

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
КАФЕДРА АВТОМАТИЗОВАНИХ СИСТЕМ ОБРОБОТКИ ІНФОРМАЦІЇ І УПРАВЛІННЯ

Лабораторна робота №1
з дисципліни «Інженерія програмного забезпечення»
на тему: *«Підготовка програмного проекту»*

Виконав:
студент гр.ІП-41
Руських Є. М.

Перевірив:
Подрубайло О.

Київ 2014

Лабораторна робота №1

Тема: 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.

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.

1. *Варіант завдання 4*

Номер залікової книжки 4126
Варіант 4126 mod 9 = 4

Дія :

Створити каталог build2. Виконати компіляцію сирцевих файлів проекту в створений каталог.

2. *Програмний код*

зміст каталогу проекту

Зміст каталогу проекту:

template/	// Кореневий каталог проекту
template/.checkstyle	// файл перевірки стилю
template/.classpath	// файл шляхів до відповідних директорій
template/.project	// файл опису проекту
template/build.xml	// файл сценарію побудови проекту
template/doc/	// каталог документації
template/lib/	// каталог бібліотек
template/out/	// каталог відкомпільованих файлів
template/out/build/	// каталог вихідного jar архіву
template/out/build/output/	
template/out/build/output/com/	
template/out/build/output/com/lab111/	// каталог вихідних класів
template/out/build/output/com/lab111/TestMain.class	// вихідний клас
template/out/build/template.jar	// вихідний архів
template/out/build/web-apps/	
template/out/eclipse-classes/	
template/out/eclipse-classes/com/	
template/out/eclipse-classes/com/lab111/	
template/out/eclipse-classes/com/lab111/TestMain.class	
template/out/tempKey.store	
template/prop	
template/readme.txt	
template/src/	
template/src/main/	
template/src/main/java/	
template/src/main/java/com/	
template/src/main/java/com/lab111/	
template/src/main/java/com/lab111/TestMain.class	
template/src/main/java/com/lab111/TestMain.java	
template/src/main/javascript/	
template/src/main/php/	
template/src/main/sql/	
template/src/res/	
template/src/test/	
template/src/test/java/	
template/src/test/javascript/	
template/src/test/php/	
template/src/test/sql/	
template/src/web/	
template/tools/	

build.xml

```

<project name="Template1" default="signJAR" basedir=". ">
<!-- ===== Application Properties ===== -->

<property name="app.name" value="template1" />

<property name="app.version" value="0.1" />

```

```

<property name="app.title" value="Template Application" />
<property name="app.author" value="Liza Russkih" />
<property name="app.company" value="Lab111 - http://www.comsys.ntu-kpi.kiev.ua/lab111" />
<property name="sourceDir" value="src/main/java" />
<property name="outputDir" value="out/eclipse-classes" />
<property name="buildDir" value="out/build" />
<property name="buildDir2" value="build2" />
<property name="resourceDir" value="src/res" />
<property name="libDir" value="lib" />

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

<property name="jar.mainClass" value="com.lab111.TestMain" />
<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}" />

    <delete dir="${buildDir2}" />

</target>

<!-- ===== Lab1 Clean Target
===== -->

<target name="lab1 clean"
    description="Clean build dirs">

    <delete dir="${buildDir2}" />

</target>

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

<target name="prepare"
    depends="clean"
    description="Prepare build dirs">

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

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

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

    <mkdir dir="${buildDir2}" />

        <mkdir dir="${buildDir2}/output" />

        <mkdir dir="${buildDir2}/web-apps" />

</target>

<!-- ===== Lab1 Prepare Target
===== -->

<target name="lab1 prepare"
    depends="lab1 clean"
    description="Prepare build dirs">

    <mkdir dir="${buildDir2}" />

</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="${compile.optimize}"
        target="1.5"
        source="1.5">
    <classpath refid="compile.classpath"/>
</javac>
<copy todir="${buildDir}/output">
    <fileset dir="${sourceDir}" excludes="**/*.java"/>
</copy>
</target>

<!-- ===== Lab1 Compil Target
===== -->

```

```

<target name="lab1 compil"
        depends="lab1 prepare"
        description="Compile Java sources">
<javac srcdir="${sourceDir}"
        destdir="${buildDir2}"
        debug="${compile.debug}"
deprecation="${compile.deprecation}"
        optimize="${compile.optimize}"
        target="1.5"
        source="1.5">
    <classpath refid="compile.classpath"/>
</javac>
</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">
        <cname>
            <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 Signing 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>
</project>

```

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

Змінено ім'я автору проекту та його назву.

pom.xml

```

    <?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.lab111</groupId>
    <packaging>jar</packaging>
    <artifactId>myTemplate</artifactId>
    <version>1</version>
    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-antrun-plugin</artifactId>
                <version>1.5</version>
                <configuration>
                    <tasks>
                        <ant antfile="build.xml" target="lab1 compil"/>
                    </tasks>
                </configuration>
            </plugin>
        </plugins>
    </build>
</project>

```

```

    </plugin>

</plugins>

<!-- maven ant plugin-->

</build>

</project>

```

.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>

```

Classpath являє собою з'єднуючу ланку між виконуємим модулем Java та файловою системою. Він описує, де компілятор та інтерпритатор шукають пакет файлів .class для завантаження.

.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>
        <buildCommand>
            <name>org.eclipse.ui.externaltools.ExternalToolBuilder</name>
            <arguments>
                <dictionary>
                    <key>LaunchConfigHandle</key>

```

```

        <value>&lt;project&gt;/.externalToolBuilders/Ant
builder.launch</value>
        </dictionary>
    </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.

Опис цілей проекту

Clean Target – ціль для видалення каталогу зборки.

Compile Target – ціль для компіляції проекту.

Create JAR archive Target – ціль для створення JAR архіву.

Generate Key for JAR signing Target – ціль для генерації ключів для архівування.

Prepare Target – ціль для створювання каталогів.

FixStyle Target – ціль для фіксування стилю у сирцевому коді.

JAR Singing Target – ціль для підписування JAR архіву.

Lab1 Clean Target, Lab1 Prepare Target, Lab1 Compile Target – власні цілі, створені відповідно до завдання.

3. Висновок

У ході лабораторної роботи я ознайомився з призначенням і структурою архівів JAR, засобом автоматизації збірки проектів ANT, вивчив синтаксис та структуру мови XML.