Министерство образования и науки РФ

*Федеральное государственное автономное*

*образовательное учреждение высшего образования*

*«Санкт-Петербургский национальный исследовательский университет*

*информационных технологий, механики и оптики»*

**Факультет ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ**

**ЛАБОРАТОРНАЯ РАБОТА № 3**

по дисциплине

«Основы Программной Инженерии»

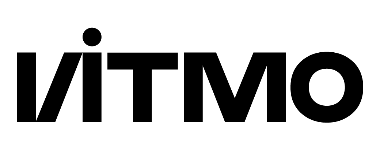
Вариант №-6141

*Выполнили:*

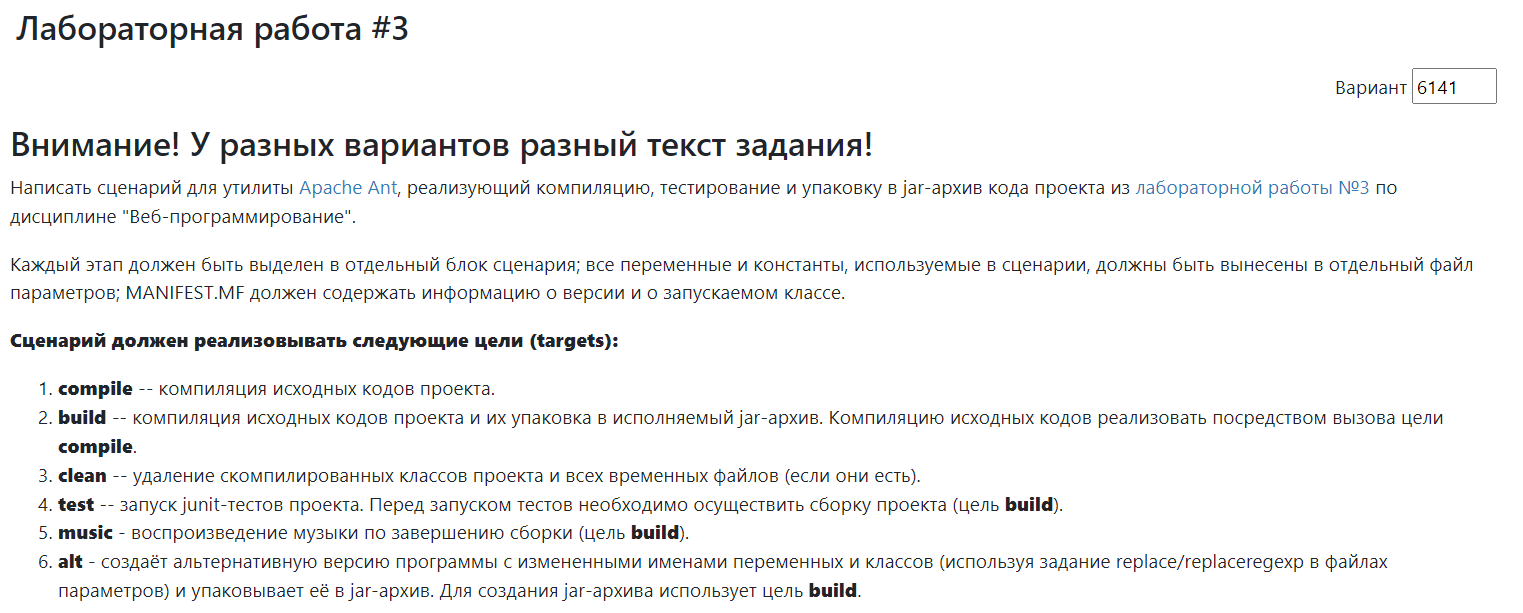
Студенты группы P3233

Берман Денис Константинович

Анисимов Максим Дмитриевич



Санкт-Петербург, 2024

Задание:

Код программы:

Build.xml

<?xml version="1.0"?>  
<project name="lab3">  
 <property file="build.properties"/>  
 <path id="classpath">  
  
 <fileset dir="${lib.dir}" includes="\*.jar"/>  
 <fileset dir="${main.dir}" includes="\*.java"/>  
 </path>  
  
 <path id="classpath.test">  
 <fileset dir="${lib.dir}" includes="\*.jar"/>  
 <fileset dir="src/main/java/test" includes="\*\*/\*Test\*.java"/>  
 <pathelement location="${compiled.classes}"/>  
 <pathelement location="${compiled.tests}"/>  
 </path>  
  
