Федеральное государственное автономное образовательное учреждение
высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Лабораторная работа №3 по дисциплине

«**Методы и средства программной инженерии**»

Вариант 1002

Выполнили:

Соболев Иван Александрович,

Тюрин Святослав Вячеславович

Факультет: ПИиКТ

Группа: P32312, P32302

Преподаватель:

Исаев Илья Владимирович

Санкт-Петербург, 2023

Задание:

Лабораторная работа #3

Вариант 1002

Внимание! У разных вариантов разный текст задания!

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 запускаемом классе.

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. **native2ascii** - преобразование native2ascii для копий файлов локализации (для тестирования сценария все строковые параметры необходимо вынести из классов в файлы локализации).
6. **history** - если проект не удаётся скомпилировать (цель **compile**), загружается предыдущая 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, изменённых в ревизии, следующей непосредственно за последней работающей.

Выполнение:

Репозиторий с кодом - [itmo-mispi/WebLab_3 at main · Ivanio1/itmo-mispi · GitHub]

build.xml

```
<?xml version="1.0" encoding="UTF-8"?>
 <project name="webapp" default="build">
      <property file="build.properties"/>

      <!--  указываем путь до Анта-->
      <taskdef resource="net/sf/antcontrib/antlib.xml">
      <classpath>
            <pathelement location="lib/ant-contrib-1.0b3.jar"/>
      </classpath>
      </taskdef>

      <!-- Путь к дополнительным джарникам -->
      <path id="classpath.source">
      <fileset dir="${lib}" includes="*.jar"/>
      </path>

      <!--  компиляция исходных кодов проекта-->
      <target name="compile">
      <mkdir dir="${compiled.classes}"/>
      <javac srcdir="${main}" destdir="${compiled.classes}"
encoding="UTF-8">
            <classpath>
                  <path refid="classpath.source"/>
                <pathelement location="${compiled.classes}"/>
            </classpath>
      </javac>
      <copy todir="${compiled.classes}">
            <fileset dir="${meta-inf}"/>
      </copy>
```

```
	<mkdir dir="${compiled.tests}"/>
	<javac srcdir="${test}" destdir="${compiled.tests}">
		<classpath>
			<path refid="classpath.source"/>
			<pathelement location="${compiled.classes}"/>
		</classpath>
	</javac>
	</target>

	<!--  компиляция исходных кодов проекта и их упаковка в исполняемый
jar-архив-->
	<target name="build" depends="compile">
	<delete file="${file.jar}"/>
	<mkdir dir="${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>
	<jar destfile="${file.jar}" basedir="${webapp}">
		<manifest>
			<attribute name="Version" value="1.0"/>
			<attribute name="Author" value="Ivan and Svyatoslav"/>
			<attribute name="Manage-Bean" value="PointsBean"/>
			<!--  <attribute name="Main-Class"
value="${main-class}"/>-->
		</manifest>
	</jar>
	</target>


	<!-- проверка на наличие скомпилированных классов проекта и всех
временных файлов -->
	<target name="check-condition">
	<available file="${target}" type="dir" property="check.status"/>
	</target>

	<!-- удаление скомпилированных классов проекта и всех временных файлов
(если они есть)-->
	<target name="clean" depends="check-condition">
	<if>
		<equals arg1="${check.status}" arg2="true"/>
		<then>
			<delete dir="${target}"/>
		</then>
	</if>
	</target>
```

```
      <!--запуск junit-тестов проекта-->
      <target name="test" depends="build">
      <mkdir dir="${reports}"/>
      <echo message="--- TEST DONE ---"/>
      <junit fork="true" printsummary="on">
            <formatter type="xml"/>
            <classpath>
                  <path refid="classpath.source"/>
                  <pathelement location="${compiled.classes}"/>
                  <pathelement location="${compiled.tests}"/>
            </classpath>
            <batchtest todir="reports">
                  <fileset dir="${test2}" includes="*.java"/>
            </batchtest>
      </junit>
      </target>

      <!-- преобразование native2ascii для копий файлов локализации -->
      <target name="native2ascii">
      <native2ascii encoding="${localization.encoding}"
src="${localization.files.path}"
                        dest="${ascii_locale}/resources" ext=".txt"
includes="*resources*.*"/>
      <mkdir dir="${ascii_locale}"/>
      <copy todir="${ascii_locale}">
            <fileset dir="${ascii_locale}"/>
      </copy>
      </target>

     <!-- если проект не удаётся скомпилировать (цель compile), загружается
предыдущая версия из репозитория git -->
      <target name="history">
      <exec executable="git" outputproperty="commits.count">
            <arg value="rev-list"/>
            <arg value="--count"/>
            <arg value="HEAD"/>
      </exec>

      <exec executable="git" outputproperty="current.commit">
            <arg value="rev-parse"/>
            <arg value="HEAD"/>
      </exec>

      <exec executable="git" outputproperty="current.work_diff">
            <arg value="diff"/>
            <arg value="HEAD"/>
      </exec>

      <if>
            <length string="${current.work_diff}" length="0" trim="true"
when="greater"/>
            <then>
```

```
                  <exec executable="git" >
                  <arg value="add"/>
                  <arg value="."/>
                  </exec>
                  <exec executable="git" >
                  <arg value="commit"/>
                  <arg value="-m"/>
                      <arg value="New commit"/>
                  </exec>
            </then>
            <else>
                  <echo>No changes in files. Nothing to commit</echo>
            </else>
      </if>

      <trycatch>
            <try>
                  <for param="i" begin="1" end="${commits.count}">
                  <sequential>
                        <trycatch>
                              <try>
                                <antcall target="compile"/>
                                <property name="breakProperty"
value="true"/>
                              </try>
                              <catch>
                              <exec executable="git"
outputproperty="previous.commit">
                                    <arg value="rev-parse"/>
                                <arg value="HEAD"/>
                                </exec>
                                <echo>Error in compile target</echo>
                              <exec executable="git">
                                    <arg value="reset"/>
                                    <arg value="--hard"/>
                                    <arg value="HEAD~1"/>
                                </exec>
                            </catch>
                              <!-->Если компиляция прошла успешна, выходим с
цикла<-->
                              <finally>
                              <fail if="breakProperty"/>
                            </finally>
                        </trycatch>
                  </sequential>
             </for>
            </try>
            <!-->Ловим ошибку при успешной компиляции<-->
            <catch>
            <echo>Finally</echo>
            </catch>
            <finally>
            <if>
```

```
                <isset property="previous.commit"/>
                <then>
                      <exec executable="git">
                      <arg value="diff"/>
                      <arg value="${previous.commit}"/>
                      <redirector output="diff.txt" alwayslog="true"/>
                      </exec>
                </then>
            </if>
            <echo>Original commit: ${current.commit}</echo>
            </finally>
      </trycatch>
   </target>


</project>
```

