

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»**

Выполнили:

Чернова Анна Ивановна,

Болдов Олег Евгеньевич

Группа: Р32301

Преподаватель:

Барсуков Илья Александрович

Санкт-Петербург, 2023

## Вариант: 40271

### Задание варианта:

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. **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.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<project name="MispiLab3" default="compile">

  <property file="build.properties"/>

  <taskdef resource="net/sf/antcontrib/antlib.xml">

    <classpath>

      <pathelement location="lib/ant-contrib-1.0b3.jar"/>
```

```

    </classpath>

</taskdef>

<path id="classpath.source">

    <fileset dir="${lib.dir}" includes="**/*.jar"/>

</path>

<target name="native2ascii">

    <native2ascii src="${native.src}" dest="${native.dest}"
        includes="*.properties" ext=".properties"/>

</target>

<target name="compile">

    <echo message="--- COMPILE DONE ---"/>

    <mkdir dir="${compiled}"/>

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

    <mkdir dir="${compiled.tests}"/>

    <javac destdir="${compiled.classes}" includeantruntime="false">

        <src path="${main}"/>

        <classpath>

            <pathelement location="${compiled.classes}"/>

            <path refid="classpath.source"/>

        </classpath>

    </javac>

    <javac destdir="${compiled.tests}" includeantruntime="false">

        <src path="${test}"/>

        <classpath>

            <pathelement location="${compiled.classes}"/>

            <path refid="classpath.source"/>

        </classpath>

    </javac>

```

```

</target>

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

    <echo message="--- BUILD DONE ---"/>

    <jar destfile="${dest.jar}" basedir="${compiled.classes}"
        compress="false" index="true">

        <manifest>

            <attribute name="Main-Class" value="${main.class}"/>

        </manifest>

    </jar>
</target>

<target name="clean">

    <echo message="--- CLEAN DONE ---"/>

    <delete dir="${compiled}"/>

    <delete dir="${diff}"/>

    <delete dir="${repots.test}"/>

    <delete dir="${native.dest}"/>
</target>

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

    <echo message="--- TEST DONE ---"/>

    <delete dir="${repots.test}"/>

    <mkdir dir="${repots.test}"/>

    <junit fork="true" printsummary="on">

        <formatter type="xml"/>

        <classpath>

            <path refid="classpath.source"/>

            <pathelement location="${compiled.classes}"/>

            <pathelement location="${compiled.tests}"/>

        </classpath>

```

```

<batchtest todir="${repots.test}">

    <fileset dir="${test}" includes="*Test*.java"/>

</batchtest>

</junit>

</target>

<target name="history">

    <trycatch>

        <try>

            <runtarget target="compile"/>

            <echo message="--- NO ERRORS, COMPILE DONE ---"/>

        </try>

        <catch>

            <exec executable="cmd" outputproperty="git_log_head">

                <arg value="/c"/>

                <arg value="git log --pretty=format:%h - %an, %ar : %s | head
-1"/>

            </exec>

            <exec executable="cmd" outputproperty="git_log_tail">

                <arg value="/c"/>

                <arg value="git log --pretty=format:%h - %an, %ar : %s | tail
-1"/>

            </exec>

            <if>

                <equals arg1="${git_log_head}" arg2="${git_log_tail}"/>

                <then>

                    <echo message="I cannot build the first commit. Stop"/>

                </then>

                <else>

                    <exec executable="cmd" output="${diff}">

```

```
        <arg value="/c"/>

        <arg value="git diff HEAD~1"/>

    </exec>

    <exec executable="cmd">

        <arg value="/c"/>

        <arg value="git reset HEAD~1 --hard"/>

    </exec>

    <runtarget target="history"/>

</else>

</if>

</catch>

</trycatch>

</target>

</project>
```

### **Вывод:**

В результате выполнения данной лабораторной работы мы изучили основные понятия систем сборки и тестирования. Узнали о make, ant, ivy, maven, gradle.

Также попытались протестировать нашу программу через Junit-тестов.