СПб НИУ ИТМО

Кафедра ИПМ

Основы программной инженерии

Лабораторная работа № 4

Сценарий для утилиты Apache Ant.

Вариант 862.

Работу выполнил:

Студент II курса

Группы № 2120

Журавлев Виталий

Санкт-Петербург

2013 г.

**Цель работы:**

Написать сценарий для утилиты Apache Ant, реализующий компиляцию, тестирование и упаковку в jar-архив кода проекта из лабораторной работы №3.

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.

Cценарий должен реализовывать следующие цели (targets):

* compile - компиляция исходных кодов проекта.
* build - компиляция исходных кодов проекта и их упаковка в исполняемый jar-архив. Компиляцию исходных кодов реализовать посредством вызова цели compile.
* clean - удаление скомпилированных классов проекта и всех временных файлов (если они есть).
* test - запуск junit-тестов проекта. Перед запуском тестов необходимо осуществить сборку проекта (цель build).
* music - воспроизведение музыки по завершению сборки (цель build).
* history - если проект не удаётся скомпилировать (цель compile), загружается предыдущиая версия из репозитория git. Операция повторяется до тех пор, пока проект не удастся собрать, либо не будет получена самая первая ревизия из репозитория. Если такая ревизия найдена, то формируется файл, содержащий результат операции diff для всех файлов, измёненных в ревизии, следующей непосредственно за последней работающей.

Apache Ant — утилита для автоматизации процесса сборки программного продукта. Управление процессом сборки происходит посредством XML-сценария.

**XML-файл Build:**

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

<project default="compile" basedir="." name=”javalab3.JavaLab3>

 <property file="build.prop"/>

<target name="compile" depends="build" description="Compile the application">

<javac srcdir="${src}" destdir="${buildSrc}" debug="false" deprecation="true" onerror="history" optimize="true"/>

<javac srcdir="${test}" destdir="${build.test}" debug="false" deprecation="true" onerror="history" optimize="true">

<classpath>

<pathelement location="${junit.dir}"/>

<pathelement location="${buildSrc}"/>

</classpath>

</javac>

</target>

<target name="build" depends="clean" description="Build the application">

<mkdir dir ="${build}"/>

<mkdir dir="${build.classes}"/>

<mkdir dir="${build.test}"/>

<antcall target = "compile"/>

<jar jarfile="${build}/${name}.jar">

<manifest>

<attribute name="Autor" value="Zhuravlev Vitaly"/>

<attribute name="Main Class" value="javalab3.JavaLab3"/>

<attribute name="Current version" value="1.0"/>

</manifest>

</jar>

</target>

<target name="clean" description="Removes all build files">

<delete dir="${build}"/>

<delete dir="${build.classes}"/>

<delete dir="${build.test}"/>

</target>

 <target name="test" depends="build">

 <junit fork="yes" haltonfailure="yes">

 <classpath>

<pathelement location="${junit.dir}"/>

 <pathelement location="${build.test}"/>

 <pathelement location="${build}"/>

 </classpath>

 <formatter type="plain" usefile="false" />

 <test name="javalab3.JavaLab3Test"/>

 <test name="javalab3.SilhouetteTest"/>

 </junit>

 </target>

 <target name="music" depends="build" description="Play music when build">

 <exec executable="cmdmusic">

 <arg value="example.mp3"/>

 </exec>

</target>

<target name="history" depends="compile" description="If build error get previous version from GIT">

 <exec executable="cmdgit">

 <arg value="HEAD~"/>

 </exec>

 <antcall target = "compile"/>

 </target>

</project>

**Файл Build.prop**

#This is the propetries for Apache Ant

<property name="src.dir" location="${basedir}/src"/>

<property name="build" location="${basedir}/build"/>

<property name="build.classes" location="${build}/classes"/>

<property name="build.test" location="${build}/tests"/>

<property name="test" location="${basedir}/test"/>

<property name="cmdmusic" value="mplayer"/>

<property name="cmdgit" value="git checkout"/>

**Вывод:**

В результате выполнения данной лабораторной работы мы изучил основы Ant и научились автоматизировать процесс сборки проектов и сразу же проводить модульное тестирование.