Санкт-Петербургский национальный исследовательский университет  
 информационных технологий, механики и оптики  
Кафедра информатики и прикладной математики

Основы программной инженерии  
Лабораторная работа №3  
“Системы автоматизированной сборки”

Выполнил: Шкаруба Н.Е.  
Проверила: Харитонова   
группа: P3218  
год: 2015

Написать сценарий для утилиты [Apache Ant](http://ant.apache.org/), реализующий компиляцию, тестирование и упаковку в jar-архив кода проекта из [лабораторной работы №2](https://helios.cs.ifmo.ru/disciplines/iaps#lab2) по дисциплине "Программирование интернет-приложений".

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):**

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](http://docs.oracle.com/javase/1.4.2/docs/tooldocs/windows/native2ascii.html) для копий файлов локализации (для тестирования сценария все строковые параметры необходимо вынести из классов в файлы локализации).
6. **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**.

**Исходная разметка:**

<project name="Lab2" >

<property file="build.properties" />

<path id="application" location="${jar.dir}/${ant.project.name}.jar" />

<target name="init">

<manifest file="MANIFEST.MF">

<section name="">

<attribute name="Author" value="${author}" />

<attribute name="Version" value="${version}" />

<attribute name="MainClass" value="${main-class}" />

</section>

</manifest>

<tstamp/>

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

</target>

<target name="compile" depends="init">

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

<javac srcdir="${src.dir}" destdir="${classes.dir}" />

</target>

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

<copy todir="${classes.dir}">

<fileset dir="${resources.dir}" includes="\*.properties"/>

</copy>

<mkdir dir="${jar.dir}" />

<jar destfile="${jar.dir}/${ant.project.name}.jar" basedir="${classes.dir}" />   
 </target>

<target name="clean">

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

</target>

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

<java fork="true" classname="${main-class}">

<classpath>

<path refid="application"/>

</classpath>

</java>

</target>

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

<junit printsummary="yes">

<classpath>

<path refid="application"/>

</classpath>

<batchtest fork="yes">

<fileset dir="${src.dir}" includes="Tests/\*.java"/>

</batchtest>

</junit>

</target>

<target name="native2ascii" depends="compile">

<native2ascii encoding="UTF-8" src="${resources.dir}/native" dest="${resources.dir}" includes="\*.properties"/>

</target>

<target name="alt" depends="clean">

<replaceregexp file="build.properties" match="src.dir=(.\*)" replace="src.dir=srcAlt" byline="true" />

<replaceregexp file="build.properties" match="resources.dir=(.\*)" replace="resources.dir=srcAlt/Resources" byline="true" />

<antcall target="build" />

<replaceregexp file="build.properties" match="resources.dir=(.\*)" replace="resources.dir=src/Resources" byline="true" />

<replaceregexp file="build.properties" match="src.dir=(.\*)" replace="src.dir=src" byline="true" />

</target>

</project>