Working with IDE and Eclipse

====================================================

1) What is an IDE, Why IDE and Adv of IDE?

2) What are the popular IDEs?

3) What is eclipse?

4) Why eclipse?

5) How to install and start eclipse?

7) What is a workspace, what is a perspective, what is a project?

8) Diff implicit configuration folders and files created by eclipse and their purpose?

9) Diff types of projects we can create in eclipse?

10) Downloading JDK 18 plugin from eclipse market place?

11) Developing a Java project with class and interface?

12) Compiling and Running a class using eclipse?

13) Running a class by reading runtime values as command line arguments and from Scanner?

14) Diff short cuts to work with eclipse?

15) Debugging project in eclipse?

16) How to Import external project into Eclipse workspace?

17)How to push Java project to Git Hub Repositary?

==============================================================

**1) What is an IDE?**

- IDE stands for Integrated Development Environment

- The IDE is an editor software that provides an evironment to develop projects

by integrating to different softwares which are used as part of this project

- That means IDE provides connectivity to different sotwares from the same editor window.

- We no need to open each software in separate window those we are using in our project

- IDE provides fast development

**2) What are the advantages of IDEs?**

1) it gives code help (less typing, more choosing)

2) automatic compilation of code

3) automatic importing packages

4) easy to copy paste and moving lines

5) auto generation of code (fields, constructors, methods, condition, loops, try, catch, finally, throws, etc...)

6) easy debugging

7) we can connect to diff sotwares from this IDE editor

8) fast development

9) time and money is saved.

**3) Popular IDEs for Java application development?**

1) Eclipse IDE

2) STS

3) IntelliJ IDEA

**4) Working with Eclipse?**

1) What is eclipse?

- An eclipse is an IDE software

- It is an text editor software

- It is an open source software (freely avilable to download, we can get its source code and even we can also involve in its features development )

- It is meant for developing projects more faster with less typing

2) Why Eclipse?

- for fast develop with ease and less efforts

3) How to install eclipse?

1) download eclipse from "eclipse.org/downloads"

2) click on "Download Packages" (don't click download button directly)

3) click "Eclipse IDE for Enterprise Java Developers"

4) click "Windows x86\_64"

5) you will find "eclipse.zip" file is downloaded into your system

6) extract it to "C:\" drive by using winrar software

7) eclipse installation is completed.

4) How to start eclipse?

1) go to eclipse installed folder "C:\eclipse"

2) double click "eclipse.exe" file

3) select workspace directory\*

4) click "Launch" button

5) eclipse is started and welcome page is opened

Q) What is a workspace and what is the use of it?

- A Workspace is a folder where you want to store projects.

- It is used by eclipse for storing all plugings, and setups we are doing for this workspace

- When we launch eclipse next time it stats eclipse by loading the setups we saved in this workspace

- To identify a folder as workspace, in this folder eclipse creates

a directory with the name ".metadata".

- In this folder eclipse stores all plugins and the setups we are doing for this workspace.

Note: Do not delete this folder, it is not a virus folder

- If you delete this folder, eclipse can not recognize your folder as workspace

- All setups and projects you created earlier are not loaded into eclipse

5) Creating project

- click "create project"

- click "Java project" -> Next

- Enter project name: Test

- Choose Execution environment: Java SE 18 (if not available add plugin from market place)

- click Next -> Finish -> Donot create

- click Open Java Project perspective

- Adding pulgin from market place

1) Click "Help" menu

2) Click "Eclipse Maket Place"

3) In search box type Java SE 18 -> press enter key

4) It will show you Java SE 18 plug-in -> click "Install" button

5) Down Left side it will show installing progress bar

6) After installation completed, restart eclipse

7) You need add this plug-in only one time per eclipse installation

- Adding JDK 18 to project

- Setting JRE to project

1) Right click on project

2) Click Build path

3) Click Add Libraries

4) Click JRE System Library

5) Select Alternative JRE

5) Click Installed JREs

6) Click Add

7) Click Standard VM

8) Click Directory

9) Select C:\JDK18.0.1.1 folder from your system, click finish

10) Select JDK18.0.1.1 check box, click Apply and Close

JDK 18 is applied to our project

Note: a new project is created with the name "Test"

