.eclipse.fx.ui.theme;*bundle-version*="3.0.0",

org.eclipse.fx.ui.services;*bundle-version*="3.0.0",

org.eclipse.jface;*bundle-version*="3.12.0",

org.eclipse.gef4.fx;*bundle-version*="1.0.1",

org.eclipse.gef4.geometry;*bundle-version*="1.0.1"

**Bundle-RequiredExecutionEnvironment**: JavaSE-1.8

**Import-Package**: com.google.common.util.concurrent;*version*="15.0.0",

net.sf.extjwnl,

net.sf.extjwnl.data,

net.sf.extjwnl.data.list,

net.sf.extjwnl.dictionary,

net.sf.extjwnl.service,

net.sf.extjwnl.utilities,

org.eclipse.fx.core.di;*version*="3.0.0"

Service-Component: OSGI-INF/services/theme-default.xml

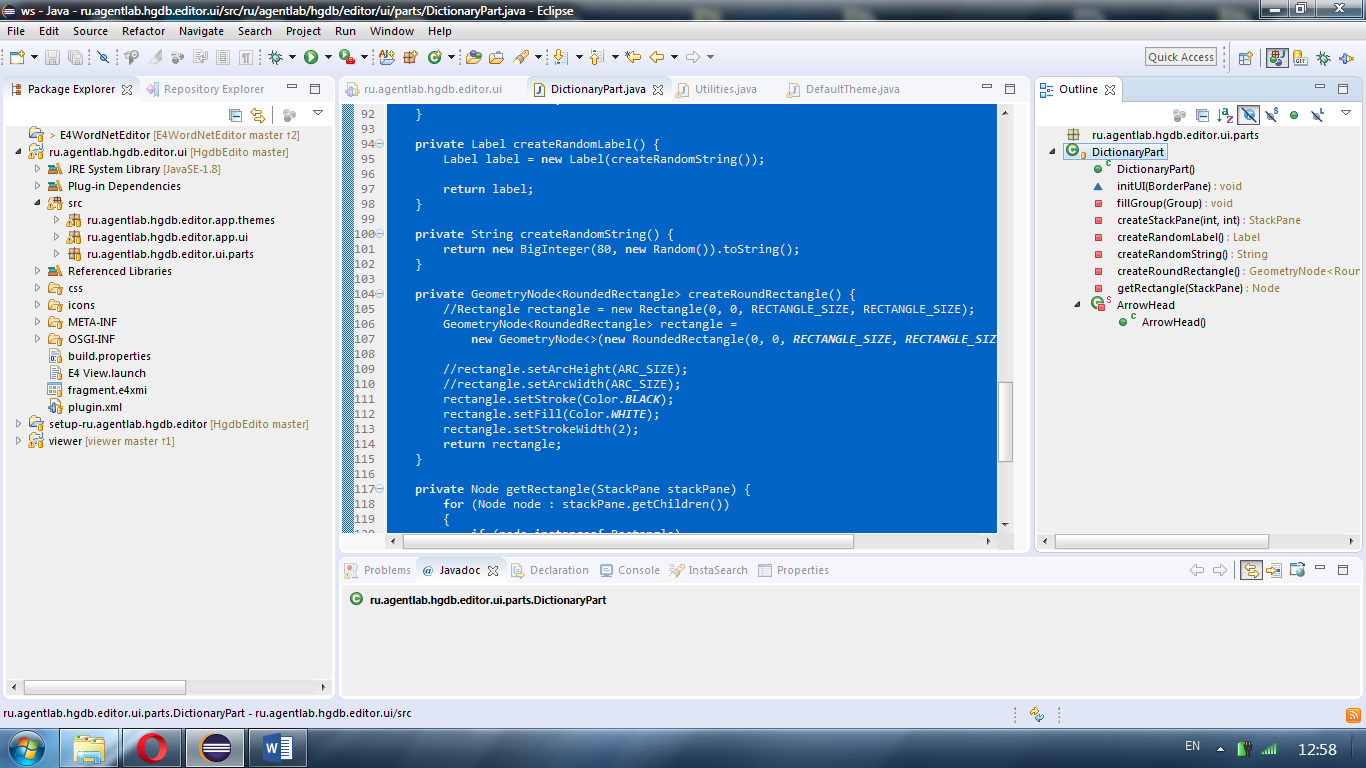
**Export-Package**: ru.agentlab.hgdb.editor.app.themes,

ru.agentlab.hgdb.editor.app.ui,

ru.agentlab.hgdb.editor.ui.parts

**Bundle-ActivationPolicy**: lazy

Сгенерированная модель:



Результат:

