

Университет ИТМО

Факультет программной инженерии и компьютерной техники
Направление подготовки 09.03.04 Программная инженерия
Дисциплина «Методы и средства программной инженерии»

Отчет

По лабораторной работе №2
Вариант 1347

Выполнили:

Колесникова С.А.

Степанов М.А.

Преподаватель:

Каюков И. А.

Санкт-Петербург, 2021 г.

Задание:

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 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. **music** - воспроизведение музыки по завершению сборки (цель **build**).
6. **report** - в случае успешного прохождения тестов сохраняет отчет junit в формате xml, добавляет его в репозиторий git и выполняет commit.

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" encoding="UTF-8"?>
<project name="webapp" default="build">

  <property file="build.properties"/>

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

  <path id="libs">
    <fileset dir="lib">
      <include name="*.jar"/>
    </fileset>
  </path>

  <target name="compile">

    <mkdir dir="${target}/classes"/>
    <javac srcdir="${src}/java" destdir="${target}/classes" includeantruntime="false"
      encoding="utf-8">
      <classpath>
        <path refid="libs"/>
      </classpath>
    </javac>

    <copy todir="${target}/classes">
      <fileset dir="${src}/resources"/>
    </copy>

  </target>

</project>
```

```
<mkdir dir="${target}/webapp"/>

<copy todir="${target}/webapp">
  <fileset dir="${src}/webapp"/>
</copy>
/target>

target name="build" depends="compile">
  <delete file="${file.jar}"/>
  <jar destfile="${file.jar}" basedir="${target}/classes" >
    <manifest>
      <attribute name="Main-Class" value="Main"/>
    </manifest>
  </jar>
  <sound>
    <success source="sounds" loops="10"/>
    <fail source="sounds" loops="10"/>
  </sound>
/target>

target name="clean">
  <delete dir="${target}"/>
/target>
```

```

<target name="test" depends="build">
  <mkdir dir="${target}/tests"/>
  <javac srcdir="${tests}/java" destdir="${target}/tests" includeantruntime="false">
    <classpath>
      <path refid="libs"/>
      <pathelement location="${target}/classes"/>
    </classpath>
  </javac>

  <mkdir dir="reports"/>
  <junit fork="true" printsummary="on" failureproperty="tests.failed" >
    <formatter type="xml"/>
    <classpath>
      <pathelement location="${target}/classes"/>
      <pathelement location="${target}/tests"/>
    </classpath>
    <batchtest todir="reports">
      <fileset dir="${test}/java" includes="*Test*.java"/>
    </batchtest>
  </junit>

</target>

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

</target>

<target name="report" depends="test" unless="${tests.failed}">

```

```

  <target name="report" depends="test" unless="${tests.failed}">

    <exec executable="git">
      <arg value="add"/>
      <arg value="reports/*.xml"/>
    </exec>

    <exec executable="git">
      <arg value="commit"/>
    </exec>

  </target>

</project>

```

github: <https://github.com/stmikeal/MISPI3>

Выводы:

Во время выполнения лабораторной работы мы изучили системы сборки приложений, написали собственные сценарии сборки с помощью утилиты Apache Ant. Также мы написали модульные тесты на основе библиотеки JUnit.