

COMPUTER SCIENCE – UNIT 2: RECURSION - LANZ SIDNEY POVEY

Boom boom acka-lacka lacka boom

Boom boom acka-lacka boom boom

It was a night like this forty million years ago

I lit an assignment, picked up a monkey Java-based application to go

The sun was spitting fire, the sky was blue as Netbeans' loading header

I felt a little tired, so I watched compiled the project for future debugging and error-catching purposes.

And walked the dinosaur, I walked the dinosaur

Hooked? You better be, because it took me a whopping 10 seconds to make those lyrics. See, you just made me waste 10 seconds of my life, now 15, 16, 17, 18.... You owe me now, you owe me a coding project... a dinosaur coding project.

"But person-who's-name-has-not-been-stated!" you ask. *"How can you have a futuristic project about something prehistoric?"*. Well first my name is Louis. Second of all, directories! Let's be real, *Windows Explorer* sucks at loading, it's so slow!

Understandably so though, as it's loading all these files. I have an alternative to this, **DinosaurDirectories™**. Like a prehistoric computer, instead of loading the actual files, it only loads their names, meaning that you can load a quick preview of your files! This is handy if you just want to search a certain folder for a specific file name that you forgot what was called without needing to open it. Thaaattttss

DinosaurDirectories™! That's also our slogan.

Requirements

- Have a way for the user to input the file directory they would like to search.
- Use the System Output to save memory resources (because we want this to load as quickly as possible)
- Use indents to show how deep the file is in a folder.

```

/** required package namespace */
package trexdrex;

/** required imports */
import java.io.File;           // for the directoryDisplay(File dir) method
import java.io.IOException;    // a Checked Exception, so it should either be handled
                               // or declared to be thrown

public class TRexDrex
{
    final static String LINE_BREAK    = "\n";    // moves text to a next line
    final static String LINE_BREAK_X2 = "\n\n";  // creates a blank line

    /**
     * The main method of the application
     * @param args don't worry, this one doesn't involve command line arguments
     */
    public static void main(String[] args)
    {
        s.outputc();
        String directoryInput = s.input( "Boom boom acka-lacka lacka boom" + LINE_BREAK
            + "Boom boom acka-lacka boom boom + " + LINE_BREAK_X2

            + "OGH UGGGHH UGH UGH OOOOGHHHHH!" + LINE_BREAK
            + "Translation: Greetings my valued patron, welcome to TRexDrex!"
            + LINE_BREAK_X2

            + "UG UG UG OGGH OGGHH UGGGHH UGGGHHH" + LINE_BREAK
            + "Translation: Quickly scower for file names via specifying a folder"
            + " with use of the input box below" + LINE_BREAK_X2

            + "OGH OGH OGH UGH UGH OGGGGHH UGGGHHH OGH UGGGHHHH" + LINE_BREAK
            + "Translation: Due to various internal system technical limitation"
            + " please assure yourself that you've " + LINE_BREAK
            + "replaced every '\\' in your inputted file path with a \\ "
            + LINE_BREAK_X2

            + "UGH UGH OGGGHHH OGGGHHHHH UGGGGGGGGGGGGGGGGGGGGHHHHHHHHH" + LINE_BREAK
            + "Translation: Copyright 40 000 000 years ago." + LINE_BREAK_X2);

        File currentDir = new File(directoryInput); // current directory
        TRexDrex.directoryDisplay(currentDir);
        // class reference for using a non-static method from within the static...
        // ... main method as an object.
    }

    /**
     * A method that recursively searches a user-specified directory for every
     * file that exists within it, displaying each of the file's (plus the folder
     * itself that they're held in) directory via a system output.
     * @param dir the user-specified directory
     */
    public static void directoryDisplay(File dir)
    {
        try

```

```
{
    File[] files = dir.listFiles();
    for (File file : files)
    {
        if (file.isDirectory())
        {
            s.sout("directory:" + file.getCanonicalPath());
            directoryDisplay(file);
            // canonical path = shortest absolute path
            // absolute path = full file directory
        }
        else s.sout("      file:" + file.getCanonicalPath());
        // the space in front of the file adds an indent to organize

    }
}

// error catching in case the file directory is not valid
// IO = Input/Output
catch (IOException e)
{
    s.output("UGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGHHHHHHHHHHH" + LINE_BREAK
            + "Translation: Error. Invalid file directory" + LINE_BREAK_X2);
    e.printStackTrace();
    // prints throwable stack trace and its backtrace to the standard error stream
}
}
```

```

/** package class namespace */
package trexdrex;

/** required imports */
import javax.swing.JOptionPane; // for the JOptionPane by itself
import javax.swing.UIManager;   // for the colored JOptionPane's look
import java.awt.Color;         // for the colored JOptionPane's colors

/**
 * Project      : The s stands for supercalifragilisticexpialidocious
 * Description  : Literally just my custom default template lol.
 * Author       : Lanz Povey
 * Date        : Mar 19th 2018 (updated)
 * Instructor   : Mr. Wachs
 */
public class s
{
    // R.I.P s.sout coloured dialog zombie code
    public static final int MAX_INT = 2147483647;

    //=====Code Shorteners=====
    /**
     * A shortened JOptionPane.showMessageDialog, as to help condense the code.
     * As it is the only type of possible output, it's generic name suits it well.
     * @param text the text to be displayed from the JOptionPane.showMessageDialog.
     */
    public static void output(String text)
    {
        JOptionPane.showMessageDialog(null, text);
    }

    /**
     * A method that sets the color properties of all preceeding JOptionPane's
     * (border, background, foreground)
     */
    public static void outputc()
    {
        UIManager UI = new UIManager();
        UI.put("OptionPane.background", Color.lightGray);
        UI.put("Panel.background", Color.gray);
        UI.put("OptionPane.messageForeground", Color.orange);
    }

    /**
     * A shortened JOptionPane.showInputDialog, as to help condense the code.
     * @param text the text to be displayed from the JOptionPane.showInputDialog.
     * @return the user's input into the JOptionPane.showInputDialog.
     */
    public static String input(String text)
    {

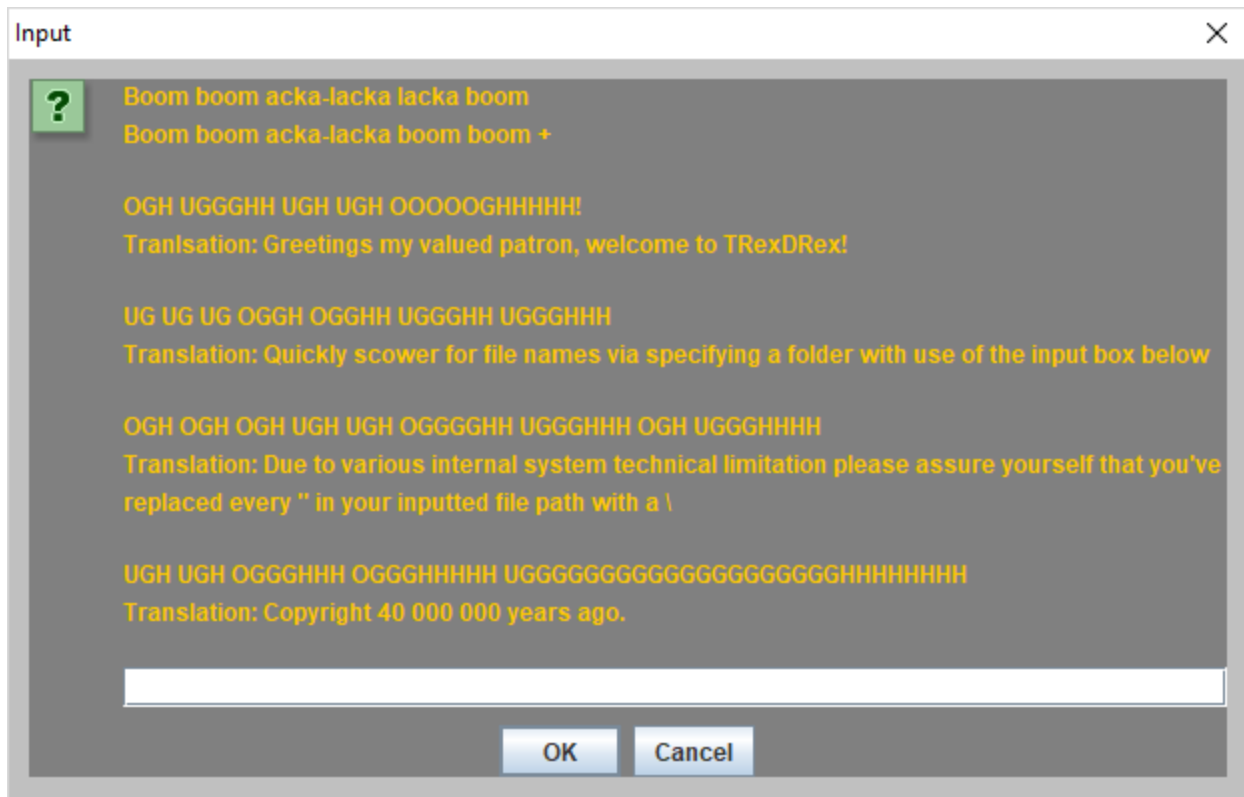
```

```

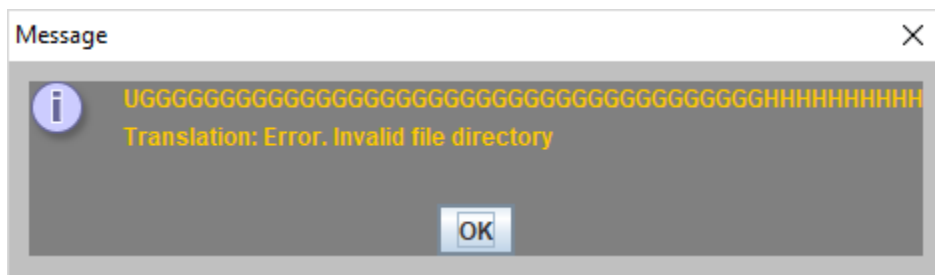
    String textInput = JOptionPane.showInputDialog(null, text);
    return textInput;
}

/**
 * So here's the back story. You know how you can shorten sout with some
 * bizarre key command right? Well I ain't got a clue how I can do that.
 * So we're doing this instead. Plus, it makes the code shortened and simple
 * #ScratchIsTheOneTrueCodingLanguage
 * @param text the text to be displayed from the system output
 */
public static void sout(String text)
{
    System.out.println(text);
    //Colors won't work if the text begins with a "\n"
}
}

