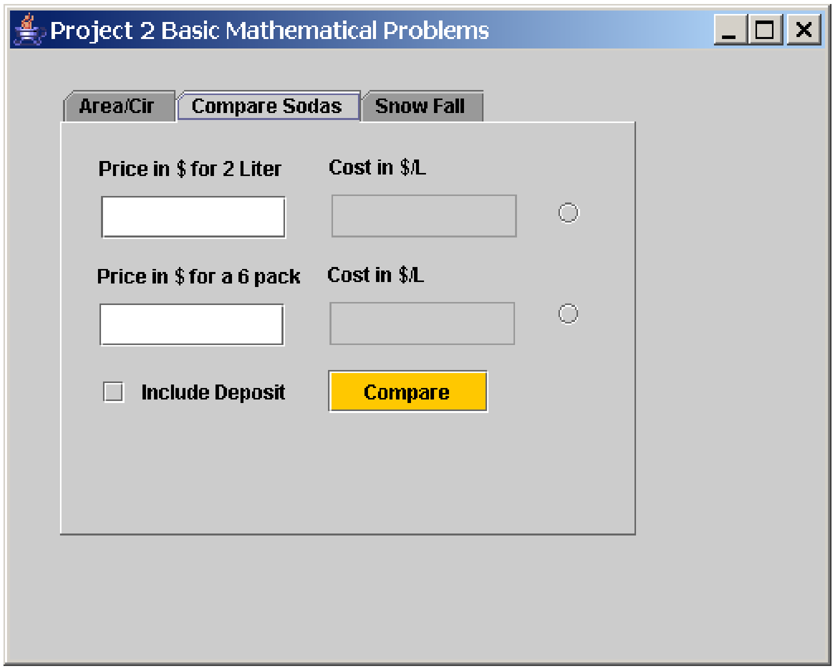
Hanley’s Survival Guide for Visual Applications with NetBeans 2.0
Last Updated: 5/20/2015

TABLE OF CONTENTS

“Never trust a computer you can't throw out a window.”
Steve Wozniak

Overview:
Visual applications allow a user to interact using a mouse or keyboard to click on buttons, pull down menus, select radio buttons, tabbed panes and other items. Thanks to Xerox Palo Alto Research Center and Steve Jobs(who saw these ideas at Xerox and brought them to computers!)

Building Visual applications is more complex than building console applications, but the rewards are greater also.



You should know some terms when dealing with GUIS (Graphical User Interfaces).

class: a class is a software blueprint, it is a category of things. It is a “cookie cutter” for objects. Example JButton

object: an object is an instance of a class. It is a running software component that lives inside the memory of a computer. They often have data and actions (methods) that they can perform. Example compareBUT is the name of the JButton in the above JFrame that triggers the program to compute and compare the prices of the sodas input. It is an instance of the JButton class. Running programs are made up of objects, not classes. Much like a house is the reality of a blue-print. It is the instanciation.

JFrame: a software blueprint that represents a window that the user can interact with

JLabel: a phrase or series of characters in a JFrame or Dialog

event: something that happens during the run of a program that could be user initiated or system initiated
Examples mouseClicked, menuItemSelected, timer event, keyTyped, keyPressed, keyReleased

actionListener: a program that can respond to events

private void initcomponents(): a method that is used by visual applications to initialize the window

public HellowWordFrame(){

}: this method is known as a constructor. Constructors are special methods that are starting points for objects. Each java object is the result of a constructor

setBounds(): this method establishes the bounds of a particular graphical element inside the window

new JLabel() or new JTextField() or new JButton(): the new command is what calls a constructor and instanciates an object. This is how objects begin occupying the computer's memory

ActionEvent: a software blueprint that represents an event that can occur during the running of a program to a JFrame.   Examples see above for event

String: a variable type that can store any sequence of characters.
Examples "this is a string" "yy$%he **" "-----Title----" "789" " %"

double: a primitive variable type that can store numbers which may include decimal parts Examples 3.14159 4000.1 -100.67896 16

Double: a software blueprint that allows a programmer to use and format double math values in various ways

Double.parseDouble(): this command takes an existing String and attempts to convert it to a double math value. Example
Double.parseDouble("134.78") yields 134.78 the numeric value
Double.parseDouble("134ea") yields an exception (a run time error)

int: a primitive variable type that can store numbers which can be integers and cannot have decimal parts

