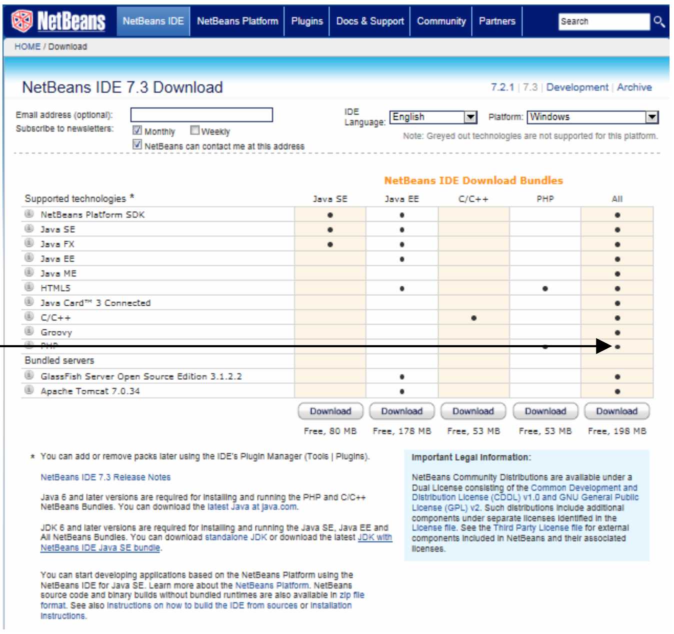
JAVA ~ as opposed to Java Script

STEP ONE: INSTALLING JAVA DEVELOPMENT TOOLS [NetBeans IDE 7.3]:-

To program with Java, the NetBeans IDE is a wise IDE to use. There are others.

https://netbeans.org/downloads/index.html



NetBeans IDE 7.3 Download

Email address (optional):
Subscribe to newsletters: [✓ Monthly] Weekly
[✓ NetBeans can contact me at this address]

IDE Language: English   Platform: Windows
Note: Greyed out technologies are not supported for this platform.

NetBeans IDE Download Bundles

Supported technologies *	Java SE	Java EE	C/C++	PHP	All
NetBeans Platform SDK	●	●			●
Java SE	●	●			●
Java FX	●	●			●
Java EE		●			●
Java ME					●
HTML5		●		●	●
Java Card™ 3 Connected					●
C/C++			●		●
Groovy					●
PHP				●	●
Bundled servers					
GlassFish Server Open Source Edition 3.1.2.2		●			●
Apache Tomcat 7.0.34		●			●
	Free, 80 MB	Free, 178 MB	Free, 53 MB	Free, 53 MB	Free, 198 MB

* You can add or remove packs later using the IDE's Plugin Manager (Tools | Plugins).

NetBeans IDE 7.3 Release Notes

Java 6 and later versions are required for installing and running the PHP and C/C++ NetBeans Bundles. You can download the latest Java at java.com.

JDK 6 and later versions are required for installing and running the Java SE, Java EE and All NetBeans Bundles. You can download standalone JDK or download the latest JDK with NetBeans IDE Java SE bundle.

You can start developing applications based on the NetBeans Platform using the NetBeans IDE for Java SE. Learn more about the NetBeans Platform. NetBeans source code and binary builds without bundled runtimes are also available in zip file format. See also instructions on how to build the IDE from sources or installation instructions.

Important Legal Information:

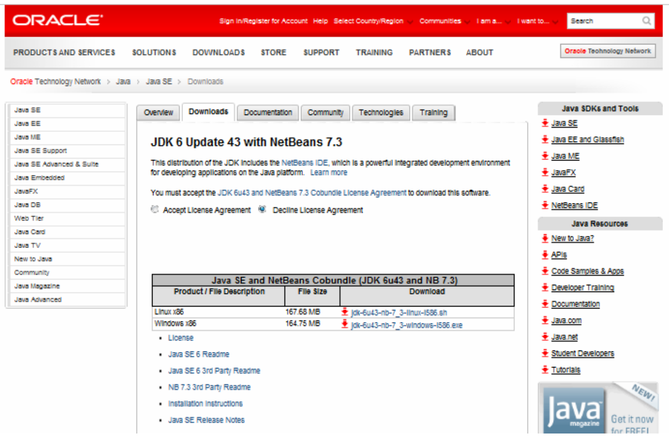
NetBeans Community Distributions are available under a Dual License consisting of the Common Development and Distribution License (CDDL) v1.0 and GNU General Public License (GPL) v2. Such distributions include additional components under separate licenses identified in the License file. See the Third Party License file for external components included in NetBeans and their associated licenses.

Elect to choose the complete system.

At which point you are asked to accept the terms, and then you can download the system. The following screen comes when you decide what system to use, its url for informational purposes is:-

http://www.oracle.com/technetwork/java/javase/downloads/jdk-netbeans-jsp-142931.html

After accepting the agreement, then the EXE file was downloaded.



JDK 6 Update 43 with NetBeans 7.3

This distribution of the JDK includes the NetBeans IDE, which is a powerful integrated development environment for developing applications on the Java platform. Learn more

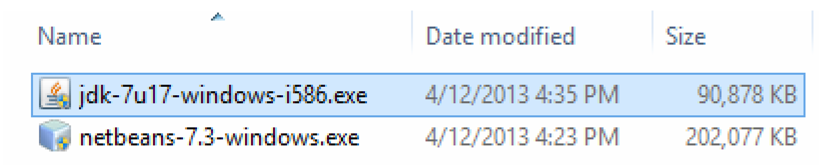
You must accept the JDK 6u43 and NetBeans 7.3 Cobundle License Agreement to download this software.

Accept License Agreement [✓ Decline License Agreement]

Java SE and NetBeans Cobundle (JDK 6u43 and NB 7.3)		
Product / File Description	File Size	Download
Linux x86	167.68 MB	jdk-6u43-nb-7_3-linux-i586.sh
Windows x86	164.75 MB	jdk-6u43-nb-7_3-windows-i586.exe

- License
- Java SE 6 Readme
- Java SE 6 3rd Party Readme
- NB 7.3 3rd Party Readme
- Installation Instructions
- Java SE Release Notes

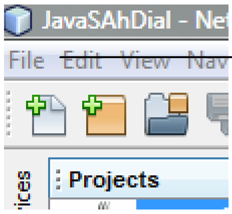
NOTE: When installing NetBeans, the installer searches for the JDK (Java SE development kit), if you do not have it because you downloaded NetBeans without the JDK, it gives you the url in a popup window. Either way, you would probably have files downloaded that look something like:-



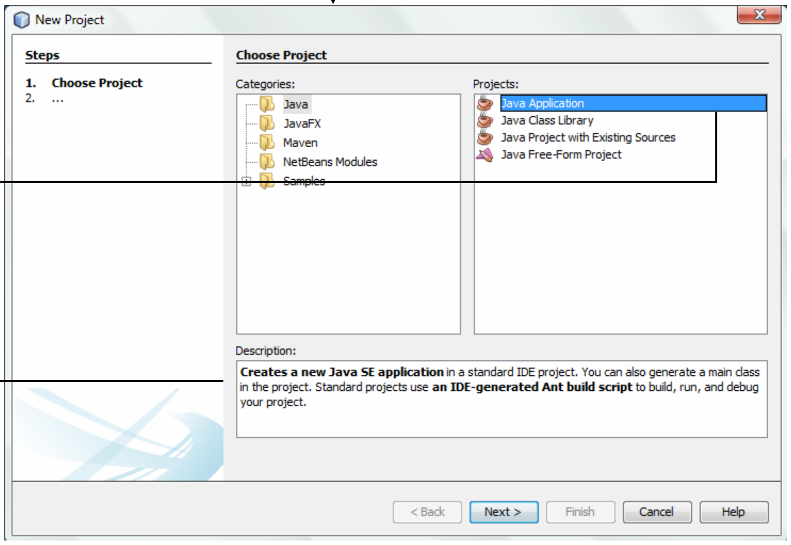
Name	Date modified	Size
jdk-7u17-windows-i586.exe	4/12/2013 4:35 PM	90,878 KB
netbeans-7.3-windows.exe	4/12/2013 4:23 PM	202,077 KB

STEP TWO: STARTING A NEW PROJECT:- hDialJavaStandAlone

In NetBeans IDE 7.3, FILE, NEW PROJECT was used to create an new application.

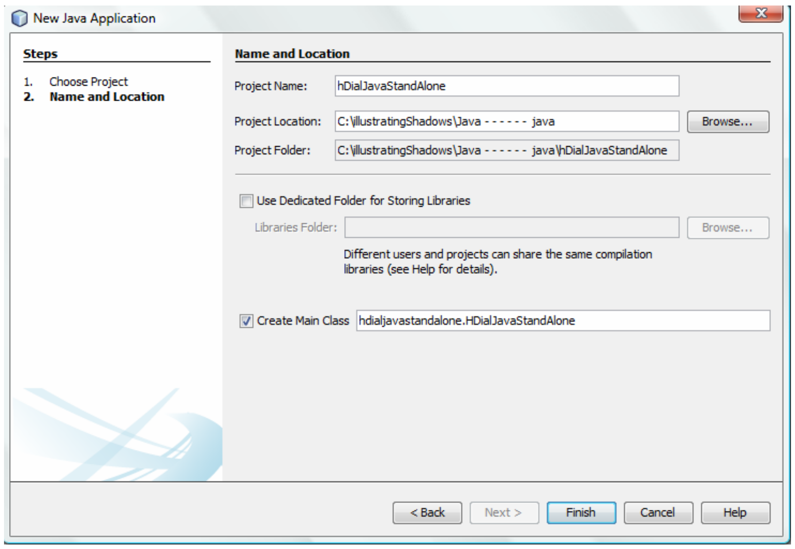


FILE, NEW PROJCT



Description:
Creates a new Java SE application in a standard IDE project. You can also generate a main class in the project. Standard projects use **an IDE-generated Ant build script** to build, run, and debug your project.

“JAVA APPLICATION: Creates a new Java SE application” in a standard IDE project. You can also generate a main class in the project. Standard projects use **an IDE-generated Ant build script** to build, run, and debug your project.



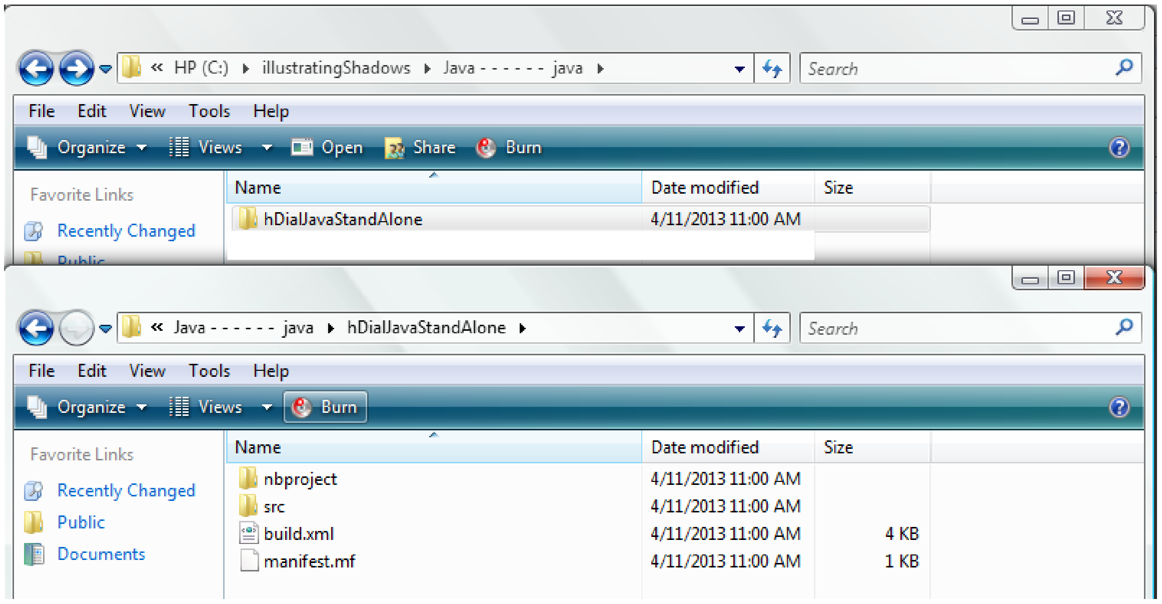
Project Name: hDialJavaStandAlone
Project Location: C:\illustratingShadows\Java - - - - - - java
Project Folder: C:\illustratingShadows\Java - - - - - - java\hDialJavaStandAlone

[ ] Use Dedicated Folder for Storing Libraries
Libraries Folder:
Different users and projects can share the same compilation libraries (see Help for details).

[✓] Create Main Class hdialjavastandalone.HDialJavaStandAlone

In NetBeans, the IDE (Integrated Development Environment), FILE, NEW PROJECT, and then JAVE APPLICATION. The project name chosen was:- hDialJavaStandAlone

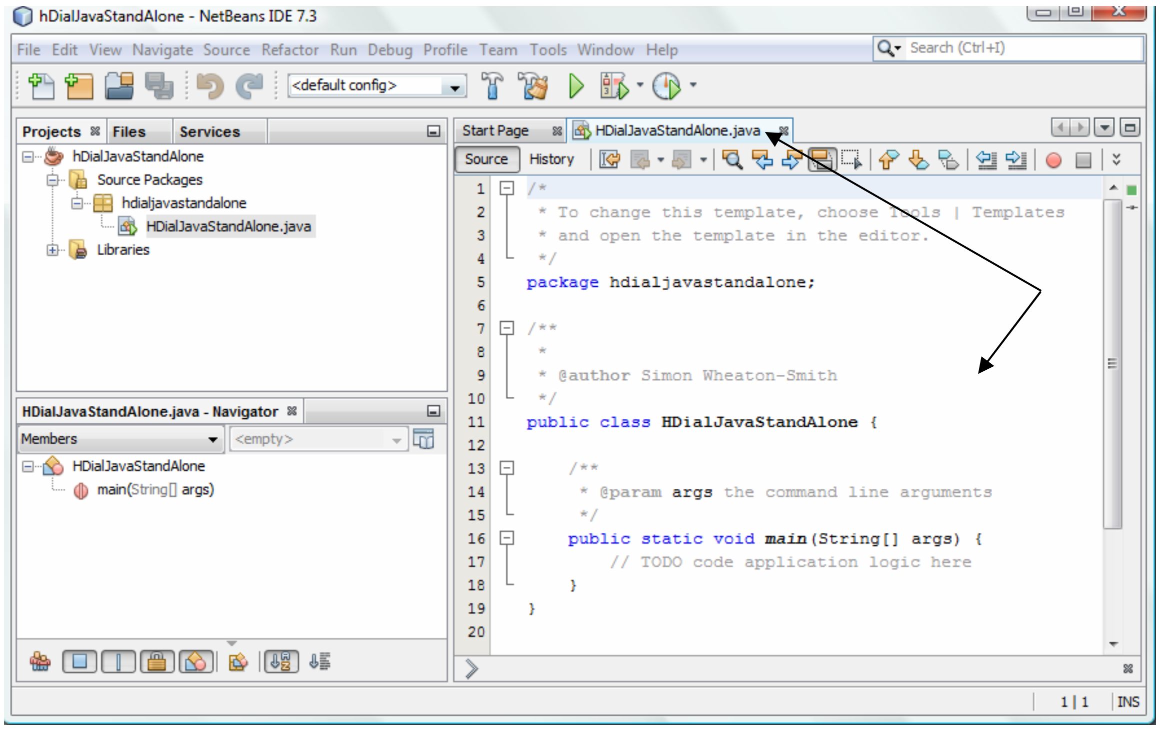
At this point the NetBeans has created a final folder.


« HP (C:) ▸ illustratingShadows ▸ Java - - - - - - java ▸
Search
File Edit View Tools Help
Organize Views Open Share Burn
Favorite Links
Recently Changed
Public
« Java - - - - - - java ▸ hDialJavaStandAlone ▸
Organize Views Burn
Documents

Name	Date modified	Size
hDialJavaStandAlone	4/11/2013 11:00 AM	

Name	Date modified	Size
nbproject	4/11/2013 11:00 AM	
src	4/11/2013 11:00 AM	
build.xml	4/11/2013 11:00 AM	4 KB
manifest.mf	4/11/2013 11:00 AM	1 KB

In NetBeans 7.3, the folder creation process was simple. At this point, the IDE also displays the entire project's skeleton.



```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package hdialjavastandalone;

/**
 *
 * @author Simon Wheaton-Smith
 */
public class HDialJavaStandAlone {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }
}
```

Which is nice of it, but irrelevant, because all our programming code will be event driven by buttons in a data input panel or form.

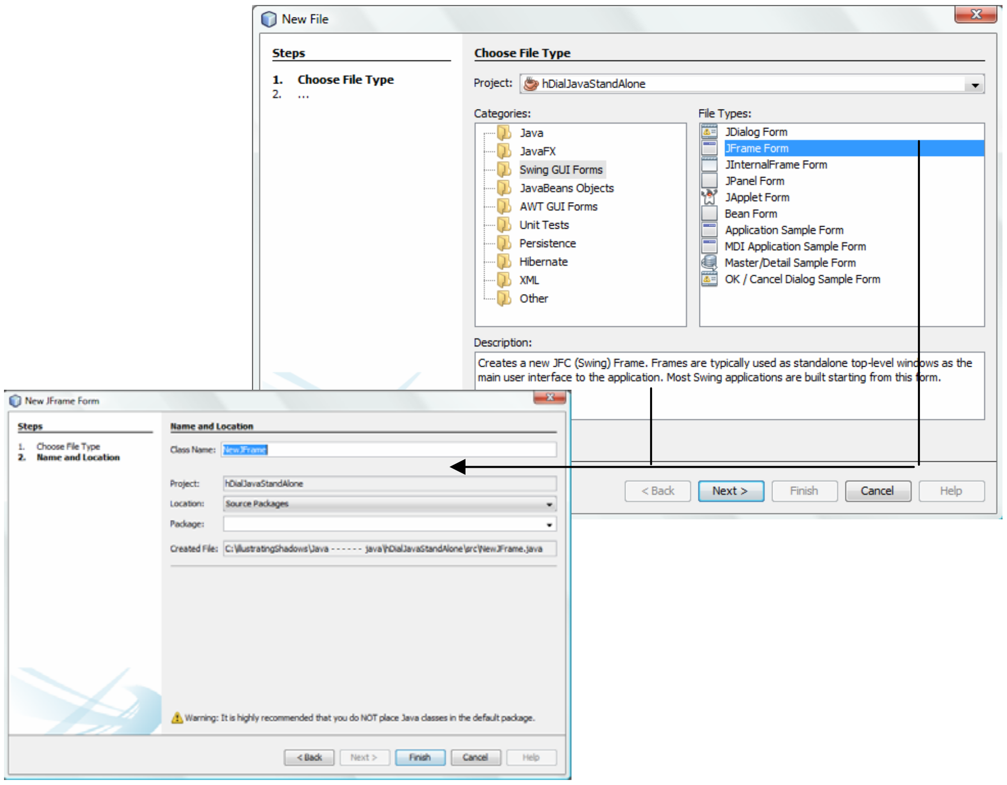
This section continues the "PBE" philosophy, namely "Programming By Example", there is no intent to explain the Java language, its inherent functions, nor its classes.

CONTINUING BUILDING THE PROGRAM

At any time, NetBeans can be exited, and upon re-entry the project can be re opened. When doing so, the START PAGE is shown on the right, and you may see it there, or on the upper left it may be there also. Either way, locate the project and open it. Once the data input panel or form is created, that is all you need in the IDE.