```



If the user enters an in a file directory that does not exist.



User Input = D:\\TRexDRex

Output - TRexDRex (run) ×

```
run:
directory:D:\TRexDRex\nbproject
    file:D:\TRexDRex\nbproject\project.xml
directory:D:\TRexDRex\nbproject\private
    file:D:\TRexDRex\nbproject\private\private.properties
    file:D:\TRexDRex\nbproject\private\private.xml
    file:D:\TRexDRex\nbproject\project.properties
    file:D:\TRexDRex\nbproject\genfiles.properties
    file:D:\TRexDRex\nbproject\build-impl.xml
    file:D:\TRexDRex\build.xml
directory:D:\TRexDRex\src
directory:D:\TRexDRex\src\trexdrrex
    file:D:\TRexDRex\src\trexdrrex\TRexDRex.java
    file:D:\TRexDRex\src\trexdrrex\s.java
    file:D:\TRexDRex\manifest.mf
directory:D:\TRexDRex\build
directory:D:\TRexDRex\build\classes
directory:D:\TRexDRex\build\classes\trexdrrex
    file:D:\TRexDRex\build\classes\trexdrrex\s.class
    file:D:\TRexDRex\build\classes\trexdrrex\TRexDRex.class
    file:D:\TRexDRex\build\classes\.netbeans_automatic_build
    file:D:\TRexDRex\build\classes\.netbeans_update_resources
directory:D:\TRexDRex\test
    file:D:\TRexDRex\TRex DRex.pdf
    file:D:\TRexDRex\Code.pdf
    file:D:\TRexDRex\Program.docx
BUILD SUCCESSFUL (total time: 22 seconds)
```