*Университет ИТМО*

Факультет Программной Инженерии и Компьютерной техники

Основы программной инженерии

Лабораторная работа №3

Вариант №1594

Выполнил:

*Онишков Валерий Вячеславович., P3314*

Преподаватель:

Бострикова Д.К.

Задание

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Реализация

*<!-- <!DOCTYPE project SYSTEM "build.dtd"> -->*<project name="Web3" default="compile" basedir="./">  
  
 *<!-- Include variables and constants -->* <property file="build.properties"/>  
 <property name="src.encoding" value="UTF-8"/>  
 *<!-- Define the classpath for the project -->* <path id="project.classpath">  
 <fileset dir="${lib.dir}">  
 <include name="\*\*/\*.jar"/>  
 </fileset>  
 </path>  
  
  
 *<!-- Clean the build directory -->* <target name="clean">  
 <delete dir="${build.dir}"/>  
 </target>  
  
 *<!-- Create the build directory structure -->* <target name="init">  
 <mkdir dir="${build.dir}"/>  
 <mkdir dir="${classes.dir}"/>  
 </target>  
  
 *<!-- [default] Compile project -->* <target name="compile" depends="init, native2ascii">  
 <javac srcdir="${src.dir}" destdir="${classes.dir}" includeantruntime="false" classpath="project.classpath" encoding="UTF-8">  
 <classpath refid="project.classpath"/>  
 <compilerarg line="-Xlint:unchecked"/>  
 </javac>  
 <copy todir="${classes.dir}">  
 <fileset dir="${resources.dir}">  
 <include name="\*\*/\*"/>  
 </fileset>  
 </copy>  
 </target>  
  
 *<!-- Build the project and package it into a WAR file -->* <target name="build" depends="compile">  
 <copy todir="${build.dir}/webapp">  
 <fileset dir="${webapp.dir}">  
 <include name="\*\*/\*"/>  
 </fileset>  
 </copy>  
 <copy todir="${build.dir}/webapp/WEB-INF/classes">  
 <fileset dir="${classes.dir}">  
 <include name="\*\*/\*"/>  
 </fileset>  
 </copy>  
 <copy todir="${build.dir}/webapp/WEB-INF/lib">  
 <fileset dir="${lib.dir}">  
 <include name="\*.jar"/>  
 </fileset>  
 </copy>  
 <war destfile="${build.dir}/${war.name}" webxml="${webapp.dir}/WEB-INF/web.xml">  
 <manifest>  
 <attribute name="Manifest-Version" value="1.0"/>  
 <attribute name="Version" value="1.0-SNAPSHOT"/>  
 </manifest>  
 <fileset dir="${build.dir}/webapp">  
 <include name="\*\*/\*"/>  
 <exclude name="\*\*/WEB-INF/web.xml"/>  
 </fileset>  
 </war>  
 </target>  
  
  
 *<!-- Classpath for JUnit tests -->* <path id="test.classpath">  
 <path refid="project.classpath"/>  
 <pathelement location="${build.dir}/webapp/WEB-INF/classes"/>  
 <pathelement location="${build.dir}/test-classes"/>  
 <fileset dir="${lib.dir}">  
 <include name="\*\*/\*.jar"/>  
 </fileset>  
 </path>  
  
 *<!-- Test isInArea function using Junit -->* <target name="test" depends="build">  
 <mkdir dir="${build.dir}/test-classes"/>  
 <javac srcdir="${test.dir}" destdir="${build.dir}/test-classes" includeantruntime="false"  
 classpathref="test.classpath">  
 <compilerarg line="-Xlint:unchecked"/>  
 </javac>  
  
 <mkdir dir="${build.dir}/test-results"/>  
  
 *<!-- Определяем JUnit task для JUnit 4 -->* <taskdef name="junit" classname="org.apache.tools.ant.taskdefs.optional.junit.JUnitTask">  
 <classpath>  
 <path refid="test.classpath"/>  
 <fileset dir="${lib.dir}">  
 <include name="junit-\*.jar"/>  
 <include name="hamcrest-\*.jar"/>  
 </fileset>  
 </classpath>  
 </taskdef>  
  
 <echo message="Запуск JUnit тестов..."/>  
  
 *<!-- Запуск тестов с JUnit 4 -->* <junit printsummary="yes"  
 haltonfailure="false"  
 haltonerror="false"  
 fork="yes"  
 showoutput="true">  
  
 <classpath>  
 <path refid="test.classpath"/>  
 <pathelement location="${build.dir}/test-classes"/>  
 <fileset dir="${lib.dir}">  
 <include name="junit-\*.jar"/>  
 <include name="hamcrest-\*.jar"/>  
 </fileset>  
 </classpath>  
  
 *<!-- Форматировщик для вывода в консоль -->* <formatter type="plain" usefile="false"/>  
 *<!-- Форматировщик для сохранения в файлы -->* <formatter type="plain" usefile="true"/>  
 <formatter type="xml" usefile="true"/>  
  
 <batchtest todir="${build.dir}/test-results">  
 <fileset dir="${build.dir}/test-classes" includes="\*\*/\*Test.class"/>  
 </batchtest>  
 </junit>  
  
 <echo message=""/>  
 <echo message="========================================"/>  
 <echo message="Тесты завершены!"/>  
 <echo message="Подробные отчеты сохранены в: ${build.dir}/test-results"/>  
 <echo message="========================================"/>  
 </target>  
  
 *<!-- Convert native2ascii for localization files -->* <target name="native2ascii">  
 <native2ascii  
 src="${locale.src.dir}"  
 dest="${locale.build.dir}"  
 encoding="UTF-8"  
 includes="\*\*/\*.properties"/>  
 </target>  
  
  
 *<!-- Find the last working revision if compilation fails -->* <target name="history">  
 <exec executable="git" outputproperty="current.revision" failonerror="false">  
 <arg value="rev-parse"/>  
 <arg value="HEAD"/>  
 </exec>  
  
 <property name="previous.revision" value="HEAD"/>  
 <property name="found.working.revision" value="false"/>  
  
 <taskdef resource="net/sf/antcontrib/antcontrib.properties">  
 <classpath>  
 <pathelement location="${lib.dir}/ant-contrib.jar"/>  
 </classpath>  
 </taskdef>  
 <foreach list="${previous.revision}" param="revision" target="find-working-revision">  
 <param name="revision" value="${previous.revision}"/>  
 </foreach>  
 </target>  
  
 <target name="find-working-revision">  
 <exec executable="git" failonerror="false">  
 <arg value="checkout"/>  
 <arg value="${revision}"/>  
 </exec>  
 <trycatch>  
 <try>  
 <antcall target="compile"/>  
 <echo>Found working revision: ${revision}</echo>  
 <exec executable="git" output="${build.dir}/diff.txt" failonerror="false">  
 <arg value="diff"/>  
 <arg value="${revision}~1"/>  
 <arg value="${current.revision}"/>  
 </exec>  
 <exec executable="git" failonerror="false">  
 <arg value="checkout"/>  
 <arg value="${current.revision}"/>  
 </exec>  
 <fail message="Compilation fixed. Please check diff.txt for changes."/>  
 </try>  
 <catch>  
 <echo>Compilation failed on revision: ${revision}. Checking previous revision...</echo>  
 </catch>  
 </trycatch>  
 </target>  
</project>

Вывод

В результате выполнения лабораторной работы я научился писать сценарии сборки с помощью утилиты Apache Ant. Также я научился писать модульные тесы с помощью библиотеки Junit.