build.properties

```
lib=./lib

main=./src/main

web=./src/main/webapp

meta-inf=./src/main/resources

test=./src/test

test2=./src/test/java

target=./target

compiled.classes=./target/classes

compiled.tests=./target/tests

webapp=./target/webapp

file.jar=./target/webapp.jar

webInf.classes=./target/webapp/WEB-INF/classes

webInf.lib=./target/webapp/WEB-INF/lib

reports=reports

ascii_locale=src/main/resources/ascii_localization
```

```
localization.files.path=src/main/resources

localization.encoding=UTF-8
```

Пример файла diff.txt

```
diff --git a/diff.txt b/diff.txt
index 2b7eb3e..08e28c8 100644
--- a/diff.txt
+++ b/diff.txt
@@ -1,12 +1,67 @@
-diff --git a/src/main/java/controller/AreaResultChecker.java
b/src/main/java/controller/AreaResultChecker.java
-index deb3802..9611f9c 100644
---- a/src/main/java/controller/AreaResultChecker.java
-+++ b/src/main/java/controller/AreaResultChecker.java
-@@ -1,6 +1,6 @@
- package controller;
-
--//import model.Point;
-+import model.Point;
-
- public class AreaResultChecker {
-    private AreaResultChecker() {
```

Вывод: В ходе выполнения данной лабораторной работы мы познакомились с утилитой для автоматизации процесса сборки Apache Ant, а также научились попробовали тестировать код при помощи Junit-тестов.