 <target name="compile" depends="clean">  
 <echo>Directory "target" has been initialized with classes.</echo>  
 <mkdir dir="${compiled.classes}"/>  
 <mkdir dir="${compiled.tests}"/>  
 <javac srcdir="${main}" destdir="${compiled.classes}" includeantruntime="false" classpathref="classpath"/>  
 <javac srcdir="${test}" destdir="${compiled.tests}" includeantruntime="false" classpathref="classpath.test"/>  
 </target>  
  
 <target name="build" depends="compile">  
 <echo>Build started.</echo>  
 <jar destfile="${jar}" basedir="${compiled.classes}">  
 <manifest>  
 <attribute name="Built-By" value="${creators}"/>  
 <attribute name="Implementation-Version" value="${version}"/>  
 </manifest>  
 </jar>  
 </target>  
  
 <target name="clean">  
 <echo>Directory "target" has been cleaned.</echo>  
 <delete dir="${compiled}"/>  
 </target>  
 <target name="test" depends="build">  
 <echo>Tests will be tested!</echo>  
 <mkdir dir="${report.test}"/>  
 <junit fork="true" printsummary="on" failureproperty="checker">  
 <classpath>  
 <path refid="classpath.test"/>  
 </classpath>  
 <batchtest todir="${report.test}">  
 <formatter type="xml"/>  
 <fileset dir="${compiled.tests}" includes="\*\*/\*Test\*.class"/>  
 </batchtest>  
 </junit>  
 </target>  
 <target name = "music">  
 <echo>Music will be played!</echo>  
 <sound>  
 <success source="${success}"/>  
 <fail source="${fail}"/>  
 </sound>  
 <antcall target="build"/>  
 </target>  
 <target name="alt">  
 <echo>Crafting an alternative version of program</echo>  
 <copy todir="${src\_old}">  
 <fileset dir="${src}"/>  
 </copy>  
 <replace dir="${src}" replacefilterfile="${altProp}"/>  
 <move file="${changeClass}" tofile="${newClass}"/>  
 <move file="${changeTest}" tofile="${newTest}"/>  
 <antcall target="build"/>  
 <delete dir="${src}"/>  
 <copy todir="${src}">  
 <fileset dir="${src\_old}"/>  
 </copy>  
 <delete dir="${src\_old}"/>  
 </target>  
</project>

AntClasses.properties

lib.dir=lib  
#junit.jar=lib/javaee-web-api-7.0.jar  
#hamcrest.jar=lib/hamcrest-core-1.3.jar  
  
compiled=target  
compiled.classes=target/classes  
compiled.tests=target/classes  
report.test=main\_report/trunk  
report.file=TEST-HitTest.xml  
  
main=src/main  
main.java=src/main/java  
main.dir=${main}/webapp  
test=src/main/java  
jar=target/lab3.jar  
success=sound/success.wav  
fail=sound/fail.wav  
src=./src  
src\_old=./src\_old  
altProp=./alt.properties  
changeClass=${main}/java/DotsBean.java  
newClass=${main}/java/AreaController.java  
changeTest=${main}/java/HitTest.java  
newTest=${main}/java/AreaControllerTest.java  
  
docks=target/doc  
  
version=1.0

Alt.properties

DotsBean=AreaController  
loadAllDotsFromDatabase=loadDB  
HitTest=AreaControllerTest

Hittest.java

import org.junit.Test;  
import org.junit.jupiter.api.Assertions;  
public class HitTest {

@Test  
public void controllerNotHit() {  
Dot point = new Dot();  
point.setX(1000.0);  
point.setY(0.0);  
point.setR(2.0);  
point.controllerHit();  
System.*out*.println("Not Hit Test: " + point.isHit());  
Assertions.assertFalse(point.isHit());  
}

@Test  
public void controllerHit() {  
Dot point = new Dot();  
point.setX(0.5);  
point.setY(-0.5);  
point.setR(1.0);  
point.controllerHit();  
System.*out*.println("Hit Test: " + point.isHit());  
Assertions.assertTrue(point.isHit());  
}  
  
}

Выводы:

В ходе выполнения лабораторной работы мы научились работать с системой сборки Apache Ant, писать и запускать тесты при помощи Ant, компилировать и собирать проект