

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

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №3
по дисциплине «**Основы программной инженерии**»

Вариант: **8289**

Преподаватель:
Вадим Миху

Выполнил: Алферов Г.А.
Группа: P3207

1. Задание

Вариант 8289

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 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. **native2ascii** - преобразование [native2ascii](#) для копий файлов локализации (для тестирования сценария все строковые параметры необходимо вынести из классов в файлы локализации).
6. **report** - в случае успешного прохождения тестов сохраняет отчет junit в формате xml, добавляет его в репозиторий svn и выполняет commit.

2. Реализация build.properties

```
lib.dir=lib
junit=${lib.dir}/junit-4.13.2.jar
hamcrest=${lib.dir}/hamcrest-core-1.3.jar
contrib=${lib.dir}/ant-contrib-1.0b3.jar
jakarta=${lib.dir}/jakarta.persistence-api-3.1.0.jar"

src.dir=src/main
main.dir=${src.dir}/java/org/GleBlassUSA
web.dir=${src.dir}/webapp
resources.dir=${src.dir}/resources
native2ascii.resources=${resources.dir}/native2ascii

ant.dir=ant
build.dir=${ant.dir}/build
build.dir.lib=${build.dir}/WEB-INF/lib
build.web.xml=${build.dir}/WEB-INF/web.xml
classes.dir=${build.dir}/WEB-INF/classes

test.classes.dir=${build.dir}/test/classes
test.dir=src/test/java
junit.report.dir=reports

compile.version=1

report.dir=./report
svn.dir=./wc
build=./build
classes=${build}/classes
report.name=test.xml
```

3. Реализация build.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<project name="web3" default="build">

    <property file="build.properties"/>

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

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

    <taskdef resource="net/sf/antcontrib/antcontrib.properties">
        <classpath>
            <pathelement location="${contrib}"/>
        </classpath>
    </taskdef>

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

    <target name="compile" depends="clean">
        <echo message="-----COMPILING-----"/>
        <mkdir dir="${classes.dir}"/>
        <mkdir dir="${test.classes.dir}"/>
        <javac srcdir="${src.dir}" destdir="${classes.dir}"
classpathref="classpath" includeantruntime="false"/>
        <javac destdir="${test.classes.dir}" srcdir="${test.dir}"
includeantruntime="false" encoding="utf-8">
            <classpath refid="classpath.test"/>
        </javac>
        <echo message="-----COMPILING COMPLETED-----"/>
    </target>

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

</project>
```

```

        </jar>
        <war destfile="${build.dir}/${ant.project.name}.war"
webxml="${build.web.xml}">
            <fileset dir="${build.dir}"/>
            <manifest>
                <attribute name="Created-By" value="wrakelft"/>
                <attribute name="Manifest-Version" value="1.0"/>
                <attribute name="Main-Class" value="NoClass"/>
            </manifest>
        </war>
        <echo message="-----BUILDING COMPLETED-----"/>
    </target>

    <target name="test" depends="build">
        <echo message="-----TESTING-----"/>
        <mkdir dir="${junit.report.dir}"/>
        <trycatch>
            <try>
                <junit printsummary="yes" fork="yes" haltonfailure="true"
showoutput="true">
                    <classpath>
                        <path refid="classpath.test"/>
                        <pathelement location="${test.classes.dir}"/>
                    </classpath>
                    <formatter type="xml"/>
                    <batchtest todir="${junit.report.dir}">
                        <fileset dir="${test.dir}" includes="**/*Test.java"/>
                    </batchtest>
                </junit>
                <property name="test.successful" value="true"/>
            </try>
            <catch>
                <property name="test.failed" value="true"/>
            </catch>
        </trycatch>
        <echo message="-----TESTING COMPLETED-----"/>
    </target>

    <target name="report" depends="test">
        <echo message="-----REPORTING-----"/>
        <condition property="tests.successful">
            <not>
                <isset property="tests.failed"/>
            </not>
        </condition>
        <if>
            <isset property="test.successful"/>
            <then>
                <move todir="${svn.dir}">
                    <fileset dir="${junit.report.dir}" includes="TEST-*.xml"/>
                </move>
                <exec executable="svn">
                    <arg value="add"/>
                    <arg value="--force"/>
                    <arg value="${svn.dir}/*"/>
                </exec>
                <exec executable="svn">
                    <arg value="update"/>
                    <arg value="${svn.dir}/*"/>
                </exec>
                <exec executable="svn">
                    <arg value="commit"/>
                    <arg value="${svn.dir}/*"/>
                    <arg value="-m"/>

```

```

        <arg value="Report update"/>
    </exec>
</then>
<else>
    <echo message="Test not successful. Report hasn't been added"/>
</else>
</if>
    <echo message="-----REPORTING COMPLETED-----"/>
</target>

<target name="native2ascii">
    <echo message="-----NATIVE2ASCII-----"/>
    <delete dir="${native2ascii.resources}"/>
    <native2ascii encoding="utf-8" src="${resources.dir}"
dest="${native2ascii.resources}" includes="**/*.properties"/>
    <echo message="-----NATIVE2ASCII COMPLETED-----"/>
</target>

</project>

```

4. Вывод

В ходе выполнения лабораторной работы я познакомился с утилитой Apache Ant для автоматизации процесса сборки. Написал сценарии для своего проекта. Попробовал протестировать код с помощью JUnit.