

Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут»  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки

**Лабораторна робота №1**  
**Підготовка програмного проекту**

Підготував  
студент I курсу  
групи ІО-32  
Довгаль Д.С.  
Варіант 3211

## 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 файла-сценарія build.xml, його структурні компоненти – цілі, завдання, залежності, тощо.
4. 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.lab111 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 командної строки.
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. Видалити з корню усі .jar файли, упакувати увесь проект у JAR окрім папки OUT.

### Роздруківка вмісту каталогу

/src/main/java — каталог містить вихідні файли з кодом (файли з розширенням .java), придатні для компіляції

/out/eclipse-classes — каталог, який містить в собі відкомпільовані файли (з розширенням .class)

/out/build — каталог, містить в собі зібраний проект (відповідно до існуючого файлу build.xml, Ant кладе сюди відкомпільовані файли а також .jar файл)

## Роздруківка файлу build.xml

```
<project name="Template" default="compile" basedir=". ">

<!-- ===== Application Properties ===== -->
<property name="app.name" value="Sample" />
<property name="app.version" value="1.0" />
<property name="app.title" value="Sample Title" />
<property name="app.author" value="Dyadya Fedor" />
<property name="app.company" value="Prostokvashino DevCo" />
<property name="deljar" value="*.jar" />

<property name="sourceDir" value="src/main/java" />
<property name="outputDir" value="out/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.ipz.Main" />
<property name="jar.name" value="MegaJar.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="1.5"
    source="1.5">
    <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">
        <dnname>
            <param name="CN" value="${app.company}" />
            <param name="OU" value="${app.title}" />
            <param name="O" value="${app.company}" />
        </dnname>
    </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>

    <!-- ===== Make Lab 1 task ===== -->
    <target name="dollarlab" description="Do first lab of IPZ">
        <delete>
            <fileset dir="${basedir}" includes="**/*.jar"/>
        </delete>

        <jar destfile="build/All_in_One.jar">
            <fileset dir="${basedir}">
                <exclude name="**/out"/>
            </fileset>
        </jar>
    </target>
</project>
```