```java
/**
 * ----------------------------------------------------------------------------
 * "THE BEER-WARE LICENSE" (Revision 42):
 * <doktor@dyregod.dk> <mads@danquah.dk> <tkrogh@ruc.dk> <tnjr@ruc.dk> wrote
 * this file. As long as you retain this notice you can do whatever you want
 * with this stuff. If we meet some day, and you think this stuff is worth it,
 * you can buy us a beer in return.
 * ----------------------------------------------------------------------------
 */

package dk.ruc.blaster.export;

import java.awt.Point;
import java.io.File;

import dk.ruc.blaster.model.Map;
import dk.ruc.blaster.model.Vertex;

import junit.framework.TestCase;

/**
 * Class to test Export to XML in the {@link Exporter} class
 *
 * Last change by: $Author: dyregod $
 *
 * $Header: /var/cvs/Alwazah\040Editor/test/dk/ruc/blaster/export/ExporterTest.j
ava,v 1.1 2002/12/15 18:40:31 dyregod Exp $
 *
 * @version $Revision: 1.1 $
 * @author dk13043
 */
public class ExporterTest extends TestCase
{

    File file = new File("fisk.xml");
    Map map = new Map();
    /**
     * Constructor for ExporterTest.
     * @param arg0
     */
    public ExporterTest(String arg0)
    {
        super(arg0);
    }

    public static void main(String[] args)
    {
        junit.textui.TestRunner.run(ExporterTest.class);
    }

    /**
     * @see TestCase#setUp()
     */
    protected void setUp() throws Exception
    {
        super.setUp();
        Vertex verts[] =
            new Vertex[] {
                map.addVertex(new Point(1, 1)),
                map.addVertex(new Point(5, 1)),
                map.addVertex(new Point(1, 5))};

        map.addTriangle(verts);
```



```java
        verts =
            new Vertex[] {
                map.addVertex(new Point(1, 5)),
                map.addVertex(new Point(5, 1)),
                map.addVertex(new Point(5, 5))};

        map.addTriangle(verts);
    }

    /**
     * @see TestCase#tearDown()
     */
    protected void tearDown() throws Exception
    {
        super.tearDown();
        file.delete();
    }

    /**
     * Tests Export method. Will only test if export does
     * not fail. Not if exported file is valid.
     */
    public void testExport()
    {

        try
        {
            Exporter.export(map, file);
            System.out.println(file.exists());
            if (file.length() != 0)
                assertTrue(true);
            else
                assertTrue("File length is 0", false);
        }
        catch (Throwable e)
        {

            assertTrue(e.getMessage(), false);
        }
    }

}
```

```
/**
 * ----------------------------------------------------------------------------
 * "THE BEER-WARE LICENSE" (Revision 42):
 * <doktor@dyregod.dk> <mads@danquah.dk> <tkrogh@ruc.dk> <tnjr@ruc.dk> wrote
 * this file. As long as you retain this notice you can do whatever you want
 * with this stuff. If we meet some day, and you think this stuff is worth it,
 * you can buy us a beer in return.
 * ----------------------------------------------------------------------------
 */

package dk.ruc.blaster.model;

import java.awt.Point;
import java.awt.Rectangle;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;

import junit.framework.TestCase;

/**
 * Used in testing the {@link Map} class
 *
 * Last change by: $Author: dyregod $
 *
 * $Header: /var/cvs/Alwazah\040Editor/test/dk/ruc/blaster/model/MapTest.java,v
1.2 2002/12/19 01:14:38 dyregod Exp $
 *
 * @version $Revision: 1.2 $
 * @author dk13043
 */
public class MapTest extends TestCase
{
    Map map = new Map();
    HashMap hmap = new HashMap();
    Triangle t1;
    Triangle t2;

    /**
     * Constructor for MapTest.
     * @param arg0
     */
    public MapTest(String arg0)
    {
        super(arg0);
    }

    /**
     * @see TestCase#setUp()
     */
    protected void setUp() throws Exception
    {
        super.setUp();

        hmap.put("width", String.valueOf(5000.0f));
        hmap.put("height", String.valueOf(5600.0f));
        hmap.put("gravityx", String.valueOf(5.0f));
        hmap.put("gravityy", String.valueOf(5.0f));
        hmap.put("name", "test");
        map.setProperties(hmap);

        Vertex verts[] =
            new Vertex[] {
```



```
                map.addVertex(new Point(10, 10)),
                map.addVertex(new Point(50, 10)),
                map.addVertex(new Point(10, 50))};

        t1 = map.addTriangle(verts);

        verts =
            new Vertex[] {
                map.addVertex(new Point(10, 50)),
                map.addVertex(new Point(50, 10)),
                map.addVertex(new Point(50, 50))};

        t2 = map.addTriangle(verts);
    }

    public void testBounds()
    {
        assertTrue( (map.bounds().width == 5000.0f) );
        assertTrue( (map.bounds().height == 5600.0f) );
    }

    public void testGetProperties()
    {
        HashMap hmapNew = map.getProperties();

        Iterator iter = hmapNew.keySet().iterator();

        while (iter.hasNext())
        {
            Object obj = iter.next();
            if (!hmapNew.get(obj).equals(hmap.get(obj)))
            {
                assertTrue(((String)obj) + " does not contain the right value", false);
                return;
            }

        }
        assertTrue(true);
    }

    public void testGetVerticesInRect()
    {
        Rectangle rect = new Rectangle(0,0,40,60);
        List list = map.getVerticesInRect(rect);
        assertTrue((list.size() == 2));

        rect = new Rectangle(5,5,10,10);
        list = map.getVerticesInRect(rect);
        assertTrue((list.size() == 1));

        rect = new Rectangle(20,20,20,20);
        list = map.getVerticesInRect(rect);
        assertTrue((list.size() == 0));
    }

    public void testGetTrianglesInRect()
    {
        Rectangle rect = new Rectangle(20,20,20,20);
        List list = map.getTrianglesInRect(rect);
        assertTrue((list.size() == 2));

        rect = new Rectangle(0,0,20,20);
        list = map.getTrianglesInRect(rect);
```

MapTest.java


```
        assertTrue(list.get(0).equals(t1));

        rect = new Rectangle(0,0,5,5);
        list = map.getTrianglesInRect(rect);
        assertTrue((list.size() == 0));
    }

    /*
     * Test for EditorObject select(Point)
     */
    public void testSelectPoint()
    {
        EditorObject obj = map.select(new Point(40,40));
        assertTrue("Object is equal to t2", obj.equals(t2));

        obj = map.select(new Point(20,20));
        assertTrue("Object is equal to t1", obj.equals(t1));

        obj = map.select(new Point(0,0));
        assertTrue("Object is equal to map", (obj.equals(map)) );

        obj = map.select(new Point(11,12));
        assertTrue("Object is an Vertex", (obj instanceof Vertex) );
    }
}
```

VertexTest.java


```
/**
 * ----------------------------------------------------------------------------
 * "THE BEER-WARE LICENSE" (Revision 42):
 * <doktor@dyregod.dk> <mads@danquah.dk> <tkrogh@ruc.dk> <tnjr@ruc.dk> wrote
 * this file. As long as you retain this notice you can do whatever you want
 * with this stuff. If we meet some day, and you think this stuff is worth it,
 * you can buy us a beer in return.
 * ----------------------------------------------------------------------------
 */

package dk.ruc.blaster.model;

import junit.framework.TestCase;

/**
 * Class to test {@link Vertex}
 *
 * Last change by: $Author$
 *
 * $Header$
 *
 * @version $Revision$
 * @author dyregod
 */
public class VertexTest extends TestCase
{

    Vertex[] va1 = new Vertex[3];
    Vertex[] va2 = new Vertex[3];

    Vertex v1;
    Vertex v2;
    Vertex v3;
    Vertex v4;
    Vertex v5;

    Triangle t1;
    Triangle t2;
    /**
     * Constructor for VertexTest.
     * @param arg0
     */
    public VertexTest(String arg0)
    {
        super(arg0);
    }

    /**
     * @see TestCase#setUp()
     */
    protected void setUp() throws Exception
    {
        super.setUp();
        v1 = new Vertex(0, 0);
        v2 = new Vertex(10, 10);
        v3 = new Vertex(10, 0);
        v4 = new Vertex(-10, 0);
        v5 = new Vertex(-10, -10);

        va1[0] = v1;
        va1[1] = v2;
        va1[2] = v3;
```

```
        va2[0] = v3;
        va2[1] = v4;
        va2[2] = v5;

        t1 = new Triangle(va1);
        t2 = new Triangle(va2);
    }

    /**
     * @see TestCase#tearDown()
     */
    protected void tearDown() throws Exception
    {
        super.tearDown();
    }

    public void testMove()
    {
        float posx = v2.getPosition().x;
        float posy = v2.getPosition().y;

        v2.move(2, 2);
        assertTrue(v2.position.x == (posx + 2.0f));
        assertTrue(v2.position.y == (posy + 2.0f));

    }

    public void testGetNumberOfAttachedPolygons()
    {

        assertTrue(
            "Number of attached polygons == 2",
            (v3.getNumberOfAttachedPolygons() == 2));
        v3.removePoly(t2);
        assertTrue(
            "Number of attached polygons == 1",
            (v3.getNumberOfAttachedPolygons() == 1));
        v3.addPoly(t2);
        assertTrue(
            "Number of attached polygons == 2",
            (v3.getNumberOfAttachedPolygons() == 2));

    }

}
```