

Университет ИТМО  
Факультет программной инженерии и компьютерной техники

Методы и средства программной  
инженерии.  
Лабораторная работа №3.  
Системы сборки

Смирнов Виктор Игоревич	P32131
Шиняков Артём Дмитриевич	R32372
Вариант	1009

# Содержание

<a href="#">1 Задание</a>	1
<a href="#">2 build.properties</a>	1
<a href="#">3 build.xml</a>	2
<a href="#">4 UserAccountTest.java</a>	5
<a href="#">5 Вывод</a>	6

## 1 Задание

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 запуске класса.

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. scp - перемещение собранного проекта по scp на выбранный сервер по завершению сборки. Предварительно необходимо выполнить сборку проекта (цель build)
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.

## 2 build.properties

```
1 # Java files (code itself)
2 src.dir=src
3 src.main.dir=${src.dir}/main
4 src.web.dir=${src.main.dir}/taparia-app/src/main/webapp
5
6 base.dir=.apps
7 # Compiled files
8 build.dir=.build
9 build.classes.dir=${build.dir}/classes
10 build.lib.dir=${build.dir}/lib
11 build.web.dir=${base.dir}/app
12 build.api=${base.dir}/api
13 build.api.dir=${build.api}/WEB-INF
14 build.web-inf.dir=${build.web.dir}/WEB-INF
15 webxml.file=${build.web.dir}/WEB-INF/web.xml
16 web.classes=${build.web-inf.dir}/classes
17 web-inf.path=${src.main.dir}/taparia-app/src/main/webapp/WEB-INF
18 path.to.app.war=${base.dir}/app.war
19 path.to.api.war=${base.dir}/api.war
20 # Lib files
21 lib.dir=lib
```

```

22 junit.lib=${lib.dir}/junit.jar
23 hamcrest.lib=${lib.dir}/hamcrest-core-1.3.jar
24
25 # Tests files
26 test.dir=${src.dir}/test
27 build.test.dir=${base.dir}/test
28 reports.test=${build.test.dir}/reports
29
30 # SCP properties
31 scp.host=se.ifmo.ru
32 scp.user=s337054
33 scp.dir=/home/studs/${scp.user}
34 scp.port=2222
35
36 sql.dir=${src.main.dir}/taparia-storage-jdbc/src/main/resources
37
38 ant-cont.dir=${lib.dir}/ant-contrib-1.0b3.jar

```

Листинг 1: build.properties

### 3 build.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <project name="MISPI3" default="build">
3
4     <property file="build.properties" />
5
6     <taskdef resource="net/sf/antcontrib/antcontrib.properties">
7         <classpath>
8             <pathelement location="${ant-cont.dir}"/>
9         </classpath>
10    </taskdef>
11
12    <path id="libpath">
13        <fileset dir="${lib.dir}" includes="**/*.jar" />
14    </path>
15
16    <path id="testpath">
17        <pathelement location="${libpath}" />
18        <pathelement location="${junit.lib}" />
19        <pathelement location="${hamcrest.lib}" />
20        <pathelement location="${build.test.dir}" />
21        <pathelement location="${build.classes.dir}" />
22    </path>
23
24    <target name="clean">
25
26        <echo> -----PACKAGE REMOVAL - START-----</echo>
27
28        <delete dir="${build.dir}" />
29        <delete dir="${base.dir}" />
30        <delete dir="${build.test.dir}" />
31        <delete dir="${build.api.dir}" />
32
33        <echo> -----PACKAGE REMOVAL - SUCCESS-----</echo>
34
35    </target>
36
37    <target name="init">
38
39        <echo> -----PACKAGE CREATION - START-----</echo>
40
41        <mkdir dir="${build.dir}" />
42        <mkdir dir="${base.dir}" />
43        <mkdir dir="${build.classes.dir}" />
44        <mkdir dir="${build.lib.dir}" />
45        <mkdir dir="${build.web.dir}" />
46        <mkdir dir="${build.api.dir}" />
47        <mkdir dir="${build.test.dir}" />
48        <mkdir dir="${reports.test}" />
49
50        <echo> -----PACKAGE CREATION - SUCCESS-----</echo>
51

```

