

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 № 248310

Студенты:  
Сиразетдинов А.Н., Шпинаева У. С.  
Группа: Р3216  
Преподаватель:  
Письмак А.Е.

г. Санкт-Петербург  
2023/24

<b>Задание.....</b>	<b>3</b>
<b>Сценарий для утилиты Ant.....</b>	<b>4</b>
<b>Полный код.....</b>	<b>8</b>
<b>Вывод.....</b>	<b>8</b>

## 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 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. **music** - воспроизведение музыки по завершению сборки (цель **build**).
6. **native2ascii** - преобразование [native2ascii](#) для копий файлов локализации (для тестирования сценария все строковые параметры необходимо вынести из классов в файлы локализации).
7. **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**.
8. **report** - в случае успешного прохождения тестов сохраняет отчет junit в формате xml, добавляет его в репозиторий svn и выполняет `commit`.

## Сценарий для утилиты Ant

```
<project name="WebLab3"
    basedir="."
    default="music"
    xmlns:ivy="antlib:org.apache.ivy.ant"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="antlib:org.apache.ivy.ant"
    xmlns:if="ant:if"
    xmlns:unless="ant:unless">
    <property file="build.properties"/>

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

    <path id="test.classpath">
        <pathelement path="dist"/>
        <pathelement path="${build.classes}"/>
        <pathelement path="${build.tests}"/>
        <fileset dir="${ant.home}/lib" includes="*.jar" />
    </path>

    <target name="init">
        <tstamp/>
        <mkdir dir="${build}"/>
        <mkdir dir="${build.classes}"/>
        <mkdir dir="${build.tests}"/>
        <mkdir dir="${alt}"/>
        <mkdir dir="${altbuild}"/>
        <mkdir dir="${altbuild.classes}"/>
        <mkdir dir="${altbuild.tests}"/>
        <mkdir dir="${dist}"/>
        <mkdir dir="${dist.lib}"/>
    </target>

    <target name="download-ivy" unless="skip.download">
        <mkdir dir="${ivy.jar.dir}"/>
        <get
src="https://repo1.maven.org/maven2/org/apache/ivy/ivy/${ivy.install.version}/ivy-${ivy.install.version}.jar" dest="${ivy.jar.file}"
usetimestamp="true"/>
    </target>

    <target name="install-ivy" depends="download-ivy"
description="--> install ivy">
        <path id="ivy.lib.path">
            <fileset dir="${ivy.jar.dir}" includes="*.jar"/>
        </path>
    </target>
```

```

        </path>
        <taskdef resource="org/apache/ivy/ant/antlib.xml"
uri="antlib:org.apache.ivy.ant" classpathref="ivy.lib.path"/>
    </target>

    <target name="resolve dependencies" depends="install-ivy"
description="--> resolve dependencies">
        <ivy:retrieve log="quiet"/>
    </target>

    <target name="compile" depends="init, resolve dependencies"
description="--> compile the source">
        <javac includeantruntime="false"
            srcdir="${src}"
            destdir="${build.classes}">
            <classpath refid="classpathRuntime"/>
        </javac>
    </target>

    <target name="compile-tests" depends="compile" description="-->
compile tests">
        <javac includeantruntime="false"
            srcdir="${src.test}"
            destdir="${build.tests}">
            <classpath >
                <path refid="classpathRuntime"/>
                <path path="${src.test}"/>
                <pathelement location="${build.classes}"/>
            </classpath>
        </javac>
    </target>

    <target name="build" depends="compile" description="-->
building war">
        <ivy:retrieve
pattern="${dist.lib}/[artifact]-[revision].[ext]" conf="runtime"/>
        <war destfile="${dist.war.name}"
webxml="${src.webapp.webxml}">
            <fileset dir="${src.webapp}">
                <include name="**/*.*/>
            </fileset>
            <lib dir="${dist.lib}"/>
            <classes dir="${build.classes}"/>
            <manifest>
                <section name="Common">
                    <attribute name="Project-Name"
value="${project.name}"/>

```

```

        <attribute name="Project-Version"
value="${project.version}"/>
    </section>
</manifest>
</war>
</target>

<target name="music" description="--> play music">
    <splash showduration="0" imageurl="${splash-gif}"/>
    <sound>
        <success source="${resource.music.success}"/>
        <fail source="${resource.music.fail}"/>
    </sound>
    <antcall target="build"/>
</target>

<target name="native2ascii" description="Преобразование
native2ascii для копий файлов локализации">
    <native2ascii src="${src.main.resources}"
        dest="${native2ascii.resources}"
        includes="**/*.properties"/>
</target>

<target name="alt" description=" создаёт альтернативную версию
программы с измененными именами переменных и классов и упаковывает
её в jar-архив">
    <copy todir="${alt}">
        <fileset dir="${src}"/>
    </copy>

    <replaceregexp match="${alt.regex}"
        flags="${alt.regex.flags}"
        replace="${alt.replace}">
        <fileset dir="${alt}" includes="**/*.java"/>
    </replaceregexp>

    <antcall target="build">
        <param name="src" value="${alt}"/>
        <param name="src.main" value="${alt.main}"/>
        <param name="src.main.resources"
value="${alt.main.resources}"/>
        <param name="src.webapp" value="${alt.webapp}"/>
        <param name="src.webapp.webxml"
value="${alt.webapp.webxml}"/>
        <param name="src.test" value="${alt.test}"/>
        <param name="build" value="${altbuild}"/>
        <param name="build.classes"
value="${altbuild.classes}"/>

```

```

        <param name="build.tests" value="\${altbuild.tests}"/>
        <param name="dist.war.name"
value="\${dist}/${ant.project.name}-alt.war"/>
    </antcall>
</target>

<target name="test" depends="build, compile-tests"
description="--> run tests">
    <mkdir dir="\${dist.test}"/>
    <junitlauncher printsummary="true">
        <classpath refid="test.classpath"/>
        <testclasses outputdir="\${dist.test}">
            <fileset dir="\${build.tests}">
                <include name="**/*.class"/>
            </fileset>
            <listener type="legacy-xml"
                sendSysOut="true"
                sendSysErr="true"/>
        </testclasses>
    </junitlauncher>
</target>

<target name="report" depends="test" description="--> commit
report to svn">
    <delete dir="\${dist.report}"/>
    <mkdir dir="\${dist.report}"/>
    <junitreport todir="\${dist.report}">
        <fileset dir="\${dist.test}">
            <include name="TEST-*.xml"/>
        </fileset>
        <report format="frames" todir="\${dist.report}"/>
    </junitreport>
    <mkdir dir="\${svn.dir}"/>
    <move file="\${dist.report}" todir="\${svn.dir}"/>
    <exec dir="\${svn.dir}" executable="svn"
if:true="\${svn.use}">
        <arg line="add *"/>
    </exec>
    <exec dir="\${svn.dir}" executable="svn"
if:true="\${svn.use}">
        <arg line="commit -m \${svn.commit.name}
--username=\${svn.username}"/>
    </exec>
</target>

<target name="clean"
description="clean up">

```

```
<delete dir="${build}"/>
<delete dir="${dist.lib}"/>
<delete dir="${dist}"/>
<delete dir="${alt}"/>
<delete dir="${altbuild}"/>
<delete dir="${native2ascii.resources}"/>
</target>
</project>
```

## Полный код

[https://github.com/Azat2202/OPI\\_lab\\_3.git](https://github.com/Azat2202/OPI_lab_3.git)

## Вывод

В ходе выполнения лабораторной работе мы узнали про работу с системой сборки Ant и узнали про тестирование программ с помощью junit. Написали скрипт сборки проекта и базовые тесты.