

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики»

ФАКУЛЬТЕТ ПИИКТ

ЛАБОРАТОРНАЯ РАБОТА № 3

по дисциплине

‘ОСНОВЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ’

Вариант №- 23438

Выполнил:

Студент группы Р3233

Шикунов Максим

Хасаншин Марат



УНИВЕРСИТЕТ ИТМО

Санкт-Петербург, 2024

Задание:

Лабораторная работа #3

Вариант 23438

Внимание! У разных вариантов разный текст задания!

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 класса.

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. **report** - в случае успешного прохождения тестов сохраняет отчет junit в формате xml, добавляет его в репозиторий svn и выполняет commit.

Build.xml:

```
<?xml version="1.0"?>
<project name="Lab3">
  <property file="build.properties"/>

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

  <path id="classpath.test">
    <fileset dir="${lib.dir}" includes="*.jar"/>
    <fileset dir="${test}" includes="**/*Test*.java"/>
    <pathelement location="${compiled.classes}"/>
    <pathelement location="${compiled.tests}"/>
  </path>

  <target name="compile" depends="clean">
    <mkdir dir="${compiled.classes}"/>
    <mkdir dir="${compiled.tests}"/>
    <javac srcdir="${main}" destdir="${compiled.classes}"
includeantruntime="false" classpathref="classpath"/>
    <javac srcdir="${test}" destdir="${compiled.tests}"
includeantruntime="false" classpathref="classpath.test"/>
  </target>

  <target name="build" depends="compile">
    <jar destfile="${jar}" basedir="${compiled.classes}">
      <manifest>
        <attribute name="Built-By" value="${creators}"/>
        <attribute name="Implementation-Version" value="${version}"/>
      </manifest>
    </jar>
  </target>

  <target name="clean">
    <delete dir="${compiled}"/>
  </target>

  <target name="test" depends="build">
    <mkdir dir="${report.test}"/>
    <junit fork="true" printsummary="on" failureproperty="checker">
      <classpath>
        <path refid="classpath.test"/>
      </classpath>
    </junit>
  </target>
```

```

        <batchtest todir="${report.test}">
            <formatter type="xml"/>
            <fileset dir="${compiled.tests}" includes="**/*Test*.class"/>
        </batchtest>
    </junit>
</target>

<target name="doc" depends="build">
    <mkdir dir="${docks}"/>
    <checksum algorithm="SHA-1" file="${jar}" property="sha1" />
    <checksum algorithm="MD5" file="${jar}" property="md5" />

    <javadoc sourcepath="${main.java}"
        destdir="${docks}" classpathref="classpath" >
    </javadoc>

    <jar update="true" destfile="${jar}">
        <manifest>
            <attribute name="MD5" value="${sha1}"/>
            <attribute name="SHA-1" value="${md5}"/>
        </manifest>
        <zipfileset dir="${docks}" prefix="docs"/>
    </jar>
</target>

<target name="report" depends="test" unless="checker">
    <exec dir="${report.test}" executable="svn">
        <arg value="add"/>
        <arg value="${report.file}"/>
        <arg value="--force"/>
    </exec>

    <exec dir="${report.test}" executable="svn">
        <arg value="commit"/>
        <arg value="-m"/>
        <arg value="Adding JUnit XML report"/>
    </exec>
</target>
</project>

```

Build.properties:

```

lib.dir=lib
junit.jar=lib/junit-4.13.2.jar
hamcrest.jar=lib/hamcrest-core-1.3.jar

compiled=target
compiled.classes=target/classes
compiled.tests=target/tests
report.test=main_report/trunk
report.file=TEST-HitTest.xml

main=src/main
main.java=src/main/java
main.dir=${main}/java/shika/jsf_site2/
test=src/test
jar=target/Lab3.jar

docks=target/doc

```

```
creators=Max and Marat  
version=1.0
```

HitTest.java:

```
import org.junit.Test;  
import org.junit.jupiter.api.Assertions;  
import shika.jsf_site2.model.Point;  
  
public class HitTest {  
    @Test  
    public void checkHit() {  
        Point point = new Point();  
        point.setX(1.0);  
        point.setY(1.0);  
        point.setR(1.0);  
        point.checkHit();  
        Assertions.assertTrue(point.isHit());  
    }  
  
    @Test  
    public void checkNotHit() {  
        Point point = new Point();  
        point.setX(-1.0);  
        point.setY(-1.0);  
        point.setR(2.0);  
        point.checkHit();  
        Assertions.assertFalse(point.isHit());  
    }  
}
```

Вывод:

Узнали, как работать с Apache Ant и как с помощью него компилировать файлы, собирать jar-ник и другое. Научились писать unit тесты.