

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

ЛАБОРАТОРНАЯ РАБОТА №3
по дисциплине
“Методы и средства программной инженерии”
Вариант № 9

Студенты:

Степанов Михаил
Андреевич

Агеев Дмитрий Сергеевич

Группа Р3231

Преподаватель:

Письмак Алексей Евгеньевич



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 запуске 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. **scp** - перемещение собранного проекта по scp на выбранный сервер по завершению сборки. Предварительно необходимо выполнить сборку проекта (цель `build`).
6. **alt** - создаёт альтернативную версию программы с измененными именами переменных и классов (используя задание `replace/replaceregexp` в файлах параметров) и упаковывает её в jar-архив. Для создания jar-архива использует цель `build`.

Выполнение:

Compile:

```
<target name="compile">
  <mkdir dir="${build.dir}/classes" />
  <mkdir dir="${build.dir}/failingTests" />
  <javac srcdir="${source.dir}/java" includeantruntime="false"
    destdir="${build.dir}/classes" classpathref="PATH">
  </javac>
</target>
```

Build:

```
<target name="build" depends="compile, music">
  <manifest file="./src/main/resources/META-INF/MANIFEST.MF" mode="update">
    <attribute name="Main-Class" value="${main.class}" />
  </manifest>
  <jar destfile="${build.dir}/${jar.name}.jar" basedir="${build.dir}/classes"
    manifest="./src/main/resources/META-INF/MANIFEST.MF">
    <fileset dir="${basedir}"/>
    <include name="lib/*.jar"/>
  </fileset>
  <fileset dir="${source.dir}">
    <exclude name="java/**" />
  </fileset>
</jar>
</target>
```

Clean:

```
<target name="clean">
  <delete dir="${build.dir}"/>
</target>
```

Test:

```
<target name="test" depends="build">
  <mkdir dir="${build.dir}/tests" />
  <javac srcdir="${test.dir}/java" includeantruntime="false"
    destdir="${build.dir}/tests" classpathref="PATH">
  </javac>
  <junit printsummary="on">
    <classpath path="./lib">
      <path refid="PATH" />
    </classpath>
    <test name="data.InformTest"/>
    <classpath>
      <pathelement location="lib/all/junit-3.8.1.jar.jar"/>
      <pathelement location="${build.dir}/tests"/>
    </classpath>
  </junit>
</target>
```

Sep:

```
<target name="scp" depends="build">
    <scp file="${build.dir}/lab3.jar" trust="true"
        todir="${login}@se.ifmo.ru:/home/${login}/"
        port="2222" password="${password}"/>
</target>
```

Alt:

```
<target name="alt">
    <delete dir="${alt.source.dir}" />
    <mkdir dir="${alt.source.dir}" />
    <copy todir="${alt.source.dir}/java">
        <fileset dir="${source.dir}/java" />
        <globmapper from="*.java" to="*Alt.java"/>
    </copy>
    <copy todir="${alt.source.dir}/">
        <fileset dir="${source.dir}/resources" />
    </copy>

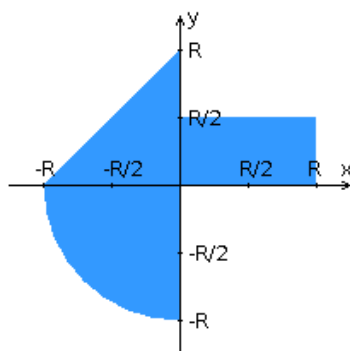
    <replace dir="${alt.source.dir}/java" token="Main" value="MainAlt" />
    <replace dir="${alt.source.dir}/java" token="CheckArea" value="CheckAreaAlt" />
    <replace dir="${alt.source.dir}/java" token="Validation" value="ValidationAlt" />
    <replace dir="${alt.source.dir}/java" token="HitBean" value="HitBeanAlt" />
    <replace dir="${alt.source.dir}/java" token="TimeBean" value="TimeBeanAlt" />
    <replace dir="${alt.source.dir}/java" token="Inform" value="InformAlt" />
    <replace dir="${alt.source.dir}/java" token="Mistake" value="MistakeAlt" />
    <replace dir="${alt.source.dir}/java" token="DAO" value="DAOAlt" />
    <replace dir="${alt.source.dir}/java" token="consequence" value="consequenceAlt" />
    <replace dir="${alt.source.dir}/java" token="Ready" value="ReadyAlt" />
    <replace dir="${alt.source.dir}/java" token="HibernateSessionFactory" value="HibernateSessionFactoryAlt" />
    <replaceregexp match="(\\W)PointData(\\W)" replace="\\1PointDataAlt\\2" flags="g">
        <fileset dir="${alt.source.dir}/java"/>
    </replaceregexp>

    <replaceregexp match="(\\W)r(\\W)" replace="\\1radius\\2" flags="g" >
        <fileset dir="${alt.source.dir}/java"/>
    </replaceregexp>

    <replaceregexp match="(\\W)odt(\\W)" replace="\\1offsetDateTime\\2" flags="g" >
        <fileset dir="${alt.source.dir}/java"/>
    </replaceregexp>

    <antcall target="build">
        <param name="source.dir" value="${alt.source.dir}" />
        <param name="jar.name" value="${jar.name}Alt" />
        <param name="main.class" value="${main.class}Alt" />
        <param name="build.dir" value="${build.dir}/alt" />
    </antcall>
</target>
```

Тесты:



```

public void testCalculateHitTrianglePositive() {
    assertEquals( expected: "Да", checkPoint( x: -2, y: 0.5, r: 4));
    assertEquals( expected: "Да", checkPoint( x: -0.3, y: 1, r: 3));
}

public void testCalculateTriangleHitNegative() {
    assertEquals( expected: "Нет", checkPoint( x: -4, y: 4, r: 4));
    assertEquals( expected: "Нет", checkPoint( x: -1, y: 3.5, r: 3.5));
}

public void testCalculateCircleHitPositive() {
    assertEquals( expected: "Да", checkPoint( x: -2, y: -2, r: 4));
    assertEquals( expected: "Да", checkPoint( x: -1.2, y: -0.1, r: 2.5));
}

public void testCalculateCircleNegative() {
    assertEquals( expected: "Нет", checkPoint( x: -2, y: -5, r: 4));
    assertEquals( expected: "Нет", checkPoint( x: -1.1, y: -4, r: 2.1));
}

public void testCalculateRectangleHitPositive() {
    assertEquals( expected: "Да", checkPoint( x: 2, y: 1, r: 4));
    assertEquals( expected: "Да", checkPoint( x: 2.45, y: 1.23, r: 3.1));
}

public void testCalculateRectangleNegative() {
    assertEquals( expected: "Нет", checkPoint( x: 5, y: 1.3, r: 2));
    assertEquals( expected: "Нет", checkPoint( x: 1.34, y: 3.23, r: 3.23));
}

public void testCalculateFreeAreaNegative() { assertEquals( expected: "Нет", checkPoint( x: 2, y: -1, r: 4)); }

```

Вывод:

В ходе выполнения данной лабораторной работы ознакомились системой сборки Apache Ant, написав сценарий реализующие различные цели. Помимо этого были реализованы модульные тесты для лабораторной работы №3 по дисциплине “Веб-программирование”. Во время их реализации мы узнали о различных видах тестирования и о их целях применения.