

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

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”

Вариант №1

Студенты:

Чухно Матвей
Романович

Воробьев Кирилл Олегович

Группа P3230 & P3231

Преподаватель:

Письмак Алексей Евгеньевич



Санкт-Петербург, 2022

Цель работы:

Написать сценарий для утилиты Apache Ant, реализующий компиляцию, тестирование и упаковку в jar-архив кода проекта из лабораторной работы №3 по дисциплине "Веб-программирование".

- **compile** -- компиляция исходных кодов проекта.
- **build** -- компиляция исходных кодов проекта и их упаковка в исполняемый jar-архив. Компиляцию исходных кодов реализовать посредством вызова цели **compile**.
- **clean** -- удаление скомпилированных классов проекта и всех временных файлов.
- **test** -- запуск junit-тестов проекта. Перед запуском тестов необходимо осуществить сборку проекта (цель **build**).
- **doc** - добавление в MANIFEST.MF MD5 и SHA-1 файлов проекта, а также генерация и добавление в архив javadoc по всем классам проекта.
- **alt** - создаёт альтернативную версию программы с измененными именами переменных и классов (используя задание replace в файлах параметров) и упаковывает её в jar-архив. Для создания jar-архива использует цель **build**.

compile:

```
<target name="compile">

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

    <javac srcdir="${src.dir}" nowarn="on" destdir="${classes.dir}" classpathref="classpath"
includeantruntime="false"/>

</target>
```

build:

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

    <mkdir dir="${jar.dir}"/>

    <jar destfile="${jar.dir}/${ant.project.name}.jar" basedir="${classes.dir}"
excludes="**/*Test.class">

        <manifest>

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

        </manifest>

    </jar>

</target>
```

</jar>

</target>

clean:

<target name="clean">

<delete dir="\${build.dir}"/>

</target>

doc:

<target name="doc">

<mkdir dir="\${docs.dir}"/>

<javadoc

 packagenames="com.example.Web3.*"

 destdir="\${docs.dir}/api"

 sourcepath="src/main/java"

 classpathref="classpath">

 <group title="Bean Package" packages="com.example.Web3.beans/*"/>

 <group title="Dao Package" packages="com.example.Web3.dao/*"/>

 <group title="Entity Package" packages="com.example.Web3.entity/*"/>

 <group title="Service Package" packages="com.example.Web3.services/*"/>

 <group title="Util Package" packages="com.example.Web3.utils/*"/>

</javadoc>

<checksum file="\${jar.dir}/\${main.class}.jar" algorithm="MD5" fileext="md5"/>

<checksum file="\${jar.dir}/\${main.class}.jar" algorithm="SHA-1" fileext="sha1"/>

<loadfile property="md5" srcfile="\${md5.file}"/>

<loadfile property="sha1" srcfile="\${sha1.file}"/>

<jar update="true" destfile="\${jar.dir}/\${ant.project.name}.jar">

 <manifest>

```
        <attribute name="MD5" value="\${md5}"/>

        <attribute name="SHA-1" value="\${sha1}"/>

    </manifest>

    <fileset dir="\${docs.dir}"/>

</jar>

</target>
```

test:

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

    <junit printsummary="yes">

        <formatter type="brief" usefile="false"/>

        <classpath>

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

        </classpath>

        <test name="com.example.Web3.utils.HitCheckerTest"/>

        <test name="com.example.Web3.utils.ValidatorTest"/>

    </junit>

</target>
```