- verify its structure in the workspace folder

- you will find

1) src folder -> contains .java files those we are creating from eclipse

2) bin folder -> contains .class files those are compiled from .java files

3) .settings folder -> contains the setting we did for this project

4) .classpath file -> contins the jars information those attached to this poject

5) .project file -> contins information above this project to load when eclipse is started

Note: if we delete any of above folders or files your project will not work from eclipse

6) Developing a class in eclipse?

1) right click on src folder in Test project

2) click "class"

3) enter class name: Test

4) click main method chck box

5) click finsh

7) add System.out.println("Hi"); in main method

7) Compiling and running a class from eclipse

1) In eclipse every class is auto compiled and .class is saved in bin folder

2) We no need to compile but we must execute

3) We have 4 options to run java class in eclipse

1) By using "Run button" available in images bar

2) Right click on editor any where -> click "Run As" -> click "Java Application"

3) In package explore view -> Right click on .java file -> click "Run As" -> click "Java Application"

4) Short cut -> press (Ctrl + F11) (or) (Ctrl + fn + F11)

8) Changing font

1) for just increasing and decreasing font size

- press ctrl ++ and ctrt --

2) for changing font name, font style and font size

- click window menu

- click Preferences menu item

- click General -> Appearances -> Colors and Fonts

- double click basic -> scroll down -> select Text font

- click Edit button -> select the font name, font style, font size

- click "Apply and close" button

9) Showing compiler and JVM errors and exceptions

- shows compile time errors with red color mark

- shows exception in console window, click on line number hyper link showing in this exception message

to goto the line number directly in this Java file

10) Running Java program in eclipse with command line arguments

cmd>java Addition 10 20

Editplus -> Tools -> Configure User Tools -> Select JVM -> Argument -> place $(prompt) at end

Eclipse

1) Right click on editor any where

2) Click "Run As"

3) Click "Run Configuration"

4) Click "Arguments" tab

5) Enter arguments with space separator

11) Short cuts to work with eclipse

=====================================

To open short cut keys list

ctrl + shift + L

For Changing Eclipse short curts

Window -> preferences -> keys

=====================================

for code drop down --> ctrl + space

for filtering code -> type some partial text --> press ctrl + space

for generating main method --> type main --> press ctrl + space

for adding System.out.println() --> sysout/sys/syso/sout + (ctrl + space)

for adding System.out.println("trace") --> systrace/sys/syst/strace + (ctrl + space)

for adding System.err.println(); --> syserr/sys/syse/serr + (ctrl + space)

For executing ctrl +F11

For commenting & uncommenting one line ctrl + shift + c -> SLC

For commening & uncommenting multiple lines ctrl + /, / -> SLC

For comming & uncommenting multiple lines ctrl + shift + /, \ -> MLC

For adding doc comment alt + shift + j -> DocC

For indentation for one line ctrl + i,

For indentation for all lines ctrl + A then ctrl + i

For formatting ctrl + shift + f,

For Duplicate line(C and P) ctrl + alt + down arrow/up arrow

-> For disabling hot keys on windows

-> Right click on desktop -> Graphics options -> Hot keys -> Disable

For Moving line down and up alt + down arrow, alt + up arrow

For deleting complete line ctrl + D

For deleting next word ctrl + Delete

For deleting previous word ctrl + Backspace

For converting to upper case ctrl + shift + X

For converting to lower case ctrl + shift + Y

For finding the selected word occurrences ctrl + shift + K

For creating a class or interface or enum or package -> ctrl + N

For quick fix/for creaing class, field, constructor, method -> ctrl + 1

Right click -> Source ->

For generting constructor alt + shift + s, o, enter

For generting getter and setter alt + shift + s, r, alt+a, alt+r

For generting toString alt + shift + s, s, enter

Right click -> Refactor -> Rename (alt + shift + r)

For replacing a PE dec name alt + shift + r

in the entire project(refactoring)

Moving to next Editor cltr + page down

Moving to previous Editor cltr + page up

for chaning editors ctrl + f6 + "hold ctrl" -> keep pressing F6 for selecting editor -> release ctrl

For chaning views ctrl + f7 + "hold ctrl" -> keep pressing F7 for selecting view -> release ctrl

For changing perspecitives ctrl + f8 + "hold ctrl" -> keep pressing F8 for selecting perspective> release ctrl

Note: if you "leave ctrl" key after function key pressed, previously opened editor, view and perspective will be opened

Attach source code to eclipse -> place cursor on class.variable or method -> press F3 -> Click Browse ->

-> select JDK-16\lib\src.zip

For opening source code F3 or ctrl + mouse click

For going to prev and next alt + <- / ->

For opening API Doc F2

open particular type ctrl + shift + t

For adding import statements ctrl + shift + o

for working line number tab ctrl + F10

For running a class ctrl + F11 (if main method is not available in this class,

previously executed class is executed)

For closing editor ctrl + F4

For closing editor ctrl + w

For closing all editors ctrl + shift + w

Summary on function keys:

f1 -> help menu will be opened

ctrl + F1 -> same work [help menu will be opened]

f2 -> for showing PE Dec &API Documentation -> place curson on PE -> press F2

ctrl + F2 -> do nothing

f3 -> for opening source code of a PE -> place curson on PE -> press F3

ctrl + F3 -> for opening PE outline

f4 -> for opening type hierarchy (for seeing its super classes and its members at a time)

ctrl + F4 -> editor is closed

f5 -> for refreshing project (this project modifications are loaded from HD to eclipse)

ctrl + F5 -> do nothing

f6 -> do nothing

ctrl + f6 -> shows all editors

f7 -> do nothing

ctrl + f7 -> shows all views

f8 -> do nothing

ctrl + f8 -> shows all perspectives

f9 -> do nothing

ctrl + f9 -> opens active task window to select a task

f10 -> selecting menus(like pressing alt)

ctrl + f10 -> shows line numbers bar shortcuts

f11 -> Program is executed in debugging mode

ctrl + f11 -> program is executed normally and output is displayed

f12 -> do nothing

ctrl + f12 -> opens active task window to select a task like as ctrl + f9

**Setting Build path(Eclipse classpath)**

Setting build path (eclipse classpath)

-> Right click on project

-> Build Path

-> Configure Build path

-> Click Libraries, Select classpath

-> Click Add External Jars

-> Select jar file from the required software

-> Click Apply and Close button

-> this jar will be shown under your project explorer

-> under Referenced libraries

**Changing Compiler/JRE system**

For changing compiler/JRE system

-> Right click on project

-> Build Path

-> Configure Build path

-> Project Facets

-> Click on Runtimes

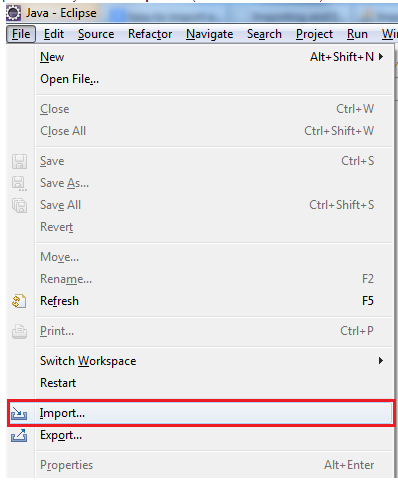
-> Unselect jdk check box

-> Under Facets selct Java version number

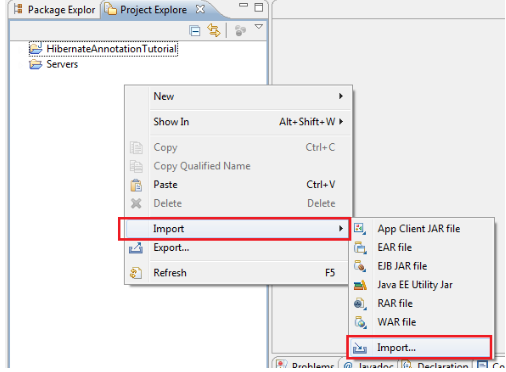
-> Click Apply and Close

16) How to Import external project into Eclipse workspace?

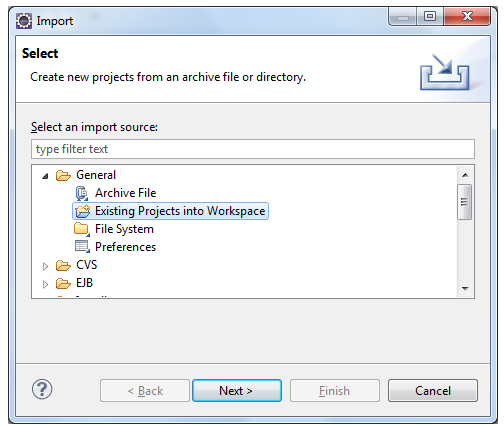
Step1:- File🡪import



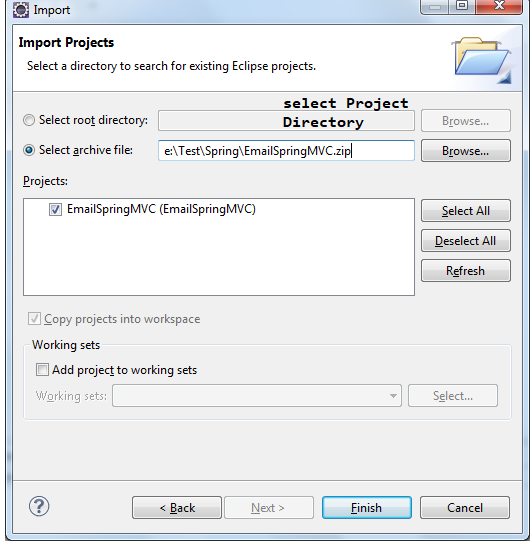
Step2:Import🡪 import



Step3: General🡪 Existing Projects into Workspace.

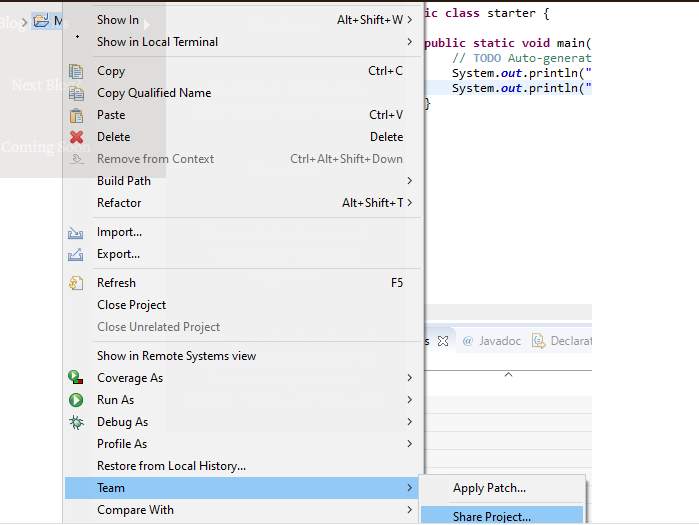


Step4: Enter the project Directory absolute path and click the finish button.



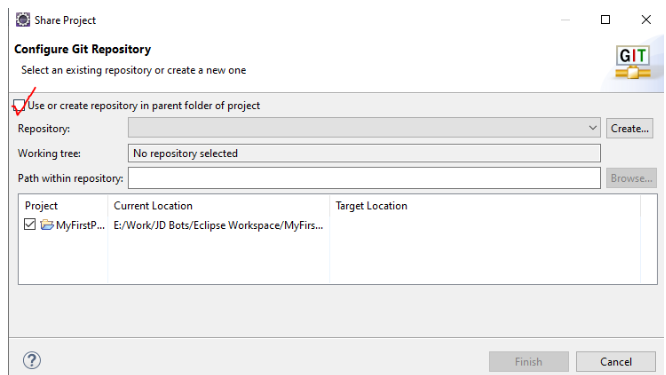
**17)How to push Java project to Git Hub Repositary?**

Step1: click on project folder in project explorer, select Team🡪share Project.

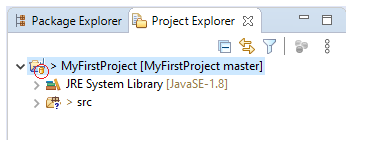


**Step-2:** use existing Local Repositary or create a new Local repository in local computer .

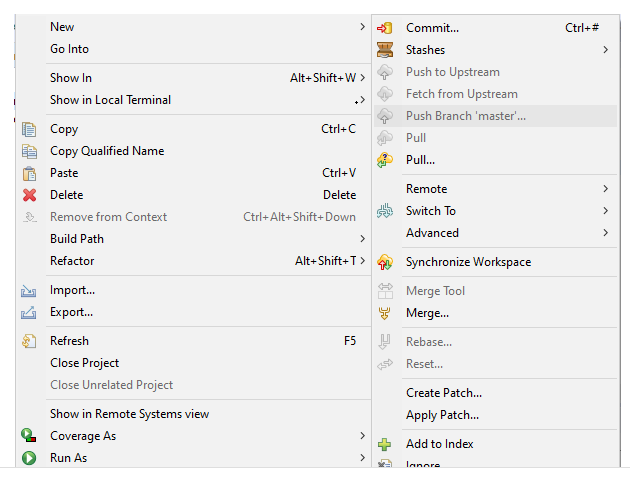
Configure project by selecting check box of right project. And click the ‘**finish**’ button.



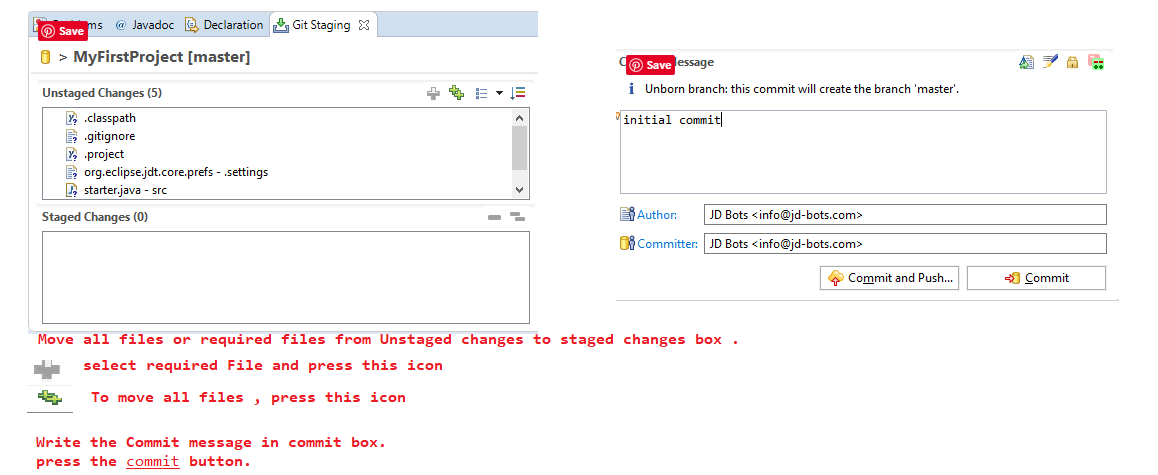
Step3: On project folder , we will yellow orange cylinder. It indicates that local repository is created.