In these examples, the project name used was "hDialJavaStandAlone" which suggests a horizontal dial, in Java, and the stand alone Java as opposed to Java Script. As mentioned before, not much is going to go anywhere without user input, in other words a form. So, in the NetBeans, IDE, with the project open, select:-

FILE, NEW FILE, SWING GUI FORMS, JFRAME FORM



Project: hDialJavaStandAlone

Description:
Creates a new JFC (Swing) Frame. Frames are typically used as standalone top-level windows as the main user interface to the application. Most Swing applications are built starting from this form.

Class Name: NewJFrame

Project: hDialJavaStandAlone

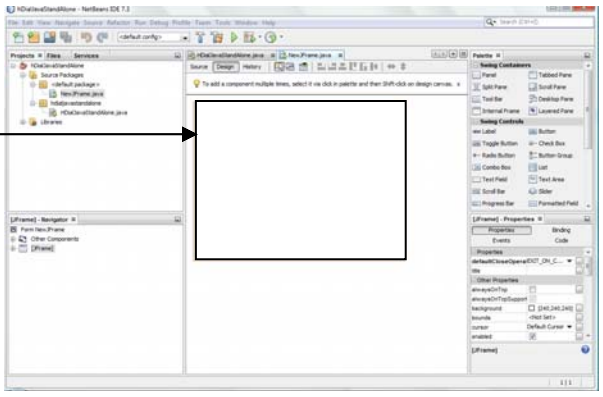
Location: Source Packages

Package:

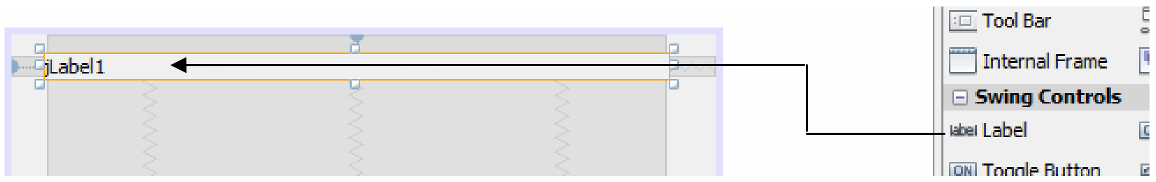
Created File: C:\IllustratingShadows\Java - - - - - - java\hDialJavaStandAlone\src\NewJFrame.java

Warning: It is highly recommended that you do NOT place Java classes in the default package.

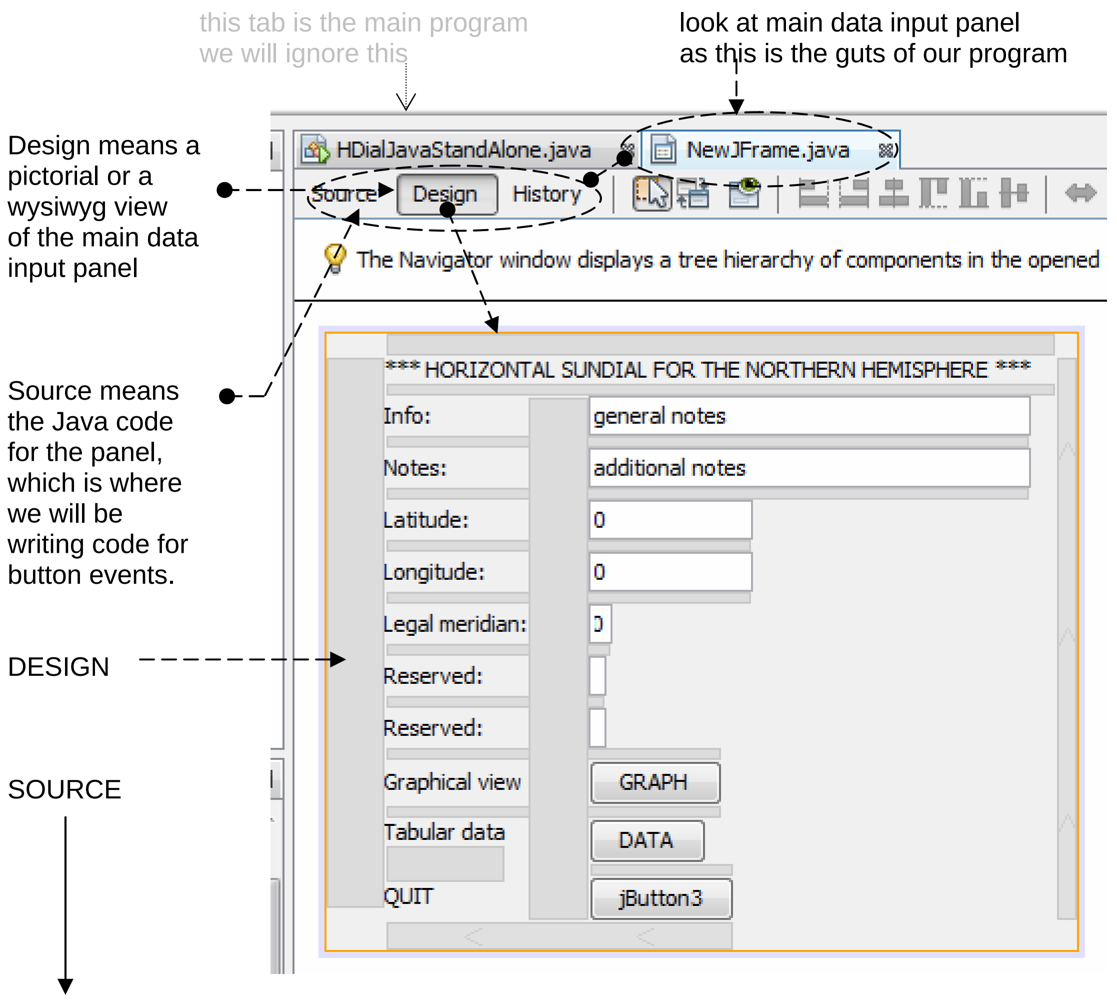
The default name for the panel was used, the highly original "NewJFrame". After a short while the frame is added to the application. Of course, it is a good idea to come up with a more meaningful name.



The FORM uses what is called “SWING”. So look for SWING CONTROLS on the right hand side, and LABEL was chosen, and dragged over to the form. This would be the header to say what this program is all about.



Some changes were made in that buttons, text input areas, and text descriptions were added.



this tab is the main program
we will ignore this

look at main data input panel
as this is the guts of our program

Design means a pictorial or a wysiwyg view of the main data input panel

Source means the Java code for the panel, which is where we will be writing code for button events.

