**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»**

*Факультет інформатики та обчислювальної техніки*

*Кафедра обчислювальної техніки*

**Лабораторна робота №1**

*з дисципліни «Інженерія програмного забезпечення»*

*Виконав студент групи ІО-44*

***Барабаш Т.А.***

*Перевірив:*

***Антонюк А.I.***

**Київ – 2015**

**ЛАБОРАТОРНА РОБОТА №1**

*Підготовка програмного проекту*

**Мета:** Отримання базових навичок з використання мови XML. Вивчення структури типового програмного проекту, форматів стандартних файлів опису проекту. Вивчення формату JAR. Здобуття навичок з використання засобів автоматизації процесу збірки програмних проектів на мові Java - Apache ANT (Another Neat Tool). Розробка програмного проекту на основі типового прикладу.

**I. Завдання**

1. Вивчити синтаксис та структуру мови XML. Вільно володіти поняттями тег, елемент, атрибут. Вміти редагувати XML-файли у текстовому редакторі.
2. Ознайомитись з призначенням і структурою архівів JAR. Вміти формувати архіви JAR та запускати на виконання Java-класи з архіву JAR за допомогою засобів командної строки. Знати призначення підпису архіву JAR.
3. Ознайомитись з засобом автоматизації збірки проектів ANT. Вивчити призначення і структуру файлу build.xml, його структурні компоненти - цілі, завдання, залежності, тощо.
4. З сайту лабораторії завантажити архів типового проекту для середовища Eclipse (template.zip). Ознайомитися з структурою каталогів типового проекту, XML-файлами опису проекту (.project, .classpath, build.xml).
5. Імпортувати типовий проект до робочого простору (workspace) середовища Eclipse. В пакеті com.lablll запустити на виконання (run) клас TestMain. Ознайомитися з засобами використання ANT у середовищі Eclipse - редагування файлу build.xml, виконання цілей у відображенні (View) ANT. Виконати цілі ANT з середовища Eclipse та з командного рядка.
6. Модифікувати типовий проект для його використання у наступних лабораторних роботах. Обов’язково замінити назву і автора проекту в файлах опису проекту (.project, build.xml). Модифікувати файл build.xml таким чином, щоб у ньому були всі потрібні цілі, вільно володіти призначенням кожної цілі.
7. Розробити та перевірити в eclipse нову ціль в файлі build.xml згідно варіанту.
8. **Визначення варіанту завдання:**

**Номер залікової книжки:** 4403;

**Варіант:** 4403 *mod* 9 = **2** - *Створити в каталозі out jar-архів з усіх файлів проекту з розширеннями java, js, html, htm. Скопіювати архів в корінь проекту.*

**IІ. Код програми**

**1. build.xml**

<project name="Software Engineering" default="make-jar" basedir=".">

<!-- ================== Application Properties =================== -->

<property name="app.name" value="Software Engineering" />

<property name="app.version" value="1.0" />

<property name="app.title" value="Template Application" />

<property name="app.author" value="Taras Barabash" />

<property name="app.company" value="KPI-FICT-IO-44" />

<property name="sourceDir" value="src/main/java" />

<property name="outputDir" value="out/eclipse-classes" />

<property name="buildDir" value="out/build" />

<property name="resourceDir" value="src/res" />

<property name="libDir" value="lib" />

<!-- ================== JAR Properties =================== -->

<property name="jar.mainClass" value="com.tb.testClass" />

<property name="jar.name" value="template.jar" />

<property name="jar.keyStore" value="${basedir}/out/tempKey.store" />

<property name="jar.keyPass" value="telpat" />

<property name="jar.keyAlias" value="tempAlias" />

<!-- ==================== Compilation Control Options ==================== -->

<property name="compile.debug" value="true"/>

<property name="compile.deprecation" value="false"/>

<property name="compile.optimize" value="true"/>

<path id="compile.classpath">

<fileset dir="${libDir}">

<include name="\*.jar"/>

</fileset>

</path>

<!-- ==================== Clean Target ================================== -->

<target name="clean"

description="Clean build dirs">

<delete dir="${buildDir}" />

</target>

<!-- ==================== Prepare Target ================================== -->

<target name="prepare"

depends="clean"

description="Prepare build dirs">

<mkdir dir="${buildDir}"/>

<mkdir dir="${buildDir}/output"/>

<mkdir dir="${buildDir}/web-apps"/>

</target>

<!-- ==================== Compile Target ================================== -->

<target name="compile"

depends="prepare"

description="Compile Java sources">

<javac srcdir="${sourceDir}"

destdir="${buildDir}/output"

debug="${compile.debug}"

deprecation="${compile.deprecation}"

optimize="${compile.optimize}"

target="1.8"

source="1.8"

includeantruntime="false">

<classpath refid="compile.classpath"/>

</javac>

<copy todir="${buildDir}/output">

<fileset dir="${sourceDir}" excludes="\*\*/\*.java"/>

</copy>

</target>

<!-- ==================== Create JAR archive Target ================================== -->