Integer: see Double above

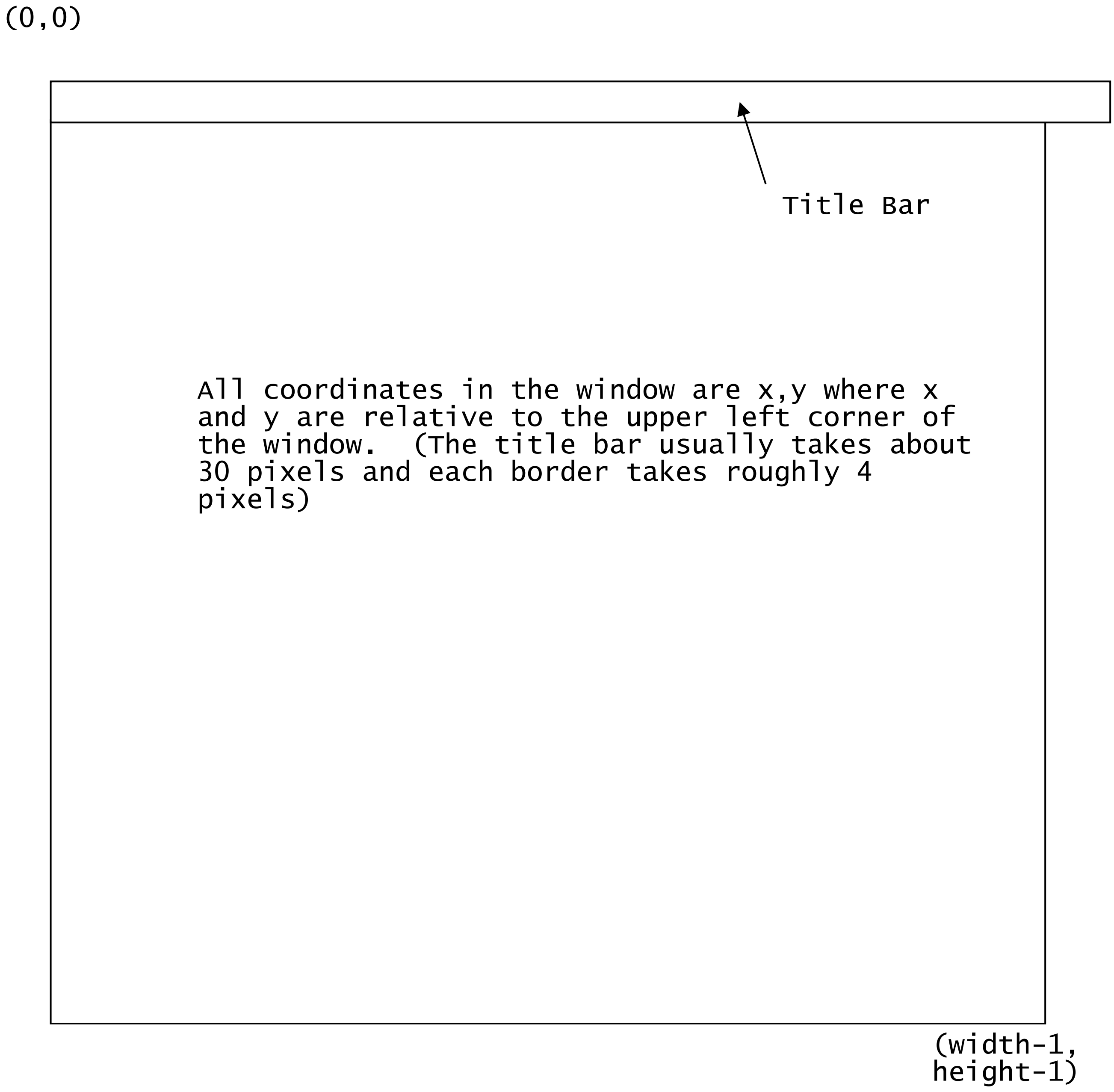
Integer.parseInt(): see Double.parseDouble() above

LayoutManager: a software blueprint that represents a component which manages the placement of visual objects in a window. Example GridLayout places components in a window by starting at the upper left and then working its way across to the end of a row before moving to the next row. Finishes at lower right. null means that all visual objects will be placed absolutely.

```
private void ansTFActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }
```

method that is called when an event occurs. Also known as an event handler. Gets activated for mouse events, timer events and menu events. Does not get activated for keyboard events!