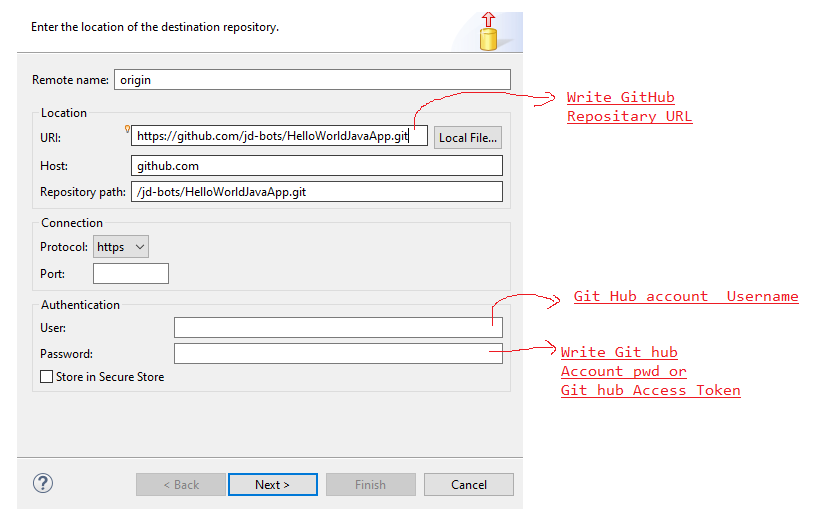
Step4: Right click on project folder 🡪 Team 🡪 commit



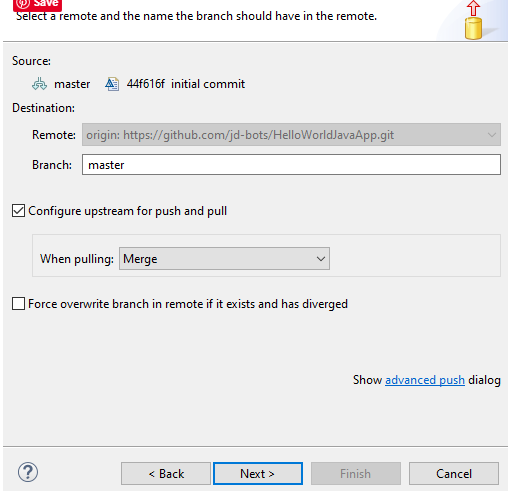
Step5: We will see the Git Staging window.



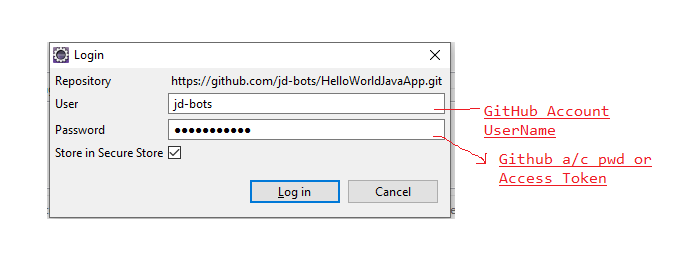
Step6: After writing the required details in window , press the next button.



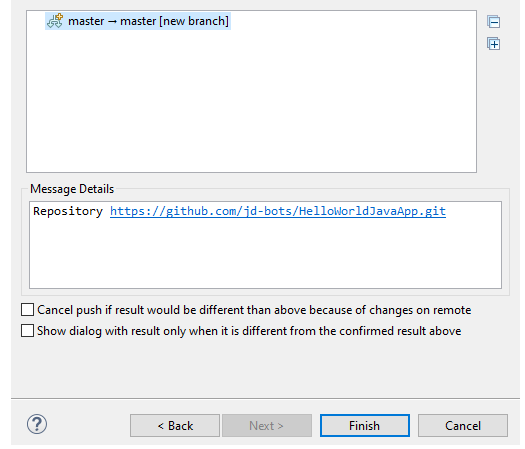
Step7: Choose the branch you want to push your changes to. The master branch is automatically populated.



Step8: login into Github



Step9: click on finish button.



Step10: If project pushed successfully, we see the below wizard.

