```
/*
 * BeamModel.java
 */

import com.comsol.model.*;
import com.comsol.model.util.*;

/** Model exported on May 10 2015, 13:08 by COMSOL 5.1.0.136. */
public class BeamModel {

  public static Model run() {
    Model model = ModelUtil.create("Model");

    model.modelPath("C:\\Program Files\\COMSOL\\COMSOL51\\demo\\api\
\beammodel");

    model.name("beam_model.mph");

    model
        .comments("This model is used to calculate Areas and moment of intertia for a
beam. The calculated values can leter be used in a beam model.");

    model.baseSystem("mpa");

    model.param().set("H", "160[mm]");
    model.param().set("b", "146[mm]");
    model.param().set("tw", "13[mm]");
    model.param().set("tf", "22[mm]");
    model.param().set("r", "12[mm]");

    model.modelNode().create("mod1");

    model.geom().create("geom1", 2);
    model.geom("geom1").feature().create("r1", "Rectangle");
    model.geom("geom1").feature().create("c1", "Circle");
    model.geom("geom1").feature().create("b1", "BezierPolygon");
    model.geom("geom1").feature().create("dif1", "Difference");
    model.geom("geom1").feature().create("mir1", "Mirror");
    model.geom("geom1").feature().create("mir2", "Mirror");
    model.geom("geom1").feature("r1").set("size", new String[]{"b/2", "H/2"});
    model.geom("geom1").feature("c1").set("pos", new String[]{"tw/2+r", "H/2-tf-r"});
    model.geom("geom1").feature("c1").set("r", "r");
    model.geom("geom1").feature("b1")
        .set("p", new String[][]{{"b/2", "b/2", "tw/2+r", "tw/2", "tw/2", "b/2"}, {"0", "H/2-tf",
"H/2-tf", "H/2-tf-r", "0", "0"}});
    model.geom("geom1").feature("b1").set("degree", new String[]{"1", "1", "1", "1",
"1"});
    model.geom("geom1").feature("b1").set("w", new String[]{"1", "1", "1", "1", "1", "1",
"1", "1", "1", "1"});
    model.geom("geom1").feature("dif1").selection("input").set(new String[]{"r1"});
```

```
    model.geom("geom1").feature("dif1").selection("input2").set(new String[]{"b1",
"c1"});
    model.geom("geom1").feature("mir1").set("keep", true);
    model.geom("geom1").feature("mir1").set("axis", new String[]{"0", "1"});
    model.geom("geom1").feature("mir1").selection("input").set(new String[]{"dif1"});
    model.geom("geom1").feature("mir2").set("keep", true);
    model.geom("geom1").feature("mir2").selection("input").set(new String[]{"dif1",
"mir1"});
    model.geom("geom1").run();

    model.variable().create("var1");
    model.variable("var1").model("mod1");
    model.variable("var1").set("A", "intop1(1)");
    model.variable("var1").set("Iy", "intop1(y^2)");
    model.variable("var1").set("Iz", "intop1(z^2)");
    model.variable("var1").set("J", "intop1(2*T)");

    model.physics().create("ht", "HeatTransfer", "geom1");
    model.physics("ht").feature().create("hs1", "HeatSource", 2);
    model.physics("ht").feature("hs1").selection().set(new int[]{1, 2, 3, 4});
    model.physics("ht").feature().create("temp1", "TemperatureBoundary", 1);
    model.physics("ht").feature("temp1").selection()
        .set(new int[]{1, 2, 3, 4, 5, 6, 7, 8, 11, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23,
24});

    model.mesh().create("mesh1", "geom1");
    model.mesh("mesh1").feature().create("ftri1", "FreeTri");

    model.cpl().create("intop1", "Integration", "geom1");
    model.cpl("intop1").selection().all();

    model.variable("var1").name("Variables 1a");

    model.view("view1").axis().set("xmin", "-103.97526550292969");
    model.view("view1").axis().set("xmax", "103.40934753417969");
    model.view("view1").axis().set("ymin", "-89.1318359375");
    model.view("view1").axis().set("ymax", "86.8681640625");
    model.view("view1").axis().set("xextra", new String[]{});
    model.view("view1").axis().set("yextra", new String[]{});

    model.physics("ht").feature("solid1").set("k_mat", "userdef");
    model.physics("ht").feature("solid1")
        .set("k", new String[][]{{"1"}, {"0"}, {"0"}, {"0"}, {"1"}, {"0"}, {"0"}, {"0"}, {"1"}});
    model.physics("ht").feature("hs1").set("Q", "2*1e-3[W/mm^3]");
    model.physics("ht").feature("temp1").set("T0", "0");

    model.mesh("mesh1").feature("size").set("hauto", 1);
    model.mesh("mesh1").run();

    model.study().create("std1");
```

```
model.study("std1").feature().create("stat", "Stationary");

model.sol().create("sol1");
model.sol("sol1").study("std1");
model.sol("sol1").attach("std1");
model.sol("sol1").feature().create("st1", "StudyStep");
model.sol("sol1").feature().create("v1", "Variables");
model.sol("sol1").feature().create("s1", "Stationary");
model.sol("sol1").feature("s1").feature().create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature().create("d1", "Direct");
model.sol("sol1").feature("s1").feature().remove("fcDef");

model.result().dataset().create("int1", "Integral");
model.result().create("pg1", "PlotGroup2D");
model.result("pg1").feature().create("surf1", "Surface");

model.sol("sol1").feature("st1").name("Compile Equations: Stationary {stat}");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").feature("fc1").set("initstep", "0.01");
model.sol("sol1").feature("s1").feature("fc1").set("minstep", "1.0E-6");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", "50");
model.sol("sol1").feature("s1").feature("d1").set("linsolver", "pardiso");
model.sol("sol1").runAll();

model.result("pg1").set("title", "Surface: Temperature (K) ");
model.result("pg1").set("windowtitle", "Graphics");
model.result("pg1").set("titleactive", false);
model.result("pg1").feature("surf1").set("differential", false);

model.name("BeamModel.mph");

model.variable("var1").set("Iy", "intop1(x^2)");
model.variable("var1").set("Iz", "intop1(y^2)");

model.resetAuthor("COMSOL");

model.name("BeamModel.mph");



model.result("pg1").run();

model.label("BeamModel.mph");

model
    .comments("BeamModel\n\nThis model is used to calculate Areas and moment
of intertia for a beam. The calculated values can leter be used in a beam model.");
```

```
    model.result("pg1").run();

    model.result().numerical().create("glA", "Global");
    model.result().numerical("glA").set("expr", "A");
    model.result().numerical().create("glIy", "Global");
    model.result().numerical("glIy").set("expr", "Iy");
    model.result().numerical().create("glIz", "Global");
    model.result().numerical("glIz").set("expr", "Iz");
    model.result().numerical().create("glJ", "Global");
    model.result().numerical("glJ").set("expr", "J");
    return model;
  }


}
```