

Clayton Turner
CSIS 603

Visitor Pattern Program

PrintDirectoryStructureWithVisitor.java

package com.turner

```
import java.io.File;
import java.nio.file.*;
import java.nio.file.attribute.BasicFileAttributes;
import java.io.IOException;
import static java.nio.file.FileVisitResult.*;

public class PrintDirectoryStructureWithVisitor extends SimpleFileVisitor<Path>
{
    /* Used for keeping track of nesting level for prints */
    private static int nestingLevel = 0;
    /**
     * Prints the structure for a given directory or file
     */
    public static void main(String[] args) throws IOException
    {
        if (args.length != 1)
        {
            printUsage();
            System.exit(-1);
        }

        String pathName = args[0];
        Path startingDir = Paths.get(pathName);

        PrintDirectoryStructureWithVisitor pdsv = new PrintDirectoryStructureWithVisitor();
        Files.walkFileTree(startingDir, pdsv);
    }

    /* Handles recursion logic and prints files, not directories */
    @Override
    public FileVisitResult visitFile(Path file, BasicFileAttributes bfAttrs)
    {
        File f = new File(file.toString());
```

```

    if (f.isFile()){
        printFile(f);
    }
    else{
        nestingLevel++;
        for(File recF : f.listFiles()){
            Path nextFile = Paths.get(recF.toString());
            visitFile(nextFile, bfAttrs);
        }
    }
    return CONTINUE;
}

/* Before visiting a directory, print it out and
   update the nesting level for subfiles/subdirectories
   */
@Override
public FileVisitResult preVisitDirectory(Path dir, BasicFileAttributes bfAttrs)
{
    printDirectory(dir.toFile());
    nestingLevel++;
    return CONTINUE;
}

/* We're done with this directory, so let's back up with our nesting level */
@Override
public FileVisitResult postVisitDirectory(Path dir, IOException ex){
    nestingLevel--;
    return CONTINUE;
}

/* Error handling */
@Override
public FileVisitResult visitFileFailed(Path file, IOException ex)
{
    System.err.println(ex);
    return CONTINUE;
}