DESIGN

SOURCE

The Navigator window displays a tree hierarchy of components in the opened

*** HORIZONTAL SUNDIAL FOR THE NORTHERN HEMISPHERE ***

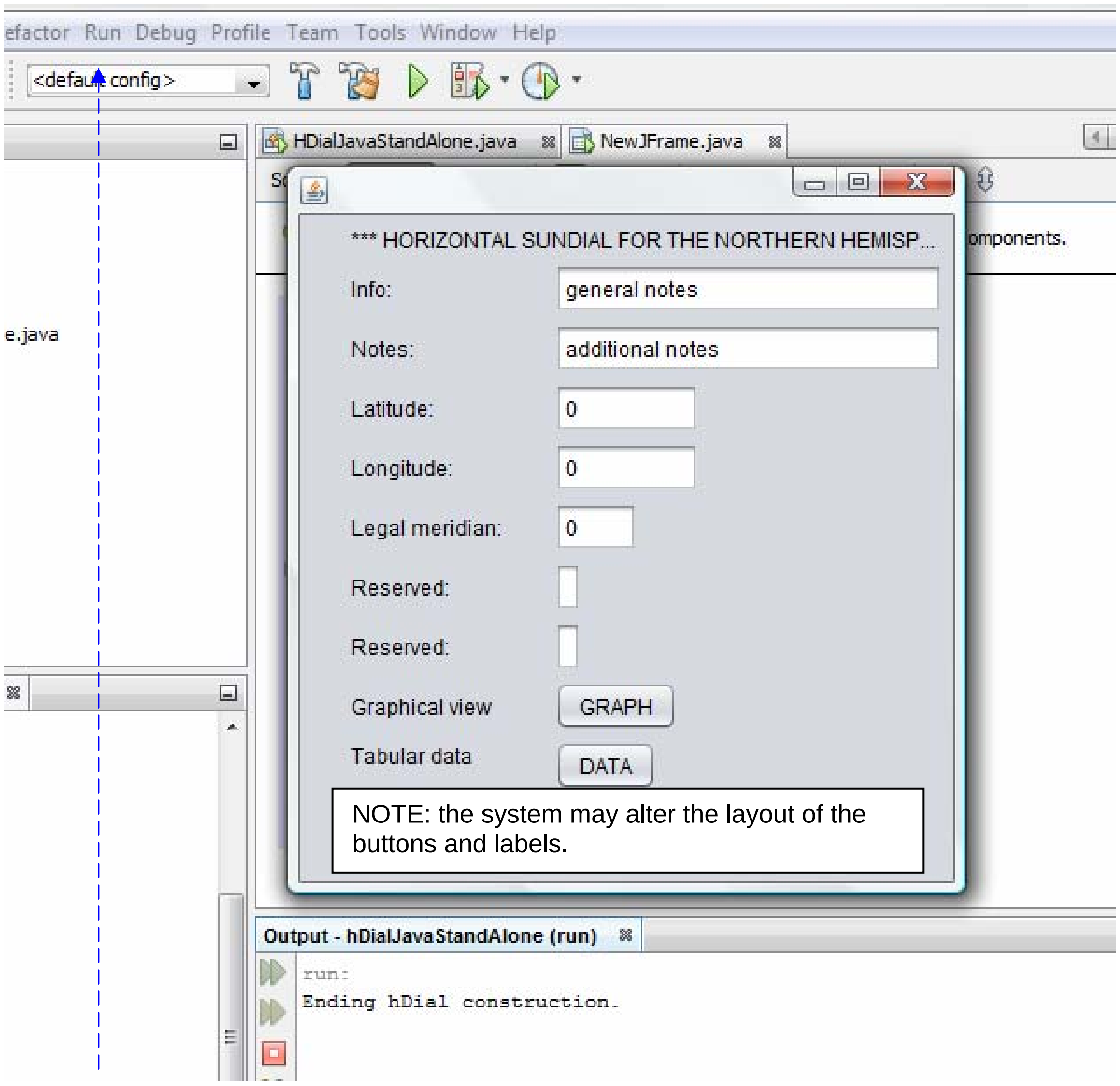
Info:	general notes
Notes:	additional notes
Latitude:	0
Longitude:	0
Legal meridian:	0
Reserved:	
Reserved:	
Graphical view	GRAPH
Tabular data	DATA
QUIT	jButton3

The code can be seen for the above panel by clicking on “Source”, and if you select the code area with ctrl-A to get all of it, and ctrl-C to copy it and then you paste it somewhere, you will see there is a ton of code, the window showing it hides it unless you tell it to show you the inner stuff. Below is the code of interest, because it is the code to be executed when one of the buttons is clicked.

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // G R A P H I C A L   D E P I C T I O N
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // T A B U L A R   D E P I C T I O N
}
```

At this point **FILE. SAVE ALL** was done and the main folder backed up along with its sub folders.

Then **RUN, RUN FILE** (which is **shift F6**) was used to test the form and the initial simple code. Then the **FILE, SAVE ALL**, and RUN, RUN FILE repeated to ensure the program and its data input panel were correct, so far. The results of the form and console output are shown below.



```
*** HORIZONTAL SUNDIAL FOR THE NORTHERN HEMISP...
Info:            general notes
Notes:           additional notes
Latitude:        0
Longitude:       0
Legal meridian:  0
Reserved:
Reserved:
Graphical view   GRAPH
Tabular data     DATA
```

NOTE: the system may alter the layout of the buttons and labels.

Output - hDialJavaStandAlone (run)

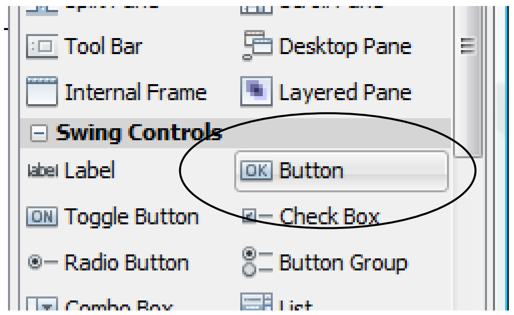
```
run:
Ending hDial construction.
```

(1)
RUN, RUN FILE is here

At this point, it was time to develop the main program. The first part would be buttons to set default latitude longitude data. Next would be the textual output and the graphical depiction of the dial.

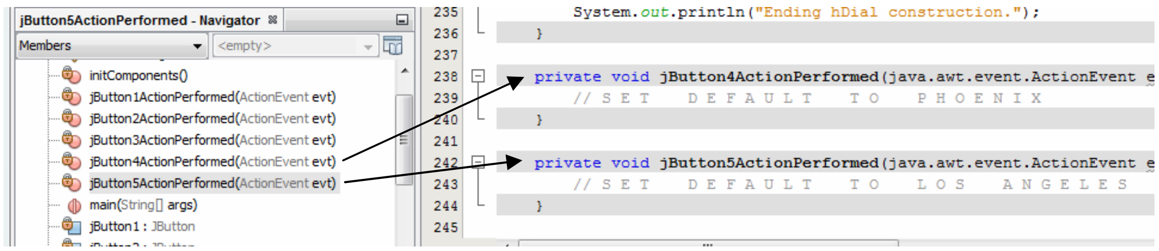
CODE TO ALLOW BUTTONS TO ALTER INPUT FIELDS IN THE FORM

A fourth button was added to set a default location “set PHX”, the button on the wysiwyg form layout. This was, as before the “OK BUTTON”.



The button was clicked slowly twice and its description changed to “set PHX”. Another button added for Los Angeles.

As before, that defined the button for display, it did not yet tell Java to generate a stub of code for when the jButton was clicked. Over to the lower right the buttons were located and clicked. And when double clicked, then a stub of code was added.


jButton5ActionPerformed - Navigator
Members
<empty>
initComponents()
jButton1ActionPerformed(ActionEvent evt)
jButton2ActionPerformed(ActionEvent evt)
jButton3ActionPerformed(ActionEvent evt)
jButton4ActionPerformed(ActionEvent evt)
jButton5ActionPerformed(ActionEvent evt)
main(String[] args)
jButton1 : JButton

