

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 программного обеспечения”**

**Вариант: 1001**

Выполнила:

Лушникова Анастасия

Группа: Р32302

Преподаватель:

Исаев Илья Владимирович

# 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. **xml** - валидация всех xml-файлов в проекте.
6. **env** - осуществляет сборку и запуск программы в альтернативных окружениях; окружение задается версией java и набором аргументов виртуальной машины в файле параметров.

## Код:

```
<project name="web33" default="compile" basedir=". ">

  <property file="build.properties"/>
  <macrodef name="git_commit">
    <attribute name="file"/>
    <sequential>
      <exec executable="git">
        <arg value="add"/>
        <arg value="@{file}"/>
      </exec>
      <exec executable="git">
        <arg value="commit"/>
        <arg value="-m"/>
        <arg value="Tests was successfully completed"/>
      </exec>
    </sequential>
  </macrodef>
```

```

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

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

<target name="init">
    <tstamp/>
    <mkdir dir="${build}"/>
    <echo>-----Init successfully finished-----</echo>
</target>

<target name="clean" description="Clean the output directory">
    <delete dir="${build}"/>
    <delete dir="${build.output.test.dir}"/>
    <delete dir="${test_report}"/>
</target>

<target name="compile" depends="init"
    description="compile the source">
    <echo>-----Compile main code-----</echo>
    <mkdir dir="${build.output.dir}"/>
    <javac srcdir="${main}" destdir="${build.output.dir}" nowarn="false"
        debug="true"
        optimize="false"
        deprecation="true"
        target="16"
        verbose="false"
        fork="false"
        source="16">
        <classpath refid="build.classpath"/>
    </javac>

    <!--      <copy todir="${build.output.dir}">-->
    <!--          <fileset dir="${build.resourceDir.1}"/>-->
    <!--      </copy>-->
    <echo>-----Compile test code-----</echo>
    <mkdir dir="${build.output.test.dir}"/>
    <javac srcdir="${test}" destdir="${build.output.test.dir}"
nowarn="false"
        debug="true"
        optimize="false"
        deprecation="true"
        target="16"
        verbose="false"
        fork="false"
        source="16">

        <classpath>
            <path refid="build.test.classpath"/>

```

```

        <pathelement location="${build.output.dir}"/>
    </classpath>
</javac>
</target>

<target name="build" depends="compile" description="Package the
application">
    <mkdir dir="${build.fileName.web-inf.lib}"/>
    <echo>copy lib</echo>
    <echo>-----start build-----</echo>
    <copy todir="${build.fileName.web-inf.lib}">
        <path refid="build.classpath"/>
    </copy>
    <copy todir="${build.fileName.web-inf.lib.classes}">
        <fileset dir="${build.output.dir}"/>
    </copy>
    <copy todir="${build.fileName.web-inf}">
        <fileset dir="${web-inf}"/>
    </copy>
    <copy todir="${build.fileName.web-inf.lib.classes}">
        <fileset dir="${build.resourceDir.0}"/>
    </copy>
    <copy todir="${build.fileName.path}">
        <fileset dir="${webapp}">
            <exclude name="${web-inf}/**"/>
        </fileset>
    </copy>
    <echo>-----Creating war-archive...-----</echo>

    <war destfile="${build}/demo-1.0-SNAPSHOT.war">
        <fileset dir="${build.fileName.path}"/>
        <manifest>
            <attribute name="Main-Class" value="${main-class}"/>
        </manifest>
    </war>
</target>

<target name="test"
    depends="build"
    description="Run the test cases">
    <echo>-----Begin executing test classes-----</echo>
    <mkdir dir="${test_report}"/>
    <junit haltonfailure="yes" printSummary="yes"
failureproperty="tests_failed">
        <formatter type="xml"/>
        <classpath>
            <path refid="build.test.classpath"/>
            <pathelement location="${build.output.dir}"/>
            <pathelement location="${build.output.test.dir}"/>
        </classpath>
        <batchtest fork="yes" todir="${test_report}">
            <path refid="build.test.classpath"/>
            <fileset dir="${build.testDir}" includes="**/*Test.java"/>
        </batchtest>
    </junit>
</target>

```

```

        </batchtest>
    </junit>
</target>

<target name="xml">
    <echo>-----validating xml-----</echo>
    <xmlvalidate failonerror="no" lenient="yes" warn="yes">
        <fileset dir="${webapp}" includes="*.xhtml"/>
        <fileset dir="${web-inf}" includes="*.xml"/>
    </xmlvalidate>
</target>

<target name="env" depends="build">
    <echo>-----executing env command-----</echo>
    <copy file="${build}/demo-1.0-SNAPSHOT.war" toDir="${env.deployments}"
/>
    <exec dir="${env.dir}" executable="${env.script}">
        <env key="JAVA_HOME" path="${java.17}" />
        <env key="JAVA_OPTS" value="${jvm.args}" />
    </exec>
</target>
</project>

```