<target name="createJAR"

depends="compile"

description="Create JAR archive" >

<jar destfile="${buildDir}/${jar.name}" basedir="${outputDir}">

<manifest>

<attribute name="Created-By" value="${app.author} - (${app.company})"/>

<attribute name="Built-By" value="${user.name}"/>

<attribute name="Main-Class" value="${jar.mainClass}"/>

<section name="${app.name}">

<attribute name="Specification-Title" value="${app.title}"/>

<attribute name="Specification-Version" value="${app.version}"/>

<attribute name="Specification-Vendor" value="${app.company}"/>

<attribute name="Implementation-Title" value="${app.name}"/>

<attribute name="Implementation-Version" value="${app.version}"/>

<attribute name="Implementation-Vendor" value="${app.company}"/>

</section>

</manifest>

</jar>

</target>

<!-- ==================== Generate Key for JAR signing Target ================================== -->

<target name="generateKey"

description="Generates Key for JAR signing">

<delete failonerror="false" file="${jar.keyStore}"/>

<genkey keystore="${jar.keyStore}" alias="${jar.keyAlias}" storepass="${jar.keyPass}" validity="720" keyalg="RSA">

<dname>

<param name="CN" value="${app.company}"/>

<param name="OU" value="${app.title}"/>

<param name="O" value="${app.company}"/>

</dname>

</genkey>

</target>

<!-- ==================== FixStyle Target ================================== -->

<target name="fixstyle"

description="Fix Style in source code" >

<fixcrlf srcdir="${basedir}"

tab="remove"

tablength="2"

includesfile="fixstyle.list"

/>

</target>

<!-- ==================== JAR Singing Target ================================== -->

<target name="signJAR"

depends="createJAR,generateKey"

description="Signing JAR archive">

<exec dir="${buildDir}" executable="jarsigner">

<arg line="-keystore ${jar.keyStore} -storepass ${jar.keyPass} ${jar.name} ${jar.keyAlias}"/>

</exec>

<delete file="${buildDir}/myKeystore" />

</target>

<!-- ==================== Make zip Target ================================== -->

<target name="make-zip-project"

description="Zip all project tree from basedir">

<tstamp />

<zip destfile="../${app.name}-${DSTAMP}-${TSTAMP}.zip"

basedir="${basedir}"

excludes="out/\*\*"/>

</target>

<!-- ==================== Lab1 Var2 Target ================================== -->

<target name="make-jar"

depends="compile"

description="Makes a JAR archive of .java,.js,.html,.htm files + copies it to root">

<jar destfile = "out/${app.name}-${app.author}-${app.version}.jar">

<fileset dir="">

<include name="\*\*/\*.java"/>

<include name="\*\*/\*.js"/>

<include name="\*\*/\*.html"/>

<include name="\*\*/\*.htm"/>

</fileset>

</jar>

<copy file="out/${app.name}-${app.author}-${app.version}.jar" todir=""/>

</target>

</project>

***Цей файл містить у собі набір цілей та опис способів їх досягнення.***

**2. .classpath**

<?xml version="1.0" encoding="UTF-8"?>

<classpath>

<classpathentry kind="src" path="src/main/java"/>

<classpathentry kind="src" path="src/test/java"/>

<classpathentry kind="con" path="org.eclipse.jdt.launching.JRE\_CONTAINER"/>

<classpathentry kind="output" path="out/eclipse-classes"/>

</classpath>

***Сlasspath – це з`єднуюча ланка між виконуючим модулем Java та файловою системою.***

**3. .project**

<?xml version="1.0" encoding="UTF-8"?>

<projectDescription>

<name>template</name>

<comment></comment>

<projects>

</projects>

<buildSpec>

<buildCommand>

<name>org.eclipse.jdt.core.javabuilder</name>

<arguments>

</arguments>

</buildCommand>

<buildCommand>

<name>com.atlassw.tools.eclipse.checkstyle.CheckstyleBuilder</name>

<arguments>

</arguments>

</buildCommand>

</buildSpec>

<natures>

<nature>org.eclipse.jdt.core.javanature</nature>

<nature>com.atlassw.tools.eclipse.checkstyle.CheckstyleNature</nature>

</natures>

</projectDescription>

***Цей файл містить детальну інформацію про проект середовища Eclipse.***

**ІIІ. Результати роботи програми**

Buildfile: C:\Workspace\template\build.xml

clean:

[delete] Deleting directory C:\Workspace\template\out\build

prepare:

[mkdir] Created dir: C:\Workspace\template\out\build

[mkdir] Created dir: C:\Workspace\template\out\build\output

[mkdir] Created dir: C:\Workspace\template\out\build\web-apps

compile:

[javac] Compiling 1 source file to C:\Workspace\template\out\build\output

make-jar:

[jar] Building jar: C:\Workspace\template\out\Software\_Engineering-Taras\_Barabash-1.0.jar

[copy] Copying 1 file to C:\Workspace\template

BUILD SUCCESSFUL

Total time: 688 milliseconds

**ІV. Висновок**

У ході лабораторної роботи було отримано базові навички з використання **ХМL**, а саме: вивчено синтаксис та структуру. Також вивчено призначення і структуру архівів **JAR** та отримані навички роботи із засобом автоматизації збірки проектів **ANT**.