```
        System.out.println("Ending hDial construction.");
    }

    private void jButton4ActionPerformed(java.awt.event.ActionEvent e
        // S E T   D E F A U L T   T O   P H O E N I X
    }

    private void jButton5ActionPerformed(java.awt.event.ActionEvent e
        // S E T   D E F A U L T   T O   L O S   A N G E L E S
    }
```

Now, code must be added to modify the fields in the data input panel. It will go into those stubs when the “set PHX” or “set LAX” buttons are clicked.

```
    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
        // S E T    D E F A U L T    T O    P H O E N I X
        // jTextField4   latitude
        // jTextField5   longitude
        // jTextField6   legal meridian
        jTextField4.setText("33.5");
        jTextField5.setText("112.1");
        jTextField6.setText("105");

    }

    private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
        // S E T    D E F A U L T    T O    L O S    A N G E L E S
        jTextField4.setText("34.0");
        jTextField5.setText("118.4");
        jTextField6.setText("120");
    }
```

The field name “jTextField4” for example is modified with the method “setText” whose parameter is the new data. The opposite method is “getText” which retrieves the data, and will be used in the final output code.

- http://www.beginner-java-tutorial.com/j-intswing-a4.pdf

is a tutorial on the getText and setText methods.

THE CODE FOR THE TABULAR AND GRAPHICAL PROCEDURES

FIRST – required code because of the need for an import for graphical support

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/* *******************************************************************
   **    BELOW WAS ADDED BY ILLUSTRATING SHADOWS AS REQUIRED         **
   *******************************************************************
*/
import java.awt.Graphics;
import java.awt.Color;
/* *******************************************************************
*/
```

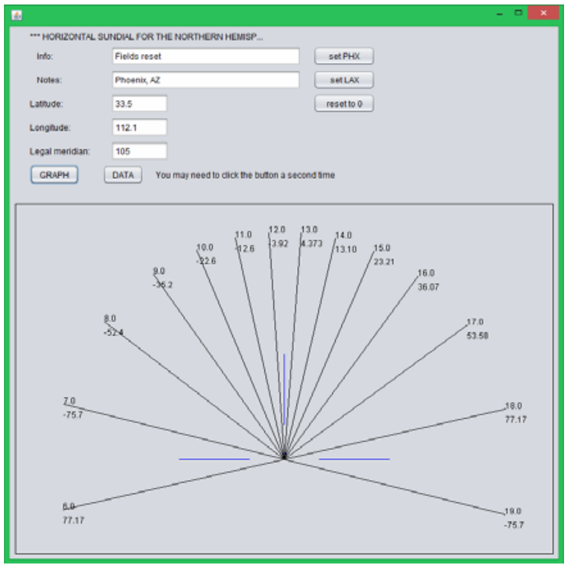
SECOND – code for the graphical display

```
    /*
     * This must be added to start of source file so our graphics
     * program functions or command will be recognised.
     *
     import java.awt.Graphics;
    */
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // G R A P H I C A L    D E P I C T I O N
        /* ********************************************************* *
        * *   this is GRAPHICAL DEPICTION button handler             * *
        * ********************************************************* *
        */

        // create a box area for graphics: graph coordinates are integer
        int leftx, rightx, ctrx, topy, boty, ctry;
        leftx  = 001;
        rightx = 799;
        ctrx   = leftx+(rightx-leftx)/2;
        topy   = 300;
        boty   = 800;
        ctry   = topy + (boty-topy)/2;

        Graphics swsxxy = getGraphics();       // establish a graph area
        setSize (rightx+1,boty+1);             // of such and such a size
        // some of these functions are defined at:-
        // docs.oracle.com/javase/1.5.0/docs/api/java/awt/Graphics.html

        swsxxy.setColor(Color.blue) ;          // border color
        swsxxy.clearRect(leftx,topy,rightx-leftx,boty-topy);
        //     drawRect(  );                   // swsxxy.fillRect(   );

        // keep the buttons showing in case they were in the graph area
        jButton1.setVisible(true);       jButton2.setVisible(true);

        // set dial center and draw an alignment line
        int dcx, dcy, r;
        dcx = ctrx;        // dial center x
        dcy = boty-150;    // dial center y
        r   = 150;         // size of an hour line
        swsxxy.drawLine(dcx,dcy-50,dcx,dcy-r);
        swsxxy.drawLine(dcx-50,dcy,dcx-r, dcy);
        swsxxy.drawLine(dcx+50,dcy,dcx+r, dcy);
        swsxxy.drawString("*", dcx,dcy);
        r   = 325;         // size of an hour line

        // start the graphical depiction
        double hr, ha, hla, hlat, myLat, myLng, myRef;
        double x,y;
        myLat = Float.parseFloat(jTextField4.getText());
        myLng = Float.parseFloat(jTextField5.getText());
        myRef = Float.parseFloat(jTextField6.getText());
        swsxxy.setColor(Color.black) ;    // border color
        for ( hr = 6; hr<20; hr=hr+1) {
                // hlaD = hlaH (hr, myLat, myLng, myRef)
                ha    = hr*15 - (myLng-myRef) ;
                hlat  = Math.sin(myLat*2*3.1416/360) *
                        Math.tan(ha*2*3.1416/360);
                hla   = 360*(Math.atan(hlat))/(2*3.146);
                x=r*Math.sin(2*3.1416*hla/360);
                y=r*Math.cos(2*3.1416*hla/360);
                /* late afternoon code to stop it showing up as morning */
                if (hr > 12) {
                   if (hla < 0) {
                      x = -x;   y = -y;
                   }
                }
                /* early morning code to stop it showing up as afternoon*/
                if (hr < 12) {
                   if (hla > 0) {
                      x = -x;   y = -y;
                   }
                }
                swsxxy.drawLine((int)dcx,(int)dcy,(int)dcx+(int)x, (int)dcy-(int)y);
                String h = ""+hr;
                swsxxy.drawString(h, (int)dcx+(int)x, (int)dcy-(int)y );
                String a0, a1 ;
                a0 = String.format("%f", hla);
                a1 = a0.substring(0,5);
                swsxxy.drawString(a1, (int)dcx+(int)x, (int)dcy+20-(int)y );
        }

        leftx = leftx+15; rightx = rightx-15;
        topy  = topy-15;  boty   = boty-15;
        swsxxy.drawLine((int)rightx,(int)topy,(int)rightx, (int)boty);
        swsxxy.drawLine((int)leftx, (int)topy,(int)leftx,  (int)boty);
        swsxxy.drawLine((int)leftx, (int)topy,(int)rightx, (int)topy);
        swsxxy.drawLine((int)leftx, (int)boty,(int)rightx, (int)boty);

    }
```

