Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет Программной инженерии и компьютерной техники

Лабораторная работа по

Основам программной инженерии №3

Вариант 202111

Работу выполнила:

Касьяненко В.М.

Группа:

P3220

Санкт-Петербург,

2024

Оглавление

[Текст задания 3](#_Toc164124627)

[Реализация 4](#_Toc164124628)

[Вывод 10](#_Toc164124629)

## Текст задания



## Реализация

build.properties

lib.dir=lib  
junit=${lib.dir}/junit-4.13.2.jar  
hamcrest=${lib.dir}/hamcrest-core-1.3.jar  
contrib=${lib.dir}/ant-contrib-1.0b3.jar  
  
src.dir=src/main  
main.dir=${src.dir}/java  
web.dir=${src.dir}/webapp  
resources.dir=${src.dir}/resources  
native2ascii.resources=${resources.dir}/native2ascii  
  
ant.dir=ant  
build.dir=${ant.dir}/build  
build.dir.lib=${build.dir}/WEB-INF/lib  
build.web.xml=${build.dir}/WEB-INF/web.xml  
classes.dir=${build.dir}/WEB-INF/classes  
  
test.classes.dir=${build.dir}/test/classes  
test.dir=src/test/java  
junit.report.dir=reports  
  
scp.dir=C:/Users/Pelmeshka/Documents/wildfly-21.0.0.Final/standalone/deployments/  
env.script=C:/Users/Pelmeshka/Documents/wildfly-21.0.0.Final/bin/standalone.bat  
  
compile.version=11  
vm.args=  
  
report.dir=./report  
svn.dir=./work  
build=./build  
classes=${build}/classes  
report.name=test.xml  
testName=DataTest

build.xml

<?xml version="1.0" encoding="UTF-8"?>  
  
<project name="lab3" default="build">  
  
 <property file="build.properties"/>  
  
 <path id="classpath">  
 <fileset dir="${lib.dir}" includes="\*.jar"/>  
 <fileset dir="${main.dir}" includes="\*.java"/>  
 </path>  
  
 <path id="classpath.test">  
 <pathelement location="${junit}"/>  
 <pathelement location="${hamcrest}"/>  
 <pathelement location="${classes.dir}"/>  
 </path>  
  
 <taskdef resource="net/sf/antcontrib/antcontrib.properties">  
 <classpath>  
 <pathelement location="${contrib}"/>  
 </classpath>  
 </taskdef>  
  
 <taskdef resource="net/sf/antcontrib/antlib.xml">  
 <classpath>  
 <pathelement location="${contrib}"/>  
 </classpath>  
 </taskdef>  
  
 <target name="compile" depends="clean">  
 <echo message="\*\*\*\*\* COMPILE STARTED \*\*\*\*\*"/>  
 <mkdir dir="${classes.dir}"/>  
 <mkdir dir="${test.classes.dir}"/>  
 <javac srcdir="${src.dir}" destdir="${classes.dir}" classpathref="classpath" includeantruntime="false"/>  
 <echo message="\*\*\*\*\* COMPILE COMPLETED \*\*\*\*\*"/>  
 </target>  
  
 <target name="build" depends="compile">  
 <echo message="\*\*\*\*\* BUILD STARTED \*\*\*\*\*"/>  
 <copy todir="${build.dir}">  
 <fileset dir="${web.dir}"/>  
 </copy>  
 <copy todir="${build.dir.lib}">  
 <fileset dir="${lib.dir}"/>  
 </copy>  
 <copy todir="${classes.dir}">  
 <fileset dir="${resources.dir}"/>  
 </copy>  
 <jar destfile="${build.dir}/${ant.project.name}.jar">  
 <fileset dir="${classes.dir}"/>  
 <manifest>  
 <attribute name="Created-By" value="Pelmeshka"/>  
 <attribute name="Manifest-Version" value="1.0"/>  
 <attribute name="Main-Class" value="Main"/>  
 </manifest>  
 </jar>  
 <war destfile="${build.dir}/${ant.project.name}.war" webxml="${build.web.xml}">  
 <fileset dir="${build.dir}"/>  
 <manifest>  
 <attribute name="Created-By" value="Pelmeshka"/>  
 <attribute name="Manifest-Version" value="1.0"/>  
 <attribute name="Main-Class" value="NoClass"/>  
 </manifest>  
 </war>  
 <echo message="\*\*\*\*\* BUILD COMPLETED \*\*\*\*\*"/>  
 </target>  
  
 <target name="clean">  
 <echo message="\*\*\*\*\* CLEAN STARTED \*\*\*\*\*"/>  
 <delete dir="${build.dir}"/>  
 <delete dir="${junit.report.dir}"/>  
 <echo message="\*\*\*\*\* CLEAN COMPLETED \*\*\*\*\*"/>  
 </target>  
  
 <target name="test" depends="build">  
 <echo message="\*\*\*\*\* TEST STARTED \*\*\*\*\*"/>  
 <mkdir dir="${test.classes.dir}"/>  
 <mkdir dir="${junit.report.dir}"/>  
 <javac destdir="${test.classes.dir}" srcdir="${test.dir}" includeantruntime="false" encoding="utf-8">  
 <classpath refid="classpath.test"/>  
 </javac>  
 <junit printsummary="on" haltonfailure="true" haltonerror="true">  
 <classpath>  
 <path refid="classpath.test"/>  
 <pathelement location="${test.classes.dir}"/>  
 </classpath>  
 <batchtest fork="yes" todir="${junit.report.dir}">  
 <formatter type="xml"/>  
 <fileset dir="${test.dir}" includes="\*Test.java"/>  
 </batchtest>  
 </junit>  
 <echo message="\*\*\*\*\* TEST COMPLETED \*\*\*\*\*"/>  
 </target>  
  
 <target name="xml">  
 <echo message="\*\*\*\*\* XML STARTED \*\*\*\*\*"/>  
 <echo>XML project files validation...</echo>  
 <xmlvalidate lenient="true">  
 <fileset dir="./" includes="\*\*/\*.xml">  
 <exclude name=".idea/\*\*"/>  
 </fileset>  
 </xmlvalidate>  
 <echo message="\*\*\*\*\* XML COMPLETED \*\*\*\*\*"/>  
 </target>  
  
 <target name="native2ascii">  
 <echo message="\*\*\*\*\* NATIVE2ASCII STARTED \*\*\*\*\*"/>  
 <delete dir="${native2ascii.resources}"/>  
 <native2ascii src="${resources.dir}"  
 dest="${native2ascii.resources}"  
 includes="\*\*/\*.loc"/>  
 <echo message="\*\*\*\*\* NATIVE2ASCII COMPLETED \*\*\*\*\*"/>  
 </target>  
  
 <target name="env" depends="clean">  
 <echo message="\*\*\*\*\* ENV STARTED \*\*\*\*\*"/>  
  
 <mkdir dir="${classes.dir}"/>  
 <javac srcdir="${main.dir}" destdir="${classes.dir}" classpathref="classpath" source="${compile.version}"  
 includeantruntime="false">  
 <compilerarg line="${vm.args}"/>  
 </javac>  
 <copy todir="${classes.dir}">  
 <fileset dir="${resources.dir}"/>  
 </copy>  
 <antcall target="build"/>  
 <copy file="${build.dir}/lab3.war" todir="${scp.dir}"/>  
 <exec executable="cmd">  
 <arg line="/c start ${env.script}"/>  
 </exec>  
 <echo message="\*\*\*\*\* ENV COMPLETED \*\*\*\*\*"/>  
 </target>  
  
 <target name="report" depends="test">  
 <echo message="\*\*\*\*\* REPORT STARTED \*\*\*\*\*"/>  
  
 <condition property="tests.successful">  
 <not>  
 <isset property="tests.failed" />  
 </not>  
 </condition>  
  
 <if>  
 <isset property="tests.successful" />  
 <then>  
 <junitreport todir="${junit.report.dir}">  
 <fileset dir="${junit.report.dir}" includes="TEST-\*.xml" />  
 <report todir="${junit.report.dir}" />  
 </junitreport>  
  
 <move todir="${svn.dir}">  
 <fileset dir="${junit.report.dir}" includes="TESTS-\*.xml"/>  
 </move>  
  
 <exec executable="svn">  
 <arg value="add" />  
 <arg value="${svn.dir}/\*" />  
 </exec>  
 <exec executable="svn">  
 <arg value="update" />  
 <arg value="${svn.dir}"/>  
 </exec>  
 <exec executable="svn">  
 <arg value="commit" />  
 <arg value="${svn.dir}"/>  
 <arg value="-m" />  
 <arg value="Update report" />  
 </exec>  
 </then>  
 <else>  
 <echo message="Tests were not successful. Skipping report generation." />  
 </else>  
 </if>  
 <echo message="\*\*\*\*\* REPORT COMPLETED \*\*\*\*\*"/>  
 </target>  
</project>

## Вывод

Во время выполнения лабораторной работы были написаны собственные сценарии сборки с помощью утилиты Apache Ant, а также модульные тесты на основе библиотеки JUnit для лабораторной работы по веб-программированию.