

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
“Санкт-Петербургский национальный исследовательский
университет
информационных технологий, механики и оптики”

Лабораторная работа № 3
по дисциплине
“МЕТОДЫ И СРЕДСТВА ПРОГРАММНОЙ ИНЖЕНЕРИИ”

Выполнили:
Студент группы R32372
Аракчиева
Ева Дмитриевна

Преподаватель:
Райла Мартин

Университет ИТМО
Санкт-Петербург
31.05.2023

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. `xml` - валидация всех xml-файлов в проекте.
6. `doc` - добавление в MANIFEST.MF MD5 и SHA-1 файлов проекта, а также генерация и добавление в архив javadoc по всем классам проекта.

Вопросы к защите лабораторной работы:

1. Тестирование ПО. Цель тестирования, виды тестирования.
2. Модульное тестирование, основные принципы и используемые подходы.
3. Пакет JUnit, основные API.
4. Системы автоматической сборки. Назначение, принципы работы, примеры систем.
5. Утилита `make`. Make-файлы, цели и правила.
6. Утилита `Ant`. Сценарии сборки, цели и команды.

```
<?xml version="1.0"?>

<project name="main.Main" default="build">

  <property file="build.properties"/>

  <path id="classpath.source">
    <fileset dir="${lib.dir}" includes="**/*.jar"/>
  </path>

  <macrodef name="generate-hash" description="Генерация хешей файлов
  проекта и запись в MANIFEST.MF">
    <attribute name="algorithm"/>
    <sequential>
      <checksum file="${build}/${ant.project.name}.jar"
        property="@{algorithm}"/>
    </sequential>
  </macrodef>

  <target name="compile" description="Компиляция исходного кода">
    <echo>Компиляция исходного кода...</echo>
    <mkdir dir="${classes}/src"/>
    <javac srcdir="${src.dir}"
      destdir="${classes}/src"
      source="8"
      target="8"
      includeantruntime="false">
    <classpath>
      <path refid="classpath.source"/>
    </classpath>
  </javac>
</target>

  <target name="build" depends="compile" description="Компиляция
  исходных кодов проекта и их упаковка в исполняемый jar-архив">
    <echo>Упаковка в jar-архив...</echo>
    <mkdir dir="${build}"/>
    <jar destfile="${build}/${ant.project.name}.jar"
      basedir="${classes}/src"
      update="true"
      excludes="Test.class">
    <manifest>
      <attribute name="Main-Class" value="main.Main"/>
    </manifest>
  </jar>
</target>
```

```
<target name="test" depends="build" description="Компиляция и
выполнение тестов">
<echo>Компиляция тестов...</echo>
<mkdir dir="${classes}/test_results"/>
<javac srcdir="${test.dir}"
destdir="${classes}/test_results"
source="8"
target="8"
includeantruntime="false">
<classpath>
<path refid="classpath.source"/>
<pathelement location="${classes}/src"/>
</classpath>
</javac>
<echo>Выполнение тестов...</echo>
<junit printsummary="yes" haltonfailure = "yes">
<classpath>
<path refid="classpath.source"/>
<pathelement location="${classes}/src"/>
<pathelement location="${classes}/test_results"/>
</classpath>
<formatter type="plain"/>
<formatter type="xml"/>
<test name="tests.Tests"/>
</junit>

</target>

<target name="xml" description="Валидация всех xml-файлов в
проекте">
<echo>Валидация всех xml-файлов в проекте...</echo>
<xmlvalidate failonerror="true">
<fileset dir="${src.dir}" includes="**/*.xml"/>
</xmlvalidate>
</target>

<target name="doc" depends="build" description="Генерация javadoc
и добавление в MANIFEST.MF MD5 и SHA-1 файлов проекта">
<mkdir dir="${doc.dir}"/>
<echo>Добавление хешей файлов проекта в MANIFEST.MF...</echo>
<generate-hash algorithm="MD5"/>
<generate-hash algorithm="SHA-1"/>
<echo>Генерация javadoc...</echo>
<javadoc sourcepath="${src.dir}"
destdir="${doc.dir}"
classpathref="classpath.source">
<fileset dir="${src.dir}" includes="*.java"/>
</javadoc>
```

```
<javadoc destdir="${doc.dir}">
<classpath>
<path refid="classpath.source"/>
</classpath>
<fileset dir="${src.dir}"/>
</javadoc>
<echo>Сборка jar с javadoc...</echo>
<jar destfile="${doc.dir}/${ant.project.name}.jar"
basedir="${doc.dir}"
update="true">
</jar>
</target>

<target name="clean" description="Удаление скомпилированных
классов проекта и всех временных файлов (если они есть)">
<echo>Удаление директории с классами...</echo>
<delete dir="${build}"/>
<echo>Удаление директории с тестами...</echo>
<delete dir="${classes}/test_results"/>
<echo>Удаление директории с документами...</echo>
<delete dir="${doc.dir}"/>
</target>

</project>
```