THIRD – code for the tabular display

```
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        /* ********************************************************** *
        * *   this is the TABULAR DEPICTION button handler           * *
        * ********************************************************** *
        */

        // create a box area for graphics
        int leftx, rightx, ctrx, topy, boty, ctry;
        leftx  = 001;
        rightx = 799;
        ctrx   = leftx+(rightx-leftx)/2;
        topy   = 300;
        boty   = 800;
        ctry   = topy + (boty-topy)/2;
        Graphics swsxxy = getGraphics();   // THIS LINE IS CRITICAL
        setSize (rightx+1,boty+1);         // same as preceding { ... }
        // some of these functions are defined at:-
        // docs.oracle.com/javase/1.5.0/docs/api/java/awt/Graphics.html
        swsxxy.setColor(Color.blue) ;      // border color
        swsxxy.clearRect(leftx,topy,rightx-leftx,boty-topy);
        //     drawRect(  );               //     swsxxy.fillRect(   );

        // keep the buttons showing in case they were in the graph area
        jButton1.setVisible(true);         jButton2.setVisible(true);

        // set dial center and draw an alignment line
        int dcx, dcy, r;
        dcx = ctrx;         // dial center x
        dcy = boty-150;     // dial center y
        r   = 150;          // size of an hour line
        r   = 350;          // size of an hour line

        // start the textual depiction
        double hr, ha, hla, hlat, myLat, myLng, myRef;
        double x,y;
        myLat = Float.parseFloat(jTextField4.getText());
        myLng = Float.parseFloat(jTextField5.getText());
        myRef = Float.parseFloat(jTextField6.getText());
        swsxxy.setColor(Color.black) ;    // border color
        int nextLine  = topy;
        nextLine = nextLine+20;
        swsxxy.drawString("HOUR", 100,(int)nextLine );
        swsxxy.drawString("HOUR LINE ANGLE", 200,(int)nextLine );
        nextLine = nextLine+15;
        for ( hr = 6; hr<20; hr=hr+1) {
                // hlaD = hlaH (hr, myLat, myLng, myRef)
                ha    = hr*15 - (myLng-myRef) ;
                hlat  = Math.sin(myLat*2*3.1416/360) *
                        Math.tan(ha*2*3.1416/360);
                hla   = 360*(Math.atan(hlat))/(2*3.146);

                String h = ""+hr;
                swsxxy.drawString(h, 100, (int)nextLine );
                String a0, a1 ;
                a0 = ""+String.format("%f", hla);
                a1 = a0.substring(0,5);
                swsxxy.drawString(a1, 200,(int)nextLine );
                nextLine = nextLine+20;
        }
        leftx = leftx+15; rightx = rightx-15;
        topy  = topy-15;  boty   = boty-15;
        swsxxy.drawLine((int)rightx,(int)topy,(int)rightx, (int)boty);
        swsxxy.drawLine((int)leftx, (int)topy,(int)leftx,  (int)boty);
        swsxxy.drawLine((int)leftx, (int)topy,(int)rightx, (int)topy);
        swsxxy.drawLine((int)leftx, (int)boty,(int)rightx, (int)boty);

    }
```

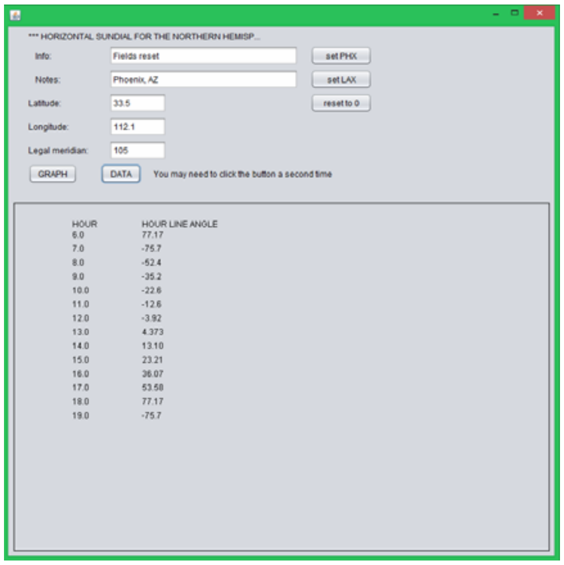
The final form and graphical and tabular output



*** HORIZONTAL SUNDIAL FOR THE NORTHERN HEMISP...

Info:	Fields reset
Notes:	Phoenix, AZ
Latitude:	33.5
Longitude:	112.1
Legal meridian:	105

You may need to click the button a second time



*** HORIZONTAL SUNDIAL FOR THE NORTHERN HEMISP...

Info:	Fields reset
Notes:	Phoenix, AZ
Latitude:	33.5
Longitude:	112.1
Legal meridian:	105

You may need to click the button a second time

```
HOUR      HOUR LINE ANGLE
6.0       77.17
7.0       -75.7
8.0       -52.4
9.0       -35.2
10.0      -22.6
11.0      -12.6
12.0      -3.92
13.0      4.373
14.0      13.10
15.0      23.21
16.0      36.07
17.0      53.58
18.0      77.17
19.0      -75.7
```