focus: a component that has the focus is “active” and able to receive key events. Only one component can have the focus at a time.
Typical Java Frame



All coordinates in the window are x,y where x and y are relative to the upper left corner of the window. (The title bar usually takes about 30 pixels and each border takes roughly 4 pixels)

Rectangle: a software blueprint that manages a rectangle using 4 numbers, the upper left x coord, upper left y coord, the width and the height.

new: one of the most important command in Java and C++. This command allows a programmer to create a new software object from a software blueprint. Example JLabel helloLBL = new JLabel(); NOTE: JLabel is a class, helloLBL is an object

addActionListener(): java method that sets up a program to listen for events that are directed at a specific component

this: a way of referring to the current object you are dealing with, in our case it will refer to the current frame

Swing: a collection of user interface components that were developed to replace some of the original components which really didn't work that well.

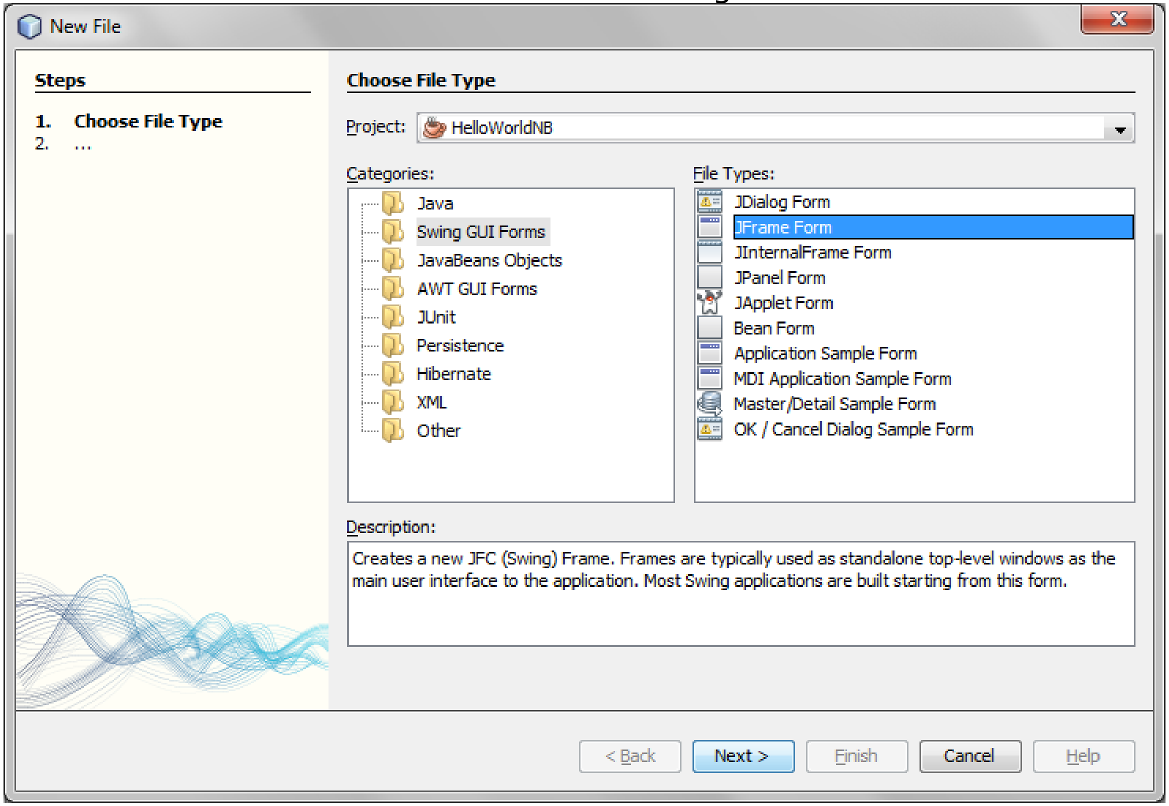
<u>A typical swing project:</u>

HelloWordApp.java – this class contains a main method which creates an instance of the HelloWordFrame class. This instance is set to visible and thus controls the running of the application.

HelloWorldFrame.java – this class extends the JFrame class, which means that it inherits all of the features of the JFrame class. This class contains lots of code. The NetBeans Visual Designer will generate much of the code for you.

