17-02-22

Practical No. 1 and 2: Setup a Java programming development environment and test using small programs

III Competency and Practical Skills

"Develop Application using Java."

The practical expected to develop the following skills:-

1. Set up Java Environment for executing Java programs.
2. Execute simple programs by setting path variable.

Setup a java programming development environment.

1. Using Command Prompt
2. Using IDEs.

IV Relevant course Outcome(s)

Develop programs using Object Oriented methodology in Java.

V Practical Outcome

Setup a Java programming development environment.

IX Resources used

S.No	Name of resource	Broad Specification
1.	Computer System with broad specs	Dell laptop, i5-4th gen 8 GB RAM
2.	Software	jdK 17.0.2

X Practical related Questions:-

1. Write installation directory path of your directory?

→ c:\program files\java\jdK 17.0.2\bin

2. Write value of path environment variable

→ c:\program files\java\jdK 17.0.2\bin

3. List folders created after installation

→ The folders created after ~~creating~~ installation are bin, conf, include, jmods, legal, lib.

4. Main method is declared as static. Justify.

→ It is declared as static so only one copy of main method is created. And we dont need any object to call the main method.

PAGE No. / DATE / /

5. Program is named with class containing main method. Justify.

→ It tells interpreter which class to load so it will load the '.class' file.

VI Exercise

Write the options provided by following JDK tools along with their use.

1. Java

→ -cp <class search path of directories and zip/jar files>

-classpath <class search path of directories and zip/jar files>

--class-path <class search path of directories and zip/jar files> A; separated list of directories, JAR Archives, and ZIP archives.

-p <module path>

--module-path <module path>... A; separated list of directories, each directory is a directory of module.

--upgrade-module-path <module path>...

A; separated list of directives, each directory is a directory of modules that replace upgradable modules in the runtime image.

--add-modules <module name> [,<module name>...]

root modules to resolve in addition to the

initial module. <module name> can also be ALL-DEFAULT, ALL-SYSTEM, ALL-MODULE-PATH.

--enable-native-access <module name>[,<module name>...]
modules are permitted to perform restricted native operations. <module name> can also be ALL-UNNAMED.

--list modules : list observable modules and exit
-d <module name>, --describe-module <module name>
describe a module and exit
--dry-run :- Create VM and load main class but do not execute main method.
~~--validation~~
-validate-modules :- Validate all modules and exit.
It can be useful for finding conflicts and other errors with modules on the module path.
-D<name>=<value> :- Set a system property
-verbose:[class|module|gc|jni]
enable verbose output for given subsystem
-version :- Print product version to the error system and exit.
--version :- Print product version to the output stream and exit.
-showversion :- Print product version to the output stream and continue
--show-module-resolution :- Show module resolution

output during setup.

-? -h -help :- Print this help message to the error output stream.

--help :- Print this help message to the output stream.

-X :- Print help on extra options to the error stream.

-help-extra :- Print help on extra options to the output stream.

-ea[:<packagename>...|:<classname>]

-enableassertions[:<packgename>...|:<classname>]

disable assertions with specified granularity.

-esa | -enablesystemassertions :- enable system assertions

-dsa | -disablesystemassertions

disable system assertions.

-agentlib:<libname>[=<options>] :- Load native agent library <libname>, eg. -agentlib:jdwp

-agentpath:<pathname>[=options]

load native agent library by full pathname.

-javaagent:<jarpath>[=<options>]

load Java programming language agent.

-splash:<image path>

show splash screen with specified image.

PAGE No.
DATE / /

@argument files :- one or more argument files containing options.

-disable-@files :- prevent further argument file expansion.

--enable-preview :- allow classes to depend on preview factory of this release to specify as argument for long instruction you can use --<name>=<value> or --<name><value>.

2. javac

→ Options are :-

@filename> :- read options and filenames from files.

-Akey[=value] :- Options to pass to annotation processors.

