Hier unten ist das einfachste Verfahren, um JOGL in Eclipse einzubinden (**3 kurzen Schritte**) und anschließend ein OpenGL-fähiges Fenster zu erstellen (**5 kurzen Schritte**) – der unten gegebene Code dient als eine Vorgabe/ein Gerüst, die/das anschließend mit weiteren Einzelheiten ergänzt wird:

-3. Unter "Windows -> Preferences -> Java -> Java Build Path -> User Libraries" auf "New" klicken - es erscheint folgender Dialog:

New User Librar	ry ×	
iry name:		
System library (added to the boot class path)		
ОК	Cancel	
	New User Librar ny name: n library (added to the OK	

Geben Sie im Textfeld einen gewünschten Namen für die neue Bibliothek an. Zu empfehlen wäre "JOGL". Dann klicken Sie auf "OK".

Wählen Sie "JOGL" und klicken Sie auf "Add External JARs…" – dann finden Sie den entpackten Ordner mit bereits heruntergeladenen JOGL-Bibliotheken und fügen Sie die Dateien "gluegen-rt.jar" und "jogl.all.jar" aus dem "jogamp-all-platforms\jar\"-Ordner hinzu.

۲	Preferences	- 🗆 🗙
type filter text	User Libraries	← → ⇒ → →
type filter text General Ant Cloud Foundry Code Recommenders Data Management GCMV Gradle GWT Help Install/Update Java Appearance Build Path Classpath Variab User Libraries Code Coverage Code Style Code Style Compiler Debug Decompiler Editor Installed JREs	User Libraries User libraries can be added to a Java Build path and bundle a number of external archives. System libraries will be class path when launched. Defined user libraries	
JUnit V		
?	Apply and Clos	e Cancel

Danach sollte es erweitert so aussehen:

-2. Nun müssen nur noch "Source attachment:" und "Javadoc location:" jeweils angegeben werden. Wählen Sie für "gluegen-rt.jar" als "Source attachment:" die Datei "gluegen-java-src.zip" aus dem "jogamp-allplatforms"-Ordner und für "jogl.all.jar" als "Source attachment:" die Datei "jogl-java-src.zip" welche sich ebenfalls in "jogamp-all-platforms" befindet:

	Source Attachment Configuration	- 🗆 🗙					
Select the location (folder, JAR or zip) containing the source for 'gluegen-rt.jar': O Wor <u>k</u> space location							
<u>P</u> ath:		<u>B</u> rowse					
• External loca	External location						
Pa <u>t</u> h:	C:/EclipsePacks/jogamp-all-platforms/gluegen-java-src.zip	External <u>F</u> ile					
Encoding:	Default (Cp1252) v	External F <u>o</u> lder					
?	ОК	Cancel					

Wählen sie nun noch (für beide Einträge) die entsprechende "Javadoc location":

۲	Javadoc For 'jogl-all.jar'	-		×	
Javadoc URL (e.g. 'http://www.sample-url.org/doc/' or 'file:///c:/myworkspace/myproject/doc')					
Javadoc location path:	https://jogamp.org/deployment/jogamp-next/javadoc/jogl/javadoc/	<u>B</u> ro	wse		
		<u>V</u> al	idate.		
$\bigcirc$ Javadoc in ar <u>c</u> hive					
	External file     Workspace file				
Archive <u>p</u> ath:		Bro	wse		
Path <u>w</u> ithin archive:		Brg	wse		
		<u>V</u> al	idate.		
?	ОК	C	ancel		

URLs bekommt ihr unter dem "Builds / Downloads"-Menüpunkt <u>auf der offiziellen Webseite</u> von JOGL (kopiert einfach die Links unter "API Docs" jeweils von GlueGen und JOGL). Anschließend "OK" klicken. **1.** Nachdem ihr ein neues Java-Projekt (mit JOGL-Unterstützung) erstellt, wird euer Projekt wie folgt aussehen:



2. Leert ihr den Inhalt von der "Main" Klasse und fügt den folgenden Code ein:

```
package cg;
import javax.swing.JFrame;
//import com.jogamp.opengl.GL2;
import com.jogamp.opengl.GLAutoDrawable;
import com.jogamp.opengl.GLCapabilities;
import com.jogamp.opengl.GLEventListener;
import com.jogamp.opengl.GLProfile;
import com.jogamp.opengl.awt.GLCanvas;
public class Main implements GLEventListener {
       @Override
       public void dispose(GLAutoDrawable arg0) {
          //body
       }
   @Override
  public void reshape(GLAutoDrawable arg0, int arg1, int arg2, int arg3, int arg4) {
          //body
   }
  @Override
  public void init(GLAutoDrawable arg0) {
      //body
   }
  @Override
  public void display(GLAutoDrawable arg0) {
        //body
   }
  public static void main(String[] args) {
         //instantiating OpenGL functionality
      final GLProfile profile = GLProfile.get(GLProfile.GL2);
      GLCapabilities capabilities = new GLCapabilities(profile);
      //creating frame
      final GLCanvas glcanvas = new GLCanvas(capabilities);
      //setting frame size inputs
      final int width = 800;
      final int height = 600;
      Main m = new Main();
      glcanvas.addGLEventListener(m);
      //creating frame
      final JFrame frame = new JFrame ("OpenGL Fenster");
      //adding canvas to frame
      frame.getContentPane().add(glcanvas);
      //setting visibility and sizes to frame:
      frame.setSize(width, height);
      frame.setVisible(true);
  }
}
```

**3.** Drückt ihr auf dem nach unten zeigenden Pfeil rechts von der "Run"-Taste (die wie ein nach rechts gerichteter weißer Pfeil in einem hellgrünen Kugel aussieht) und wählt "Run Configurations..." aus:



**4.** Stellt sicher, dass das aufgepoppte Fenster so (wie unten) aussieht und drückt anschließend auf "Run":

Run Configurations		×
Create, manage, and run cont Run a Java application	figurations	
Image: Second system         Image: Second system <th>Name: Main          Main</th> <th>Common Browse Search</th>	Name: Main          Main	Common Browse Search
Filter matched 6 of 6 items	Apply	Revert
?	Run	Close

5. Das schwarze Fenster mit dem Titel "OpenGL Fenster" wird erstellt und durchs Drücken auf "X" geschlossen:

