Университет ИТМО

Факультет программной инженерии и компьютерных технологий

Лабораторная работа №3 по Методам и Средствам Программной Инженерии

Выполнил: Богатов Александр Сергеевич

Группа: P3233

Вариант: 1437

Преподаватель: Цопа Евгений Алексеевич

Санкт-Петербург

2022

Задание:

Написать сценарий для утилиты Apache Ant, реализующий компиляцию, тестирование и упаковку в jar-архив кода проекта из лабораторной работы №3 по дисциплине "Веб-программирование".

Каждый этап должен быть выделен в отдельный блок сценария; все переменные и константы, используемые в сценарии, должны быть вынесены в отдельный файл параметров; MANIFEST.MF должен содержать информацию о версии и о запускаемом классе.

Сценарий должен реализовывать следующие цели (targets):

1. **compile** -- компиляция исходных кодов проекта.
2. **build** -- компиляция исходных кодов проекта и их упаковка в исполняемый jar-архив. Компиляцию исходных кодов реализовать посредством вызова цели **compile**.
3. **clean** -- удаление скомпилированных классов проекта и всех временных файлов (если они есть).
4. **test** -- запуск junit-тестов проекта. Перед запуском тестов необходимо осуществить сборку проекта (цель **build**).
5. **doc** - добавление в MANIFEST.MF MD5 и SHA-1 файлов проекта, а также генерация и добавление в архив javadoc по всем классам проекта.
6. **env** - осуществляет сборку и запуск программы в альтернативных окружениях; окружение задается версией java и набором аргументов виртуальной машины в файле параметров.

Unit test:

```
import data.Result;
import org.junit.Test;

import static org.junit.Assert.assertTrue;
import static org.junit.Assert.assertFalse;

public class CalculationTest {
    private final Result = new Result();

    @Test
    public void hitCircle() {
        result.setX(-0.2);
        result.setY(-0.2);
        result.setR(1);
        assertTrue(result.testHit());
    }

    @Test
    public void hitCircleFarBorder() {
        result.setX(-1);
        result.setY(0);
        result.setR(1);
```

```
    assertTrue(result.testHit());
}

@Test
public void hitCircleLowBorder() {
    result.setX(0);
    result.setY(-1);
    result.setR(1);
    assertTrue(result.testHit());
}

@Test
public void hitCircleArchBorder() {
    result.setX(-Math.sqrt(2)/2);
    result.setY(-Math.sqrt(2)/2);
    result.setR(1);
    assertTrue(result.testHit());
}

@Test
public void missCircle() {
    result.setX(-1);
    result.setY(-1);
    result.setR(1);
    assertFalse(result.testHit());
}

@Test
public void missCircleFarBorder() {
    result.setX(-1.001);
    result.setY(0);
    result.setR(1);
    assertFalse(result.testHit());
}

@Test
public void missCircleLowBorder() {
    result.setX(0);
    result.setY(-1.001);
    result.setR(1);
    assertFalse(result.testHit());
}

@Test
public void missCircleArchBorder() {
    result.setX(-Math.sqrt(2)/2 - 0.001);
    result.setY(-Math.sqrt(2)/2 - 0.001);
    result.setR(1);
    assertFalse(result.testHit());
}

@Test
```

```
public void hitTriangle() {
    result.setX(0.2);
    result.setY(-0.2);
    result.setR(1);
    assertTrue(result.testHit());
}

@Test
public void hitTriangleLowBorder() {
    result.setX(0.001);
    result.setY(-0.499);
    result.setR(1);
    assertTrue(result.testHit());
}

@Test
public void hitTriangleTopBorder() {
    result.setX(0.5);
    result.setY(0);
    result.setR(1);
    assertTrue(result.testHit());
}

@Test
public void hitTriangleHBorder() {
    result.setX(0.25);
    result.setY(-0.25);
    result.setR(1);
    assertTrue(result.testHit());
}

@Test
public void missTriangle() {
    result.setX(0.5);
    result.setY(0.5);
    result.setR(1);
    assertFalse(result.testHit());
}

@Test
public void missTriangleLowBorder() {
    result.setX(0.001);
    result.setY(-0.5);
    result.setR(1);
    assertFalse(result.testHit());
}

@Test
public void missTriangleTopBorder() {
    result.setX(0.501);
    result.setY(0);
    result.setR(1);
```

```
    assertFalse(result.testHit());
}

@Test
public void missTriangleHBorder() {
    result.setX(Math.sqrt(3)*1/4 + 0.001);
    result.setY(Math.sqrt(3)*1/4 + 0.001);
    result.setR(1);
    assertFalse(result.testHit());
}

@Test
public void hitRectangle() {
    result.setX(-0.2);
    result.setY(0.2);
    result.setR(1);
    assertTrue(result.testHit());
}

@Test
public void hitRectangleLowBorder() {
    result.setX(-0.5);
    result.setY(0);
    result.setR(1);
    assertTrue(result.testHit());
}

@Test
public void hitRectangleTopBorder() {
    result.setX(0);
    result.setY(1);
    result.setR(1);
    assertTrue(result.testHit());
}

@Test
public void hitRectangleFarTopBorder() {
    result.setX(-0.5);
    result.setY(1);
    result.setR(1);
    assertTrue(result.testHit());
}

@Test
public void missRectangle() {
    result.setX(-1);
    result.setY(1);
    result.setR(1);
    assertFalse(result.testHit());
}

@Test
```

```
    public void missRectangleLowBorder() {
        result.setX(- 0.5 - 0.001);
        result.setY(0.001);
        result.setR(1);
        assertFalse(result.testHit());
    }

    @Test
    public void missRectangleTopBorder() {
        result.setX(0);
        result.setY(1 + 0.001);
        result.setR(1);
        assertFalse(result.testHit());
    }

    @Test
    public void missRectangleFarTopBorder() {
        result.setX(-0.5);
        result.setY(-1 - 0.001);
        result.setR(1);
        assertFalse(result.testHit());
    }

    @Test
    public void hitEmptySector() {
        result.setX(0.5);
        result.setY(0.5);
        result.setR(1);
        assertFalse(result.testHit());
    }

    @Test
    public void rSwitch() {
        result.setX(-1);
        result.setY(2);
        result.setR(1);
        assertFalse(result.testHit());
        result.setR(2.5);
        assertTrue(result.testHit());
        result.setR(5);
        assertTrue(result.testHit());
    }
}
```

Build.xml:

```
<?xml version="1.0"?>
<project name="Web3" default="build">
            <property file="build.properties"/>

            <path id="classpath">
        <fileset dir="${lib.dir}" includes="*.jar"/>
```

```
    <fileset dir="${main.dir}" includes="*.java"/>
</path>

        <path id="classpath.test">
    <pathelement location="${junit}"/>
                <pathelement location="${hamcrest}"/>
    <pathelement location="${classes.dir}"/>
</path>

<target name="compile" depends="clean">
    <mkdir dir="${classes.dir}"/>
    <mkdir dir="${test.classes.dir}"/>
    <javac srcdir="${src.dir}" destdir="${classes.dir}" classpathref="classpath"/>
    <echo message="***** COMPILING COMPLETED *****"/>
</target>

        <target name="build" depends="compile">
                <copy todir="${build.dir}" >
       <fileset dir="${web.dir}"/>
    </copy>
    <copy todir="${build.dir.lib}">
       <fileset dir="${lib.dir}"/>
    </copy>
    <jar destfile="${build.dir}/${ant.project.name}.jar">
       <fileset dir="${classes.dir}"/>
       <manifest>
          <attribute name="Created-By" value="Voldemort" />
          <attribute name="Manifest-Version" value="1.0"/>
          <attribute name="Main-Class" value="NoClass"/>
       </manifest>
    </jar>
    <war destfile="${build.dir}/${ant.project.name}.war" webxml="${build.webxml}">
       <fileset dir="${build.dir}"/>
       <manifest>
          <attribute name="Created-By" value="Voldemort" />
          <attribute name="Manifest-Version" value="1.0" />
          <attribute name="Main-Class" value="NoClass" />
       </manifest>
    </war>
    <echo message="***** BUILDING COMPLETED *****"/>
</target>

<target name="clean">
    <delete dir="${build.dir}"/>
    <delete dir="${junit.report.dir}"/>
    <delete dir="${doc.dir}"/>
    <echo message="***** CLEANING COMPLETED *****"/>
</target>

        <target name="test" depends="build">
    <mkdir dir="${test.classes.dir}"/>
                <mkdir dir="${junit.report.dir}"/>
```

```
    <javac destdir="${test.classes.dir}" srcdir="${test.dir}" includeantruntime="false"
encoding="utf-8">
                                <classpath refid="classpath.test"/>
                        </javac>
    <junit printsummary="on" haltonfailure="true" haltonerror="true">
      <classpath>
        <path refid="classpath.test"/>
        <pathelement location="${test.classes.dir}"/>
      </classpath>
      <batchtest fork="yes" todir="${junit.report.dir}">
        <formatter type="xml"/>
        <fileset dir="${test.dir}" includes="*Test.java"/>
      </batchtest>
    </junit>
    <echo message="***** TESTING COMPLETED *****"/>
  </target>

        <target name="doc" depends="build">
    <mkdir dir="${doc.dir}"/>
    <signjar jar="${build.dir}/${ant.project.name}.jar"
          alias="me"
          storepass="itmo22"
          keystore="my-release-key.keystore"
          digestalg="MD5"
          />
    <signjar jar="${build.dir}/${ant.project.name}.jar"
          alias="me"
          storepass="itmo22"
          keystore="my-release-key.keystore"
          digestalg="SHA1"
          />
    <javadoc sourcepath="${src.dir}"
          destdir="${doc.dir}"
          classpathref="classpath">
      <fileset dir="${main.dir}" includes="*/*.java"/>
      <fileset dir="${main.dir}" includes="*.java"/>
    </javadoc>
    <javadoc classpath="${src.dir}"
          classpathref="classpath"
         destdir="${doc.dir}"
         author="true"
         version="true"
         use="true">
      <fileset dir="${main.dir}"/>
    </javadoc>
    <jar destfile="${build.dir}/${ant.project.name}.jar"
       basedir="${doc.dir}"
       update="true">
    </jar>
                        <echo message="***** DOCUMENTING COMPLETED *****"/>
  </target>
```

```
   <target name="env">
                        <mkdir dir="${classes.dir}"/>
      <javac srcdir="${main.dir}" destdir="${classes.dir}" classpathref="classpath"
source="${compile.version}"
            includeantruntime="false">
      </javac>
      <copy todir="${classes.dir}">
         <fileset dir="${resources.dir}"/>
      </copy>
      <antcall target="build"/>
      <delete file="${server.deploy}/${ant.project.name}.war"/>
      <delete file="${server.deploy}/${ant.project.name}.war.deployed"/>
      <copy file="${build.dir}/${ant.project.name}.war" todir="${server.deploy}"/>
      <exec executable="bash" spawn="true">
         <arg value="${server.launch}"/>
      </exec>
   </target>
</project>
```

Вывод:

При выполнении данной работы была изучена система сборки Apache Ant.