--add-modules <module>(,<module>)*

Root modules to resolve in addition to the initial modules, or all modules on the module path if <module> is ALL-MODULE-PATH.

--boot-class-path <path>, -bootclasspath <path>

Override location of bootstrap class files.

--class-path <path>, -classpath <path>, -cp <path>

Specify where to find user class files and annotation processors.

-d <directory> :- Specify where to find place generated class files.

PAGE No.
DATE / /

-deprecation :- Output source locations where deprecated API's are used.
--enable-preview :- Enable preview language features. To be used in conjunction with either -source or --release.
-encoding <encoding> :- Specify character encoding used by source files.
-endorseddirs <dirs> :- Override location of standard path
-extdirs <dirs> :- Override location of installed extensions.
-g :- Generate all debugging info.
-g:{lines, vars, source} :- Generate only some debugging info.
-g:none :- Generate no debugging info.
-h <directory> :- Specify where to place generated native header files.
--help, -help, -? :- Print this help message.
--help-extra, -x :- Print help on extra options.
-implicit:{none, class}
Specify whether or not to generate class files for implicitly referenced files
-J<flag> :- Pass <flag> directly to the runtime system.
--limit-modules <module>(,<module>)*
Limit the universe of observable modules.

--module <module>(,<module>)*, -m <module>(,<module>)*
Compile only specified modules, check tim timestamps.
--module-path <path>, -p <path>
Specify where to find application modules.
--module-source-path <module-source-path>
Specify where to find input source files for multiple modules
--module-version <version>
Specify version of modules that are being compiled.
-nowarn :- Generate no warnings.
-parameters :- Generate metadata for reflection on method parameters.
-proc:{none,only}
Control whether annotation processing and/or compilation is done.
-processor <class1>[,<class2>,<class3>...]
Names of the annotations processors to run; bypasses default discovery process.
--processor-module-path <path>
Specify a module path where to find annotation processors
--processor-path <path>, -processor path <path>
Specify where to find annotation processors.

-profile <profile> :- Check that API used is available in the specified profile.

--release <release>
Compile for the specified Java SE release.

-s <directory> :- Specify where to find place generated source files.

-source <release>, -source <release>
Provide source compatibility with the specified JAVA SE.

--source-path <path>, -sourcepath <path>
Specify where to find input source files.

--system <jdk>/none :- Override location of system modules.

--target <release>, -target <release>
Generate class files suitable for the specified JAVA SE release.

--upgrade-module-path <path>
Override location of upgradeable modules.

-verbose :- Output messages about what the compiler is doing.

--version, -version :- Version info.

-Werror :- Terminate compilation if warnings occur.

3. javadoc

→ @<file> Read options and filenames from file

PAGE No.
DATE / /

--add-modules <module>(,<module>)*
-bootclasspath <path> :- Override location of platform class files used for non-modular releases
-breakiterator :- Compute first sentence with breakiterator.
--class-path <path> ,-cp <path> -classpath <path>
-doclet <class> :- Generate output via doclet.
-doclet path <path> :- Specify where to find doclet files.
--enable-preview
-encoding <name> - Source file encoding name
-exclude <pkglist>
Specify list of package to exclude.
-expand-requires <value>
Instruct tool to expand the set of modules to be documented. By default, only the modules given explicitly on command line will be documented. A value of "transitive" will additionally include all "requires transitive" dependencies of those modules. A value of "all" will include all dependencies of those modules.
-extdirs <dirlist>
Override location of installed extensions.
--help, -help, -?, -h
--help-extra, -x

-J<flags>
--limit-modules <module>(,<module>)*
-locale <name> :- Locale to be used, eg. en-US or en-US-WIN.
--module <module>(,<module>)*
Document the specified modules.
--module-path <path>, -p <path>
specify where to find application modules.
--module-source-path <path>
Specify where to find input source files for multiple modules
-package
Show package/protected/public types and members. For named modules, show all packages and all module details.
-private :- Show private types and members. For named modules, packages and all module details.
-protected :- Show protected/public types and members (default). For named modules, show exported packages and the modules API.
-public :- Show only public types and members. For named modules, show exported packages and the Modules API.
-quiet :- Do not show status message.
--release <release> :- Provide source compatibility

PAGE No.
DATE / /

--show-members <value>

Specifies which members (fields, methods, etc.) will be documented where value can be one of "public", "protected", "package" or "private". The default is "protected" which will show public and protected members, "public" will show only public members, "package" will show public, protected and package members and "private" will show all members.

