```
esalina@biolinux:~/phylostuff$ cat phymlanimate.pl
#!/usr/bin/perl

#phyml tree animator!

#Variables
my $infile="/home/esalina/phylostuff/flu2010.phylip_phyml_trace_nolk.txt";
my $outfileBase="tree";
my $tempLine;
my $cmdLine;
my $tempFilePath="/tmp/eddietmp.txt";
my $jpegnum=0;
my $total=`wc -l $infile`;



#open the file with the trace
open(INFILE,"<$infile") or die "ERROR - COULDN'T OPEN $infile !\n";
while(<INFILE>)
        {
        #get a tree
        $tempLine=$_;
        chomp($tempLine);

        #write the tree to a temp file
        open(TEMPFILE,">$tempFilePath");
        print TEMPFILE $tempLine;
        close(TEMPFILE);

        #Issue a command to write a tree to a JPEG file
        $cmdLine="java PhyloCanvas $tempFilePath ".$outfileBase."0000".$jpegnum.".jpeg";
        `$cmdLine`;

        #increment the number of trees processed
        #and give a status/progress message
        $jpegnum++;
        print "$jpegnum of $total trees processed!\n";
        }
close(INFILE);
print "Program ending normally!\n";
```

```
esalina@biolinux:~/phylostuff$ cat Animator.java
import java.io.*;
import javax.imageio.*;
import java.awt.image.*;
import java.util.*;
import java.util.regex.*;


/**
EXAMPLE OF HOW TO USE THE GIF ANIMATOR!
 * <pre>
 *  Example:
 *     AnimatedGifEncoder e = new AnimatedGifEncoder();
 *     e.start(outputFileName);
 *     e.setDelay(1000);   // 1 frame per sec
 *     e.addFrame(image1);
 *     e.addFrame(image2);
 *     e.finish();
 * </pre>
*/

public class Animator
       {

       //to extract JUST .jpeg files from a direcory listing
       public static class JPEGFilter implements FilenameFilter
              {

              public boolean accept(File dir,String name)
                     {
                     return name.endsWith(".jpeg");
                     }


              }

       //to sort jpeg files (useful becuase 0004.jpeg would come after 00039.jpeg!
       public static class JPEGFileNameSorter implements Comparator
              {

              //get the number from the file name
              private static int getNumFromObject(Object o)
                     {
                     String s=new String(o.toString());
                     Pattern p=Pattern.compile("(\\d+)\\.jpeg",Pattern.CASE_INSENSITIVE);
                     Matcher m=p.matcher(s);
                     if(m.find())
                            {
                            String numString=new String(m.group(1));
                            int x=Integer.parseInt(numString);
                            return  x;
                            }
                     else
                            {
                            throw new RuntimeException("Error!, can't get number from '"+o.toString()+"'");
                            }
                     }
```

```
/*
Returns: a negative integer, zero, or a positive integer
 as the first argument is less than
, equal to, or greater than the second.*/
public int compare(Object o1, Object o2)
	{
	int i1=getNumFromObject(o1);
	int i2=getNumFromObject(o2);

	if(i1<i2)
		{
		return (-1);
		}
	else if(i1==i2)
		{
		return 0;
		}
	else
		{
		return 1;
		}



	}

//to fulfill contract of comparator
public boolean equals(Object o1,Object o2)
	{
	if(compare(o1,o2)==0)
		{
		return true;
		}
	else
		{
		return false;
		}
	}


}
```

```
	public static void main(String[] args)
		{
		String outputName=new String("mytrees.gif");
		int numWritten=0;

		try
			{
			//prepare for animation
			AnimatedGifEncoder e = new AnimatedGifEncoder();
			e.start(outputName);
			e.setDelay(1000);

			//get the jpeg files in the passed argument/directory and SORT the files!
			File dir=new File(args[0]);
			File[] listing=dir.listFiles(new JPEGFilter());
			Arrays.sort(listing,new JPEGFileNameSorter());


			//scan for jpeg files
			for(int f=0;f<listing.length;f++)
				{
				if(listing[f].getCanonicalPath().endsWith(".jpeg"))
					{
					System.out.println("Found "+listing[f].getCanonicalPath());
					//extract the image from the jpeg file
					BufferedImage bi=ImageIO.read(listing[f]);
					//add the image to the animated gif
					e.addFrame(bi);
					//increment number written
					numWritten++;
					}
				}
			//finalize the gif!
			e.finish();
			}
		catch(Exception e)
			{
			e.printStackTrace();
			}

		//end with a status message
		System.out.println("File "+outputName+" written with "+numWritten+" trees!");
		}

	}
esalina@biolinux:~/phylostuff$
```