

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное
образовательное учреждение высшего образования
“Национальный исследовательский университет ИТМО”
ФКТиУ, Кафедра Информатики и вычислительной техники

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3
“Методы и средства программной инженерии”
Вариант № 342

Выполнили:
Студенты группы Р3232
Куценко А. В.
Федоров Н. С.

Преподаватель:
Покид А. В.

Санкт-Петербург
2021 г.

Задание

Вариант 342

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. **team** - осуществляет получение из git-репозитория 3 предыдущих ревизий, их сборку (по аналогии с основной) и упаковку получившихся jar-файлов в zip-архив. Сборку реализовать посредством вызова цели **build**.

Цели сценария

Compile

```
<target name="compile" depends="resolve">
  <echo>Compile</echo>
  <mkdir dir="${classes}" />
  <javac destdir="${classes}" includeantruntime="false">
    <src path="${main}" />
    <exclude name="*/package-info.java"/>
    <classpath refid="build-path" />
  </javac>
```

Build

```
<target name="build" depends="compile">
  <echo>Build</echo>
  <mkdir dir="${out}"/>
  <war destfile="${out}/${ant.project.name}.war"
webxml="${main}/webapp/WEB-INF/web.xml">
  <classes dir="${classes}"/>
  <lib dir="${lib}"/>
  <fileset dir="${main}/webapp"/>
  <manifest>
    <attribute name="Created-By" value="Web" />
    <attribute name="Manifest-Version" value="1.0" />
    <attribute name="Main-Class" value="NoClass" />
  </manifest>
</war>
  <echo>End Build</echo>
</target>
```

Clean

```
<target name="clean">
  <echo>Clean</echo>
  <delete dir="${build}" />
  <delete dir="${tests}" />
  <delete dir="${report}"/>
  <echo>End Clean</echo>
</target>
```

Test

```
<target name="test" depends="build">
  <echo>Start Testing</echo>
  <junit haltonfailure="yes" printsummary="yes">
    <classpath>
      <pathelement path="${classes}"/>
      <path refid="build-path"/>
      <pathelement location="${tests}"/>
    </classpath>
    <test name="${testName}" todir="${report}"/>
  </junit>
  <echo>Continue Testing</echo>
</target>
```

Team

```
<target name="team">
  <echo>Task Team Start</echo>
  <mkdir dir="team" />

  <echo>Take commit 1</echo>
  <exec executable="git">
    <arg value="checkout" />
    <arg value="--no-commit" />
    <arg value="HEAD~1" />
  </exec>
  <antcall target="build"/>
  <copy file="${out}/Web3.jar" tofile="team/Web3.1.jar" overwrite="false" />

  <echo>Take commit 2</echo>
  <exec executable="git">
    <arg value="checkout" />
    <arg value="--no-commit" />
    <arg value="HEAD~1" />
  </exec>
  <antcall target="build"/>
  <copy file="${out}/Web3.jar" tofile="team/Web3.2.jar" overwrite="false" />

  <echo>Take commit 3</echo>
  <exec executable="git">
    <arg value="checkout" />
    <arg value="--no-commit" />
    <arg value="HEAD~1" />
  </exec>
  <antcall target="build"/>
  <copy file="${out}/Web3.jar" tofile="team/Web3.3.jar" overwrite="false" />

  <echo>Zipping</echo>
  <zip destfile="team/team.zip"
```

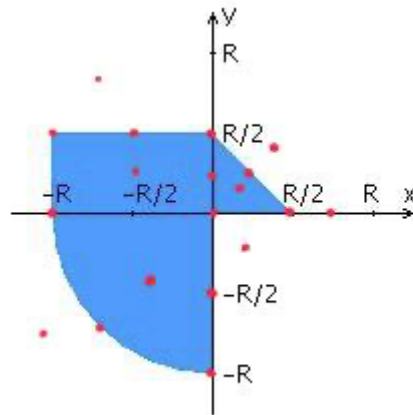
```
        basedir="team/" />
    <exec executable="git">
        <arg value="checkout" />
        <arg value="master" />
    </exec>
    <echo>Task Team End</echo>
</target>
```

native2ascii

```
<target name="native2ascii">
    <native2ascii dest="localizationascii" />
</target>
```

Пример тестов

Исходная область:



```
public class DataTest {

    private Data data;

    public DataTest() {
    }

    @Before
    public void init() {
        this.data = new Data();
    }

    @Test
    public void emptyPoint() {
        assertFalse(data.check());
    }

    @Test
    public void invalidPoint() {
        double x=-1.6, y=9, r=1;
        data.setX(x);
        data.setY(y);
        data.setR(r);
        assertFalse(data.check());
    }

    @Test
    public void center() {
        double x=0, y=0, r=2.5;
        data.setX(x);
        data.setY(y);
        data.setR(r);
        assertTrue(data.check());
    }
}
```

Вывод

В ходе лабораторной работы были получены базовые навыки работы с системами автоматической сборки проектов на примере Ant.



Выполнение :

[x] war -> jar

[x] нарисовать точки для тестов

Защита :