To create a visual application:

1. Create a new Project within NetBeans (HelloWorld Project)
2. Choose New..File and then select Swing GUI Forms JFrame form

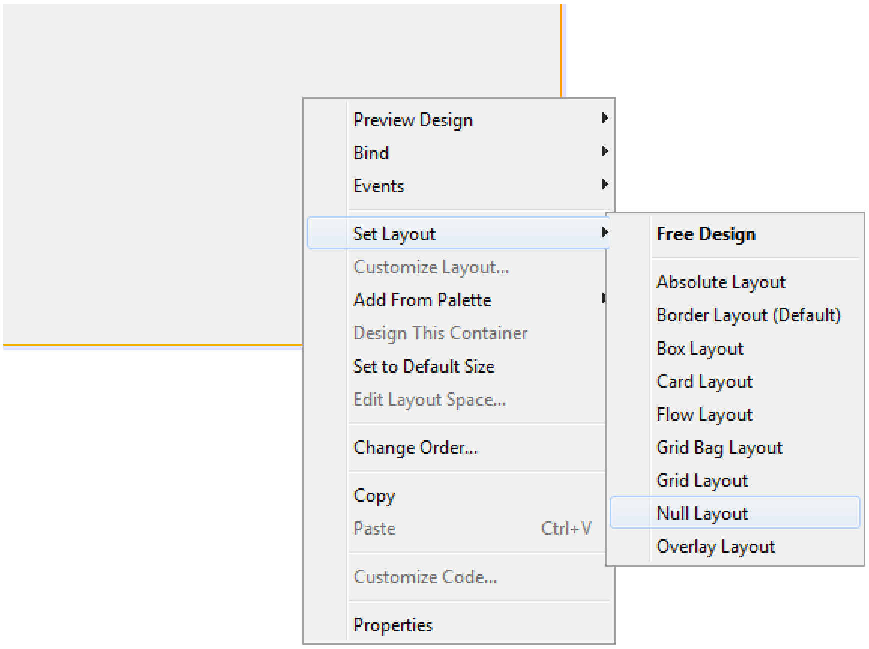


Project: HelloWorldNB

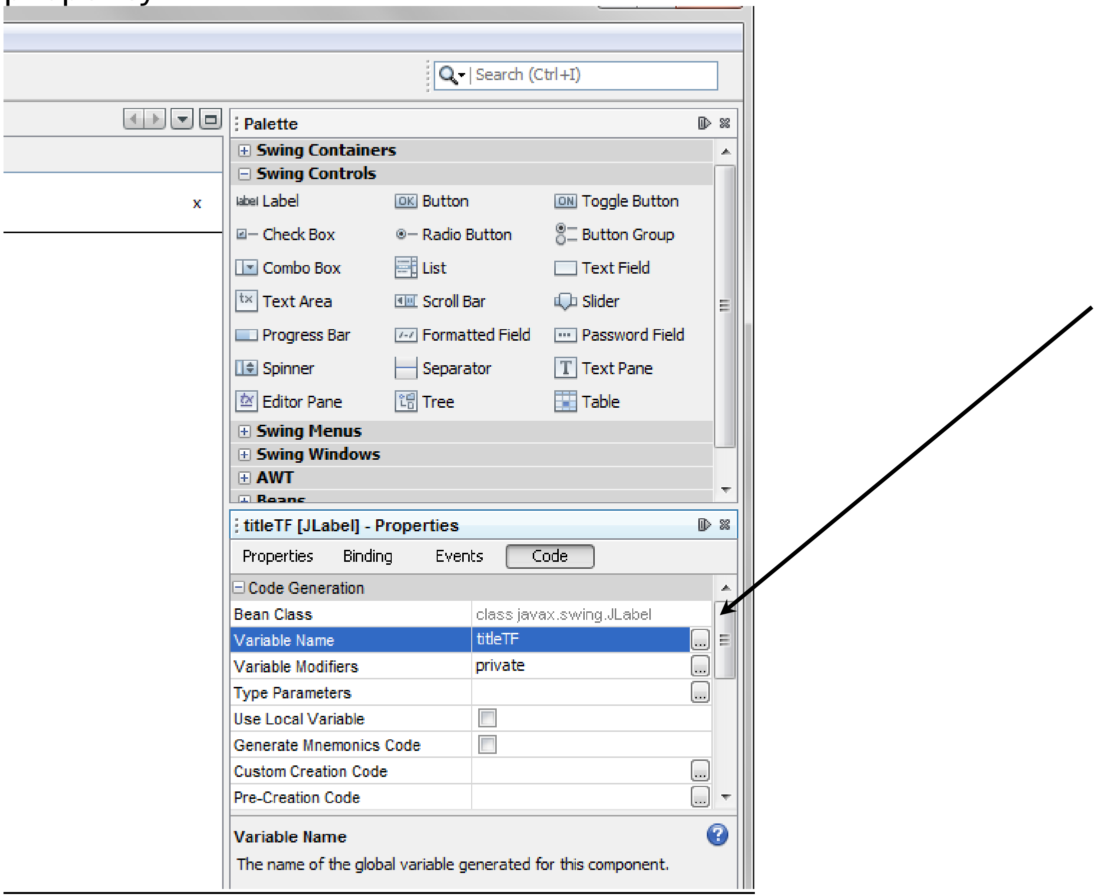
Description:

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.

3. Type in the name HellowWorldFrame and choose Finish
4. RIGHT CLICK IN THE MIDDLE OF THE WINDOW AND SET LAYOUT TO NULL!!!



5. Drag a JLabel out and set its name to titleLBL by clicking on the Code button on the right hand side and setting the Variable name property



Bean Class	class javax.swing.JLabel
Variable Name	titleTF
Variable Modifiers	private
Type Parameters	
Use Local Variable	□
Generate Mnemonics Code	□
Custom Creation Code	
Pre-Creation Code	

Variable Name

The name of the global variable generated for this component.

6. Click back on the Properties button to set the Font for this label
7. Keep adding components as needed
8. Double click on your button
   NetBeans will generate a method for you like;
   private void hwBUTActionPerformed(java.awt.event.ActionEvent evt){

```
        // TODO add your handling code here:
    }
```

9. Put any logic needed inside your button. You can access global variables, set text fields, change colors, etc.

   ```
   //For example
   private void hwBUTActionPerformed(java.awt.event.ActionEvent evt){
       hwTF.setText(“Hello World”);
   }
   ```

10. Create a new class called HelloWorldApp(see the attached HelloWorldApp.java)
11. Delete the main from your Frame
12. Run your app NOT YOUR FRAME

```java
/**
 * ---------------------------------------------------------------
 * S-h-e-n-e-n-d-e-h-o-w-a--H-i-g-h--S-c-h-o-o-l--T-e-c-h--D-e-p-t
 * ---------------------------------------------------------------
 * FILE:  HWApp.java
 *
 * DATE:  5/20/2015
 *
 * PURPOSE: Demonstrate simple app for starting up frame
 *
 * @author mr Hanley
 * @version 1.0
 * ---------------------------------------------------------------
 *
 * h-a-n-l-e-y.c-o-.-n-r------t-e-a-m-2-0-.-c-o-m-----------------
 */

public class HWApp {
    public static void main(String[] args) {
        HWFrame f = new HWFrame();
        f.setBounds(100,100,600,400);
        //upper left x, upper left y, width, height
        f.setVisible(true);
    }
}
```

```
/**
 * ---------------------------------------------------------------
 * S-h-e-n-e-n-d-e-h-o-w-a--H-i-g-h--S-c-h-o-o-l--T-e-c-h--D-e-p-t
 * ---------------------------------------------------------------
 * FILE:  HWFrame.java
 *
 * DATE:  5/20/2015
 *
 * PURPOSE: Demonstrate GUI Building
 *
 * @author mr Hanley
 * @version 1.0
 * --------------------------------------------------------------
 *
 * h-a-n-l-e-y.c-o-.-n-r------t-e-a-m-2-0-.-c-o-m---------------
 */

public class HWFrame extends javax.swing.JFrame {

    /**
     * Creates new form HWFrame
     */
    public HWFrame() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the
form.
     * WARNING: Do NOT modify this code. The content of this method is
always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

    //THIS CODE IS USUALLY FOLDED -> DO NOT CHANGE THIS STUFF
    }

    private void hwBUTActionPerformed(java.awt.event.ActionEvent evt) {
        //Students -> Add Handling Code Here
        hwTF.setText("Hola Mundo!!");
    }

     //NOTE: I DELETED THE MAIN METHOD THAT NETBEANS GENERATED FOR ME!!!

    // Variables declaration - do not modify
    private javax.swing.JButton hwBUT;
    private javax.swing.JTextField hwTF;
    private javax.swing.JLabel titleLBL;
    // End of variables declaration
}
```