```

52 </target>
53
54 <target name="compile" depends="clean, init">
55
56     <echo> -----COMPILE - START-----</echo>
57
58     <javac srcdir="${src.main.dir}"
59         destdir="${build.classes.dir}"
60         includeantruntime="false"
61         includes="**/*.java">
62
63         <classpath>
64             <path refid="libpath"/>
65         </classpath>
66
67     </javac>
68
69     <javac srcdir="${test.dir}"
70         destdir="${build.test.dir}"
71         includeantruntime="false">
72
73         <classpath>
74             <path refid="testpath" />
75         </classpath>
76
77     </javac>
78
79     <copy todir="${build.lib.dir}">
80         <fileset dir="${lib.dir}" />
81     </copy>
82
83     <copy todir="${build.classes.dir}">
84         <fileset dir="${sql.dir}" includes="**/*.sql" />
85     </copy>
86
87     <echo> -----COMPILE - SUCCESS-----</echo>
88
89 </target>
90
91 <target name="prepare-build" depends="compile">
92
93     <echo> -----PREPARATION - START-----</echo>
94
95     <copy todir="${build.web.dir}">
96         <fileset dir="${src.web.dir}" />
97     </copy>
98
99     <copy todir="${build.web-inf.dir}">
100         <fileset dir="${build.dir}" excludes="**/api/**" />
101     </copy>
102
103     <copy todir="${build.api.dir}" >
104         <fileset dir="${build.dir}" excludes="**/app/**" />
105     </copy>
106
107     <copy todir="${build.api.dir}" >
108         <fileset dir="${web-inf.path}" />
109     </copy>
110
111     <echo> -----PREPARATION - SUCCESS-----</echo>
112
113 </target>
114
115 <target name="build" depends="prepare-build">
116
117     <echo> -----BUILD PROCESS - START-----</echo>
118
119     <war destfile="${base.dir}/app.war" webxml="${webxml.file}">
120         <fileset dir="${build.web.dir}" />
121         <manifest>
122             <attribute name="Created-By" value="${user.name}" />
123             <attribute name="Manifest-Version" value="1.0" />
124             <attribute name="Main-Class" value="NoClass" />

```

```

125         </manifest>
126     </war>
127
128     <war destfile="${base.dir}/api.war" needxmlfile="false">
129         <fileset dir="${build.api}" />
130         <manifest>
131             <attribute name="Created-By" value="${user.name}" />
132             <attribute name="Manifest-Version" value="1.0" />
133         </manifest>
134     </war>
135
136     <echo> -----BUILD PROCESS - SUCCESS-----</echo>
137
138 </target>
139
140 <target name="test" depends="build">
141
142     <echo> -----TEST - START-----</echo>
143
144     <junit printsummary="yes" fork="yes" haltonfailure="yes" haltonerror="yes">
145
146         <classpath>
147             <path refid="testpath" />
148         </classpath>
149
150         <batchtest fork="yes" todir="${reports.test}">
151             <formatter type="xml" />
152             <fileset dir="${test.dir}">
153                 <include name="**/*Test*.java" />
154             </fileset>
155         </batchtest>
156
157     </junit>
158
159     <echo> -----TEST - SUCCESS-----</echo>
160
161 </target>
162 <!-- not working -->
163 <target name="scp" depends="build">
164
165     <echo> -----TRANSFER - START-----</echo>
166
167     <input message="Enter password:" addproperty="scp.password">
168         <handler type="secure" />
169     </input>
170
171     <trycatch>
172         <try>
173
174             <scp todir="${scp.user}@${scp.host}:${scp.dir}"
175                 password="${scp.password}"
176                 port="${scp.port}"
177                 trust="true">
178
179                 <fileset dir="${base.dir}" includes="*.war" />
180
181             </scp>
182
183             <echo> -----TRANSFER - SUCCESS-----</echo>
184
185         </try>
186         <catch>
187
188             <echo>-----FAILED TO TRANSFER WAR FILE-----</echo>
189
190         </catch>
191     </trycatch>
192
193 </target>
194
195 <target name="history">
196
197

```