alt:

```
<target name="alt">

    <copy todir="\${src_new.dir}">

        <fileset dir="\${src.dir}"/>

    </copy>

    <property name="src_old_path.dir" value="\${src.dir}"/>

    <propertyreset name="src.dir" value="\${src_new.dir}"/>

</target>
```

```
<replace dir="${src.dir}/${project.structure}/beans" token="Result"
value="Consequence"/>
```

```
<replace dir="${src.dir}/${project.structure}/beans" token="result"
value="consequence"/>
```

```
<replace dir="${src.dir}/${project.structure}/entity" token="Result"
value="Consequence"/>
```

```
<replace dir="${src.dir}/${project.structure}/entity" token="result"
value="consequence"/>
```

```
<replace dir="${src.dir}/${project.structure}/services" token="Result"
value="Consequence"/>
```

```
<replace dir="${src.dir}/${project.structure}/services" token="result"
value="consequence"/>
```

```
<replace dir="${src.dir}/${project.structure}/dao" token="Result" value="Consequence"/>
```

```
<replace dir="${src.dir}/${project.structure}/dao" token="result" value="consequence"/>
```

```
<replace dir="${src.dir}/${project.structure}/utils" token="Result"
value="Consequence"/>
```

```
<replace dir="${src.dir}/${project.structure}/utils" token="result" value="consequence"/>
```

```
<replace dir="${src.dir}/main/test/java/com/example/Web3/utils" token="Result"
value="Consequence"/>
```

```
<replace dir="${src.dir}/main/test/java/com/example/Web3/utils" token="result"
value="consequence"/>
```

```
<move file="${src.dir}/${project.structure}/beans/ResultBean.java"
```

```
  tofile="${src.dir}/${project.structure}/beans/ConsequenceBean.java"/>
```

```
<move file="${src.dir}/${project.structure}/entity/Result.java"
```

```
  tofile="${src.dir}/${project.structure}/entity/Consequence.java"/>
```

```
<move file="${src.dir}/${project.structure}/services/ResultService.java"
```

```
  tofile="${src.dir}/${project.structure}/services/ConsequenceService.java"/>
```

```
<move file="${src.dir}/${project.structure}/dao/ResultDao.java"
```

```
  tofile="${src.dir}/${project.structure}/services/ConsequenceDao.java"/>
```

```
<delete dir="${classes.dir}"/>
```

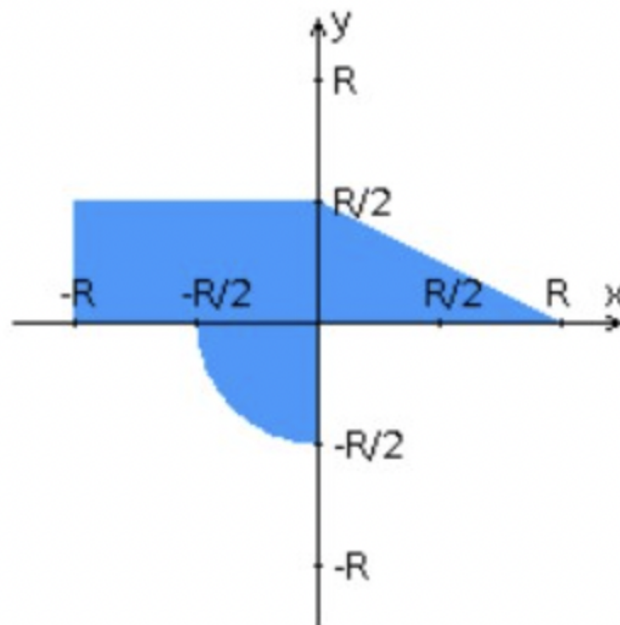
```
<antcall target="build"/>
```

```
<delete dir="${src_new.dir}"/>
```

```
<propertyreset name="src.dir" value="${src_old_path.dir}"/>
```

```
</target>
```

Тесты:



Тест, проверяющий первый квадрант

```
public void test1() {  
    for (double x = 0; x <= r; x += 0.1) {  
        for (double y = (-x * 0.5) + (r / 2); y > 0; y += -0.1) {  
            assertTrue(hitChecker.isHit(x, y, r));  
        }  
    }  
}
```

Тест, проверяющий второй квадрант

```
public void test2() {  
    for (double x = -r; x <= 0; x += 0.1) {  
        for (double y = 0; y <= (r / 2); y += 0.1) {  
            assertTrue(hitChecker.isHit(x, y, r));  
        }  
    }  
}
```

Тест, проверяющий третий квадрант

```
public void test3() {  
    for (double x = (-r / 2); x <= 0; x += 0.1) {  
        for (double y = Math.sqrt((Math.pow(r, 2) / 2) - Math.pow(x, 2)); y <= 0; y += 0.1) {  
            assertTrue(hitChecker.isHit(x, y, r));  
        }  
    }  
}
```

Вывод:

В данной лабораторной работе мы познакомились с императивной системой сборки Apache Ant , написали сценарии для сборки, компиляции, запуска тестов, генерации Javadoc и изменения названия переменных и классов. Также написали свои первые модульные тесты и узнали про различные виды тестирования