**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение

высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Отчет**

по лабораторной работе № 3 по дисциплине «**Методы и средства программной инженерии**»

Вариант 1779

Автор: Иванов Андрей Вячеславович

Факультет: ПИиКТ

Группа: P32101

Преподаватель: Цопа Евгений Алексеевич



Санкт-Петербург, 2023

**Задание:**

Написать сценарий для утилиты [Apache Ant](http://ant.apache.org/" \t "_blank), реализующий компиляцию, тестирование и упаковку в jar-архив кода проекта из [лабораторной работы №3](https://se.ifmo.ru/courses/web" \l "lab3) 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.

**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. **music** - воспроизведение музыки по завершению сборки (цель **build**).
6. **team** - осуществляет получение из git-репозитория 4 предыдущих ревизий, их сборку (по аналогии с основной) и упаковку получившихся jar-файлов в zip-архив. Сборку реализовать посредством вызова цели **build**.

**Выполнение:**

**Файл build.xml**:

<?xml version="1.0" encoding="UTF-8"?>  
<project name="web-programming-lab3" default="build">  
  
 *<!-- Загрузка параметров из файла -->* <property file="build.properties"/>  
  
 *<!-- Путь к дополнительным джарникам -->* <path id="classpath.source">  
 <fileset dir="${lib}" includes="\*.jar"/>  
 </path>  
  
 *<!-- Определение целей (targets) -->  
  
 <!-- Компиляция исходных кодов проекта -->* <target name="compile" description="Компиляция исходных кодов проекта">  
 <echo>Compilation...</echo>  
 <mkdir dir="${compiled.classes}"/>  
 <javac includeantruntime="false" srcdir="${main}" destdir="${compiled.classes}">  
 <classpath refid="classpath.source"/>  
 </javac>  
 <mkdir dir="${compiled.tests}"/>  
 <javac includeantruntime="false" srcdir="${test}" destdir="${compiled.tests}">  
 <classpath>  
 <path refid="classpath.source"/>  
 <pathelement location="${compiled.classes}"/>  
 </classpath>  
 </javac>  
 <echo>Compilation success!</echo>  
 </target>  
  
*<!-- Упаковка проекта в исполняемый JAR-архив -->* <target name="build" depends="compile" description="Упаковка скомпилированных файлов в исполняемый jar-архив">  
 <echo>Building...</echo>  
 <mkdir dir="${webapp}"/>  
 <copy todir="${webapp}">  
 <fileset dir="${web}"/>  
 </copy>  
 <mkdir dir="${webInf.classes}"/>  
 <copy todir="${webInf.classes}">  
 <fileset dir="${compiled.classes}"/>  
 </copy>  
 <mkdir dir="${webInf.lib}"/>  
 <copy todir="${webInf.lib}">  
 <fileset dir="${lib}"/>  
 </copy>  
 <buildnumber/>  
 <jar destfile="${file.jar}" basedir="${webapp}">  
 <manifest>  
 <attribute name="Version" value="${build.number}"/>  
 <attribute name="Main-Class" value="${main-class}"/>  
 <attribute name="Author" value="Ivanov Andrey"/>  
 </manifest>  
 </jar>  
 <echo>Building success!</echo>  
 </target>  
  
 *<!-- Удаление скомпилированных классов и всех временных файлов (если они есть) -->* <target name="clean" description="Удаление скомпилированных классов проекта и всех временных файлов (если они есть)">  
 <echo>Cleaning...</echo>  
 <delete dir="${target}"/>  
 <echo>Cleaning done!</echo>  
 </target>  
  
 *<!-- Запуск junit-тестов -->* <target name="test" depends="build" description="Запуск junit-тестов проекта">  
 <echo>Testing..</echo>  
 <mkdir dir="${reports}"/>  
 <junit printsummary="yes" fork="yes" haltonfailure="no">  
 <formatter type="xml"/>  
 <classpath>  
 <path refid="classpath.source"/>  
 <pathelement location="${compiled.classes}"/>  
 <pathelement location="${compiled.tests}"/>  
 </classpath>  
 <batchtest todir="${reports}">  
 <fileset dir="${test.java}">  
 <include name="\*.java" />  
 </fileset>  
 </batchtest>  
 </junit>  
 <echo>Testing complete!</echo>  
 </target>  
  