--show-module-contents <value>

Show Specify the documentation granularity of module declaration.

--show-packages <value>

Specifies which of module package will be documented.

--show-types <values>

Specifies which types (classes, interfaces, etc.) will be document. Same as show members.

--source <release>, -source <release>

--source-path <path>

Specify where to find source files.

-subpackages <subpkglist>

Specify subpackages to recursively load.

--system <jdk>

--upgrade-module-path <path>

PAGE No.
DATE / /

-verbose :- Output message of what javadoc is doing

--version :- Print version info.

-Werror :- Report an error if any warnings occur.

Provided by standard doclet:-

--add-stylesheet <file>

Additional stylesheet file for the generated documentation.

--allow-script-in-comments

-author :- Include @author paragraphs.

-bottom <html-code> :- Include bottom text for each page

-charset <charset> :- Charset of cross platform viewing of generated documentation.

-d <directory> :- ~~Destruction~~ Destination directory for output files.

-docencoding <name> :- Specify the character encoding for the output.

-docfilessubdirs :- Recursively copy doc-file subdirectories

-doctitle <html-code> :- Include title for the overview page.

-excludedocfilessubdir <name>:...

Exclude any doc-files subdirectories with given name.

PAGE No.
DATE / /

-footer <html-code> :- Include footer for each page.

-group <name> <g1> : <g2> ...
Group specified elements in overview page.

-header <html-code>

-helpfile <file> :- Include file that help link links to

-html5 :- Generate html5 output.

--javafx, -javafx :- Enable javafx function.

-keywords :- Include HTML meta tags with package class and member info.

-link <url> :- Create links to javadoc output.

-linkoffline <url1> <url2> :- Link to docs at <url1> using package list at <url2>.

-linksource :- Generate source in HTML.

--main-stylesheet <file>, -stylesheetfile <file>
File to change style of the generated documentation.

-nocomment :- Suppress description and tags, generate only declarations.

-nodeprecated :- Do not include @deprecated info.

-nodeprecatedlist :- Do not generate deprecated list.

-nohelp :- Do not generate help link.

-noindex :- Do not generate index

-nonavbar :- Do not generate navigation bar.
-no-platform-links :- Do not generate platform documentation.
-noqualifier <name1>:<name2>:...
Exclude list of qualifiers.
-nosince :- Do not include @since info.
-notimestamp :- Do not include hidden timestamp.
-notree :- Do not generate class hierarchy.
--override-methods (detail|summary)
Document overridden methods in the detail or summary sections.
-overview <file>
-serialwarn :- Generate warning about @serial tags.
--since <release>(,<release>)*
Document new and deperated API in the specified releases.
--since-label <text> :- Provide text to use in the heading of the "New API" page
-sourcetab <tab length> :- Specify the number of spaces each tab takes up in the source
-splitindex :- Split index into 1 file per letter.
-tag <name>:<locations>:<header>
Specify argument custom tags.
-taglet :- The fully qualified name of the taglet to register.

-taglet path

-top <html-code> :- Include top text for each page.

-use :- Create class and package usage pages

-version :- Include @version paragraphs

-windowtitle <text> :- Browser window title for the documentation.

2. List different versions of JDK.

→ JDK 1.0, JDK 1.1, J2SE 1.2, 1.3, 1.4, 5.0, Java SE 7, 8.

3. Test the setup using similar programs.

→
```
class abc
{
public static void main(String args[])
  {
    System.out.println("Hello Java World!");
  }
}
```