**ИТМО Кафедра Вычислительной техники**

Отчет по лабораторной работе №3

Основы программной инженерии

Вариант 520

**Выполнил: студент группы P3217**

**Плюхин Д.А.**

**2016 год**

1. **Задание к лабораторной работе**

Написать сценарий для утилиты Apache Ant, реализующий компиляцию, тестирование и упаковку в jar-архив кода проекта из лабораторной работы №3 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. **doc** - добавление в MANIFEST.MF MD5 и SHA-1 файлов проекта, а также генерация и добавление в архив javadoc по всем классам проекта.
6. **env** - осуществляет сборку и запуск программы в альтернативных окружениях; окружение задается версией java и набором аргументов виртуальной машины в файле параметров.
7. **Исходные файлы**
   1. **build.xml**

**<project name="Lab3" default="build" basedir=".">**

**<property environment=”env”/>**

**<property file="build.properties"/>**

**<taskdef file="${antlib.file}">**

**<classpath>**

**<pathelement location="${antcontrib.file}"/>**

**</classpath>**

**</taskdef>**

**<target name="makedirs.compile" description="Create necessary directories for .class files">**

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

**<mkdir dir="${class.dir}"/>**

**</target>**

**<target name="compile" depends="makedirs.compile" description="Compile our .java files">**

**<javac srcdir="${src.dir}" destdir="${class.dir}" includeAntRuntime="false"/>**

**</target>**

**<target name="build" depends="compile" description="Put compiled classes to the .jar file">**

**<jar destfile="${jar.file}" basedir="${class.dir}">**

**<manifest>**

**<attribute name="Version" value="2.0"/>**

**<attribute name="Main-Class" value="Lab3"/>**

**</manifest>**

**</jar>**

**</target>**

**<target name="makedirs.test" description="Make dirs for test">**

**<mkdir dir="${testclass.dir}"/>**

**</target>**

**<target name="compile.test" description="Compilation of the tests" depends="build, makedirs.test">**

**<javac srcdir="${test.dir}" destdir="${testclass.dir}" includeantruntime="false">**

**<classpath>**

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

**<pathelement path="${class.dir}"/>**

**</classpath>**

**</javac>**

**</target>**

**<target name="test" description="Run unit tests for the project" depends="compile.test">**

**<junit printsummary="yes" haltonerror="yes" haltonfailure="yes" fork="yes">**

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

**<classpath>**

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

**<pathelement location="${jar.file}"/>**

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

**<pathelement location="${hamcrest.file}"/>**

**</classpath>**

**<batchtest fork="yes" todir="${testreport.dir}">**

**<fileset dir="${test.dir}">**

**<include name="\*\*/\*Test.java"/>**

**</fileset>**

**</batchtest>**

**</junit>**

**</target>**

**<target name="doc" description="Add docs to the jar file" depends="build">**

**<unjar src="${jar.file}" dest="${unjar.dir}"/>**

**<for param="file">**

**<path>**

**<fileset dir="${unjar.dir}">**

**<exclude name="\*\*/\*.html"/>**

**<exclude name="\*\*/\*.js"/>**

**<exclude name="\*\*/\*.MF"/>**

**</fileset>**

**</path>**

**<sequential>**

**<local name="md5" />**

**<local name="sha1" />**

**<local name="relativeFile" />**

**<checksum file="@{file}" algorithm="md5" property="md5"/>**

**<checksum file="@{file}" algorithm="SHA-1" property="sha1"/>**

**<property name="relativeFile" location="@{file}" basedir="${unjar.dir}" relative="true" />**

**<manifest file="${unjaredmf.file}" mode="update">**

**<section name="${relativeFile}">**

**<attribute name="MD5" value="${md5}"/>**

**<attribute name="SHA-1" value="${sha1}"/>**

**</section>**

**</manifest>**

**</sequential>**

**</for>**

**<javadoc destdir="${unjar.dir}/doc">**

**<fileset dir="${src.dir}" defaultexcludes="yes">**

**<include name="\*\*/\*.java"/>**

**</fileset>**

**</javadoc>**

**<jar destfile="${jar.file}" manifest="${unjaredmf.file}" basedir="${unjar.dir}"/>**

**</target>**

**<target name="env.compile" description="Compile project in some invironment" depends="makedirs">**

**<javac destdir="${class.dir}" compiler="${compiler}">**

**<src path="${src.dir}"/>**

**</javac>**

**</target>**

**<target name="env" depends="env.compile">**

**<java classpath="${class.dir}" classname="Lab3" fork="true" jvm="${jvm}">**

**<arg value="${radius}"/>**

**<jvmarg value="${driver}"/>**

**<jvmarg value="${maxpermsize}"/>**

**<jvmarg value="${xms}"/>**

**</java>**

**</target>**

**<target name="clean" description="Delete all compiled files and temporary files">**

**<delete dir="${class.dir}"/>**

**<delete dir="${temp.dir}"/>**

**</target>**

**</project>**

* 1. **build.properties**

**build.dir=build**

**class.dir=${build.dir}/class**

**src.dir=src**

**lib.dir=${build.dir}/lib**

**jar.file=${lib.dir}/lab3.jar**

**temp.dir=${build.dir}/tmp**

**testclass.dir=${temp.dir}/test**

**test.dir=test**

**junit.file=${lib.dir}/junit-4.12.jar**

**testreport.dir=${temp.dir}/testreport**

**hamcrest.file=${lib.dir}/hamcrest-core-1.3.jar**

**unjar.dir=${temp.dir}/jar**

**antcontrib.file=${lib.dir}/ant-contrib-1.0b3.jar**

**antlib.file=${lib.dir}/antlib.xml**

**unjaredmf.file=${unjar.dir}/META-INF/MANIFEST.MF**

**antlib.file=${lib.dir}/antlib.xml**

**antcontrib.file=${lib.dir}/ant-contrib-1.0b3.jar**

**driver=-Dsql.driver=oracle.jdbc.driver.OracleDriver**

**maxpermsize=-XX:MaxPermSize=256m**

**xms=-Xms1024M**

**compiler=javac1.8**

**jvm=${env.JAVA\_HOME}/bin/java**

**radius=10**

1. **Вывод**

Таким образом, в результате лабораторной работы я познакомился с системами автоматической сборки и изучил основы модульного тестирования. На мой взгляд, системы автоматической сборки довольно полезны в деятельности разработчика за счет своей гибкости – с помощью них можно автоматизировать практически любое действие над файлами программы. Модульное тестирование же является неотъемлимой частью процесса разработки, поскольку позволяет убедиться в том, что сделанное изменение не испортило уже существующую функциональность приложения, что сокращает время поиска и исправления ошибок.