```

198     <echo>-----LET THE HISTORY BEGIN-----</echo>
199
200     <trycatch>
201         <try>
202
203             <antcall target="compile" />
204
205             <echo>-----HISTORY SUCCESSFULLY ENDS HERE-----</
echo>
206
207         </try>
208         <catch>
209
210             <exec executable="git" outputproperty="git-head">
211                 <arg value=" log HEAD -1" />
212             </exec>
213
214             <echo>${git-head}</echo>
215
216             <exec executable="cmd" outputproperty="git-tail">
217                 <arg value="/c" />
218                 <arg value="git log --pretty=format:%h | tail -1" />
219             </exec>
220
221             <if>
222                 <equals arg1="${git-head}" arg2="${git-tail}" />
223                 <then>
224
225                     <echo>-----HISTORY FOUND THAT NO WORKING COPIES HAVE
BEEN COMMITED-----</echo>
226
227                 </then>
228                 <else>
229
230                     <exec executable="cmd" output="git_diff">
231                         <arg value="/c" />
232                         <arg value="git diff HEAD~1" />
233                     </exec>
234
235                     <exec executable="cmd">
236                         <arg value="/c" />
237                         <arg value="git reset HEAD~1 --hard" />
238                     </exec>
239
240                     <antcall target="history" />
241
242                 </else>
243             </if>
244
245         </catch>
246     </trycatch>
247
248 </target>
249
250 </project>

```

Листинг 2: build.xml

## 4 UserAccountTest.java

```

1  import org.junit.Test;
2  import ru.vityaman.itmo.web.lab.taparia.user.UserAccount;
3
4  import java.util.stream.Collectors;
5  import java.util.stream.Stream;
6
7  import static org.junit.Assert.assertArrayEquals;
8  import static org.junit.Assert.assertEquals;
9
10 public class UserAccountTest {
11     @Test
12     public void testValidUser() {
13         long id = 213;

```

```

14     String name = "tester";
15     byte[] hash = new byte[] {12, 23};
16     byte[] salt = new byte[] {1, 2};
17
18     UserAccount account = UserAccount.of(
19         UserAccount.Id.of(id),
20         UserAccount.Name.of(name),
21         UserAccount.Password.fromHashAndSalt(hash, salt)
22     );
23     assertEquals(account.id().value(), id);
24     assertEquals(account.name().value(), name);
25     assertEquals(account.password().hash(), hash);
26     assertEquals(account.password().salt(), salt);
27 }
28
29 @Test(expected = IllegalArgumentException.class)
30 public void testInvalidNameRestrictedSymbol() {
31     UserAccount.Name name = UserAccount.Name.of("lala_la");
32 }
33
34 @Test(expected = IllegalArgumentException.class)
35 public void testInvalidNameTooShort() {
36     UserAccount.Name name = UserAccount.Name.of("la");
37 }
38
39 @Test(expected = IllegalArgumentException.class)
40 public void testInvalidNameTooLong() {
41     UserAccount.Name name = UserAccount.Name.of(
42         Stream
43             .iterate("1", s -> s)
44             .limit(100)
45             .collect(Collectors.joining(","))
46     );
47 }
48
49 @Test(expected = IllegalArgumentException.class)
50 public void testInvalidIdNegative() {
51     UserAccount.Id id = UserAccount.Id.of(-12);
52 }
53
54 @Test(expected = IllegalArgumentException.class)
55 public void testInvalidIdZero() {
56     UserAccount.Id id = UserAccount.Id.of(0);
57 }
58 }

```

Листинг 3: UserAccountTest.java

## 5 Вывод

Выполнив данную лабораторную работу мы научились познакомиться с основными принципами работы систем автоматической сборки проектов на примере Ant, а также поняли, как использовать фреймворк JUnit для юнит-тестирования нашего программного продукта.