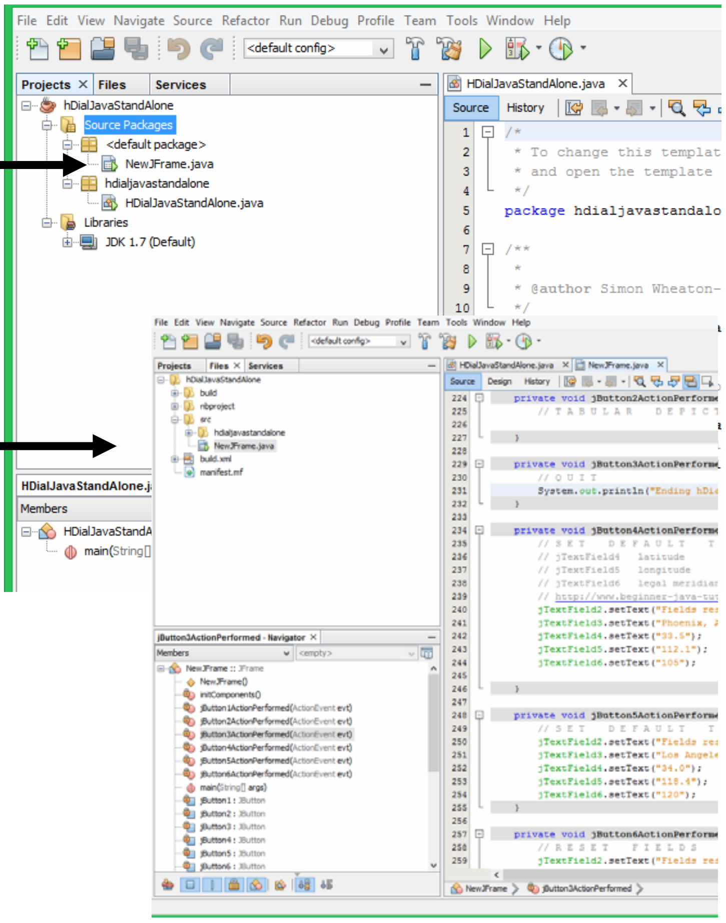
PROBLEM: IF YOU DO NOT SEE YOUR FORM WHEN STARTING NETBEANS

The top left panel in NetBeans allows sub folders to be clicked and their files to be opened. In this manner the (a) main form, (b) the main code, and (c) the event driven code for the buttons will once again appear.

When the PROJECT is opened the data input panel is not always automatically shown. To bring it into view, try the following:-

data input panel
may be in
DEFAULT PACKAGE

or may be in
another folder



```
/*
 * To change this templat
 * and open the template
 */
package hdialjavastandalo

/**
 *
 * @author Simon Wheaton-
 */
```

```
    private void jButton2ActionPerforme
        // T A B U L A R   D E P I C T

    }

    private void jButton3ActionPerforme
        // Q U I T
        System.out.println("Ending hDia
    }

    private void jButton4ActionPerforme
        // S E T   D E F A U L T   T
        // jTextField4   latitude
        // jTextField5   longitude
        // jTextField6   legal meridiar
        // http://www.beginner-java-tut
        jTextField2.setText("Fields res
        jTextField3.setText("Phoenix, A
        jTextField4.setText("33.5");
        jTextField5.setText("112.1");
        jTextField6.setText("105");

    }

    private void jButton5ActionPerforme
        // S E T   D E F A U L T   T
        jTextField2.setText("Fields res
        jTextField3.setText("Los Angele
        jTextField4.setText("34.0");
        jTextField5.setText("118.4");
        jTextField6.setText("120");
    }

    private void jButton6ActionPerforme
        // R E S E T   F I E L D S
        jTextField2.setText("Fields res
```

PROBLEM: IF RUN, RUN FILE DOES NOT WORK:-

Change something in the source, and repeat.

PROBLEM: EXECUTING A ".JAVA" FILE IN WINDOWS 8. While the ".java" file type can be associated with "java.exe" in the program associations, and while Java will be invoked when double clicking a ".java" file, Java itself fails as it cannot find some of the things it needs, such as the ".class" files. NetBeans says that doing a "RUN, CLEAN AND BUILD" will create a JAR file, and Windows 8 has ".jar" associated with Java, but nothing happens. In reality, Java itself is not a priority in Windows 8 so at best, Windows 8 can develop and test within NetBeans. But running Java programs in Windows 8 may become a thing of the past.

PROBLEM: User functions for common programming

Functions are not a part of Java, use a class and methods instead. In other words you are forced into object oriented programming.

THE FOLLOWING ARE USEFUL NOTES FROM THE EARLIER VERSION OF JAVA NOTES

There are several books that are helpful, one is "JAVA In Easy Steps" by Mike McGrath, and is based on a text based non graphical development environment such as JPadPro. Another includes an SDK on a CD and is "Programming With JAVA in 24 Hours" by Rogers Cadenhead that gets into graphical IDEs such as NetBeans. A third is "Core JAVA 2 – Volume 1 – Fundamentals" by Cay Horstmann and Gary Cornell which has useful internal theory.

TEXT STRINGS AND STRICT TYPING: Code would be added to extract the TEXT STRINGS of longitude and legal meridian, convert them to FLOAT, perform the math, and convert them back to STRING. This sounds simple however the JAVA IDE "help" system is not overburdened with practical examples. A search on the web did locate the "float to string" function using HELP, SUPPORT AND DOCS ONLINE, then selecting "The JAVA Tutorial" and following links until the helpful url was found:-

http://java.sun.com/docs/books/tutorial/java/data/strings.html

This showed the "String.format" function. Finding useful functions complicates learning JAVA since what are normally language functions may now be an object's methods.

```
// derive the dial location and hour correction
s1 = jTextField2.getText();
s2 = jTextField3.getText();
t1 = Float.parseFloat(s1);
t2 = Float.parseFloat(s2);
t3 = 4*(t1-t2)/60 ;
// show hour correction
s3 = String.format("%f", t3);
jLabel6.setText ( s3 );
```

MATHEMATICAL FUNCTIONS

At this point the math functions are needed, and the online help was used using HELP, SUPPORT AND DOCS ONLINE, then selecting "The JAVA Tutorial" and following links until the helpful url was found:-

http://java.sun.com/javase/6/docs/api/java/lang/Math.html

BENEFITS OF TRUE OBJECT ORIENTED CLASSES WITH METHODS

Not shown in the program that follows is the repainting of a java window when it is covered up by some other window. A true benefit of a fully implemented object oriented system is that methods belonging to classes can be interlinked or inter-related, and invoked "when things happen". One such method may be invoked to redraw a java window when that window has been affected by some other window.