 *<!-- Воспроизведение музыки по завершению сборки -->* <target name="music" depends="build" description="Воспроизведение музыки по завершению сборки">  
 <sound>  
 <success source="${sound-success}"/>  
 <fail source="${sound-failure}"/>  
 </sound>  
 </target>  
  
*<!-- Макрос для работы с гитом-->* <macrodef name = "git">  
 <attribute name = "command" />  
 <attribute name = "dir" default = "" />  
 <attribute name = "arg" default = "" />  
 <element name = "args" optional = "true" />  
 <sequential>  
 <echo message = "git @{command} @{arg}" />  
 <exec executable = "git" dir = "@{dir}">  
 <arg value = "@{command}" />  
 <arg value = "@{arg}" />  
 <args/>  
 </exec>  
 </sequential>  
 </macrodef>  
  
  
 *<!-- Получение 4 предыдущих ревизий из git, сборка и упаковка в zip-архив -->* <target name="team" description="Получение 4 предыдущих ревизий из git, сборка и упаковка в zip-архив">  
 <mkdir dir="${team}" />  
 <echo message = "Switching to the previous commit..." />  
 <git command="checkout" arg="HEAD~1"/>  
 <antcall target="build"/>  
 <move file="${file.jar}" tofile="${file1.jar}" />  
 <move file="${file1.jar}" todir="${team}"/>  
 <echo message = "Switching to the previous commit..." />  
 <git command="checkout" arg="HEAD~1"/>  
 <antcall target="build"/>  
 <move file="${file.jar}" tofile="${file2.jar}" />  
 <move file="${file2.jar}" todir="${team}"/>  
 <echo message = "Switching to the previous commit..." />  
 <git command="checkout" arg="HEAD~1"/>  
 <antcall target="build"/>  
 <move file="${file.jar}" tofile="${file3.jar}" />  
 <move file="${file3.jar}" todir="${team}"/>  
 <echo message = "Switching to the previous commit..." />  
 <git command="checkout" arg="HEAD~1"/>  
 <antcall target="build"/>  
 <move file="${file.jar}" tofile="${file4.jar}" />  
 <move file="${file4.jar}" todir="${team}"/>  
 <echo message = "Packing the resulting .jar files into a .zip archive..." />  
 <zip destfile="${zip}">  
 <fileset dir="${team}" includes="\*.jar"/>  
 </zip>  
 <echo message = "Packing success!" />  
 </target>  
</project>

**Файл build.properties:**

compiled.classes=./target/classes  
main=./src/main  
lib=./lib  
compiled.tests=./target/tests  
test=./src/test  
webapp=./target/webapp  
web=./src/main/webapp  
webInf.classes=./target/webapp/WEB-INF/classes  
webInf.lib=./target/webapp/WEB-INF/lib  
file.jar=./target/webapp.jar  
file1.jar=./target/webapp1.jar  
file2.jar=./target/webapp2.jar  
file3.jar=./target/webapp3.jar  
file4.jar=./target/webapp4.jar  
main-class=./src/main/java/com/anvisero/lab3mispi/beans/MarkManager.java  
target=./target  
reports=reports  
test.java=./src/test/java  
sound-success=sounds/fanfaryi-podvodyaschie-itogi.wav  
sound-failure=sounds/nepravilno-poprobuy-esch-raz.wav  
team=./target/team  
zip=./target/team/webapp\_4\_previous\_commits.zip

**Вывод:**

В ходе выполнения данной лабораторной работы я познакомился с программным инструментом, используемым для автоматизации процессов сборки программного обеспечения – Apache Ant. Более того я научился автоматизировать процесс сборки проектов, а так же проводить модульное тестирование кода при помощи Junit-тестов.