```

```

/* Usage for command line */
private static void printUsage()
{
    System.out.println("Usage: java com.turner.PrintDirectoryStructureWithVisitor(<path>);
    System.out.println("    where <path> is the path of a file or directory");
    System.out.println();
}

private static String getIndentString(int nestingLevel)
{
    StringBuilder s = new StringBuilder();

    for (int i = 0; i < nestingLevel; i++)
        s.append(" ");

    return s.toString();
}

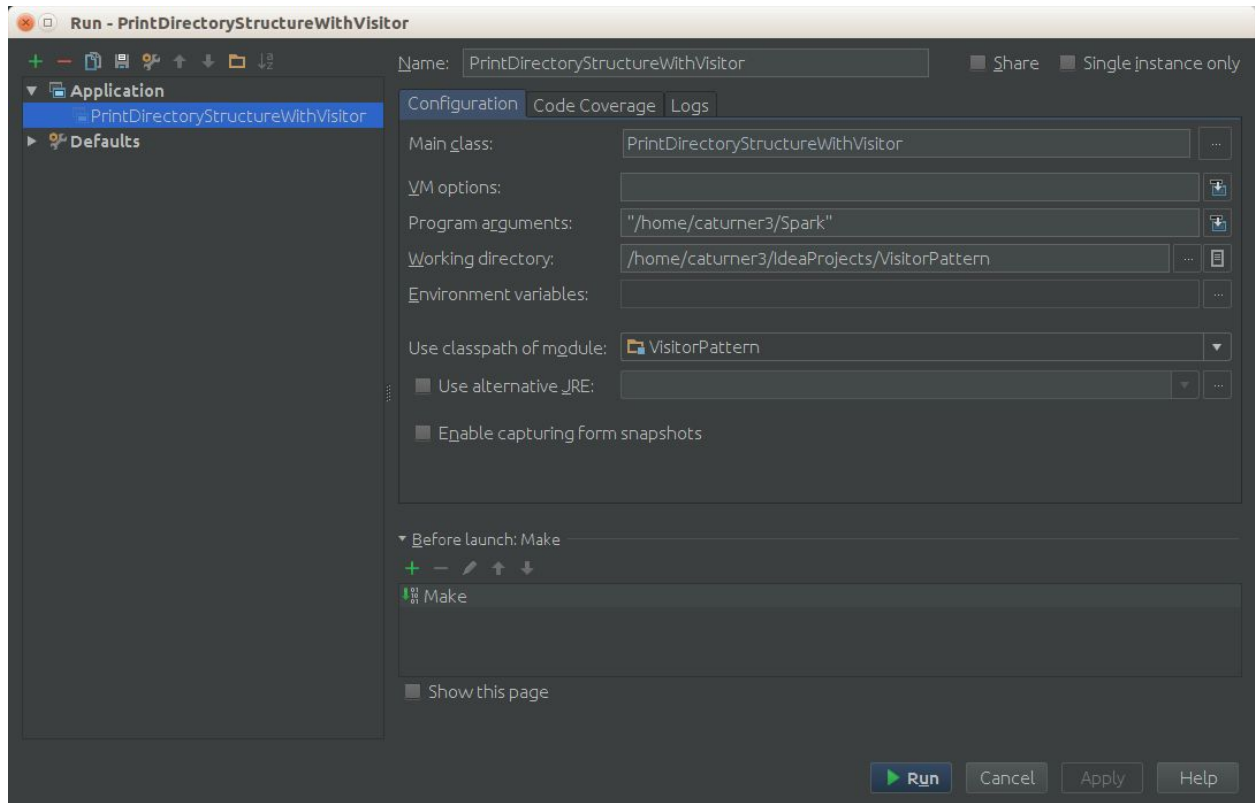
/* Abstracted printer */
private static String clean_printer(File dirOrFile){
    String[] sList = dirOrFile.toString().split("/");
    return sList[sList.length-1];
}

private static void printDirectory(File dir)
{
    System.out.println(getIndentString(nestingLevel)+" "+clean_printer(dir));
}

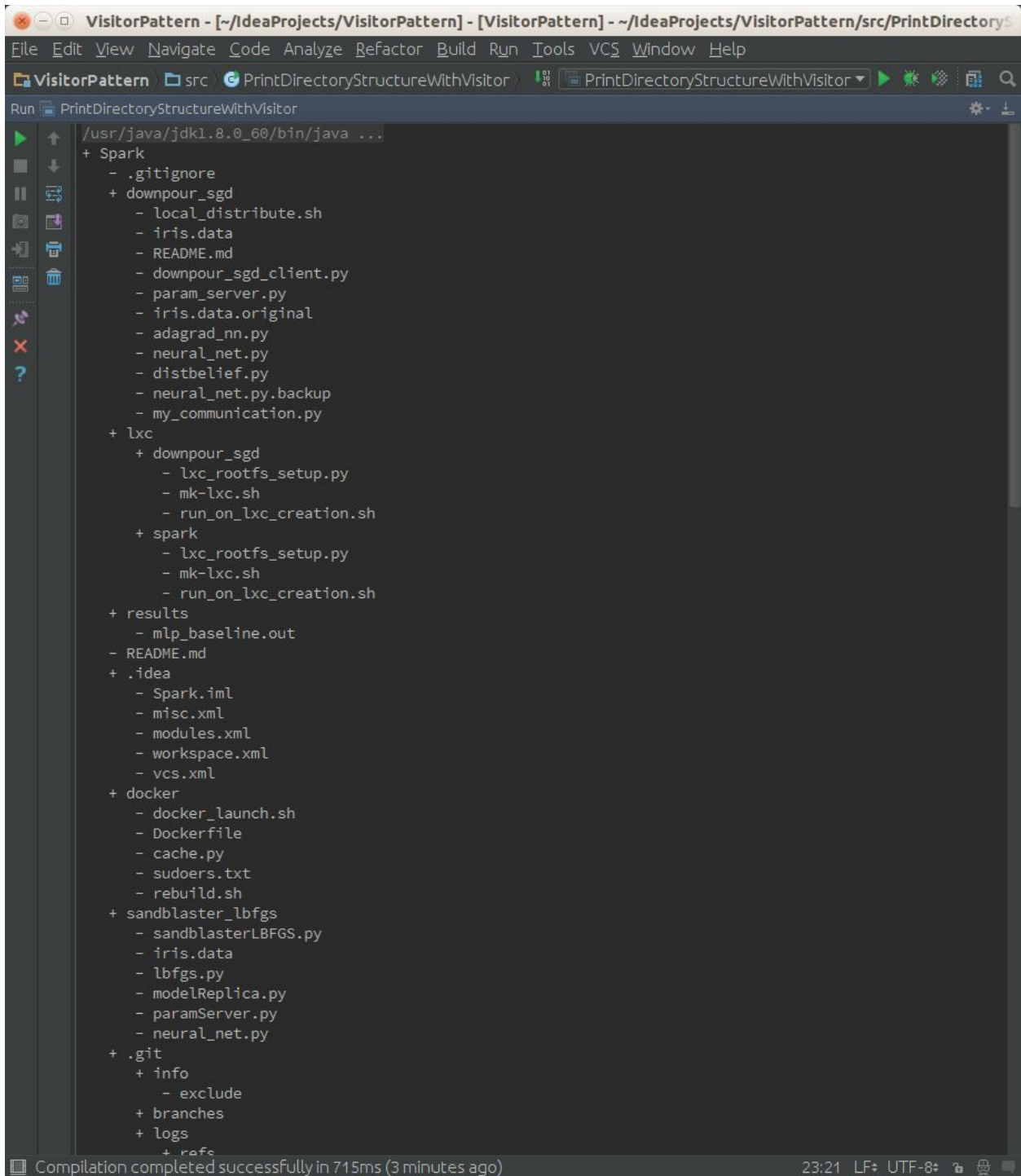
private static void printFile(File file)
{
    System.out.println(getIndentString(nestingLevel)+"- "+clean_printer(file));
}
}

```

IntelliJ Argument Screenshot



Test Screenshots



VisitorPattern - [~/IdeaProjects/VisitorPattern] - [VisitorPattern] - ~/IdeaProjects/VisitorPattern/src/PrintDirectoryS

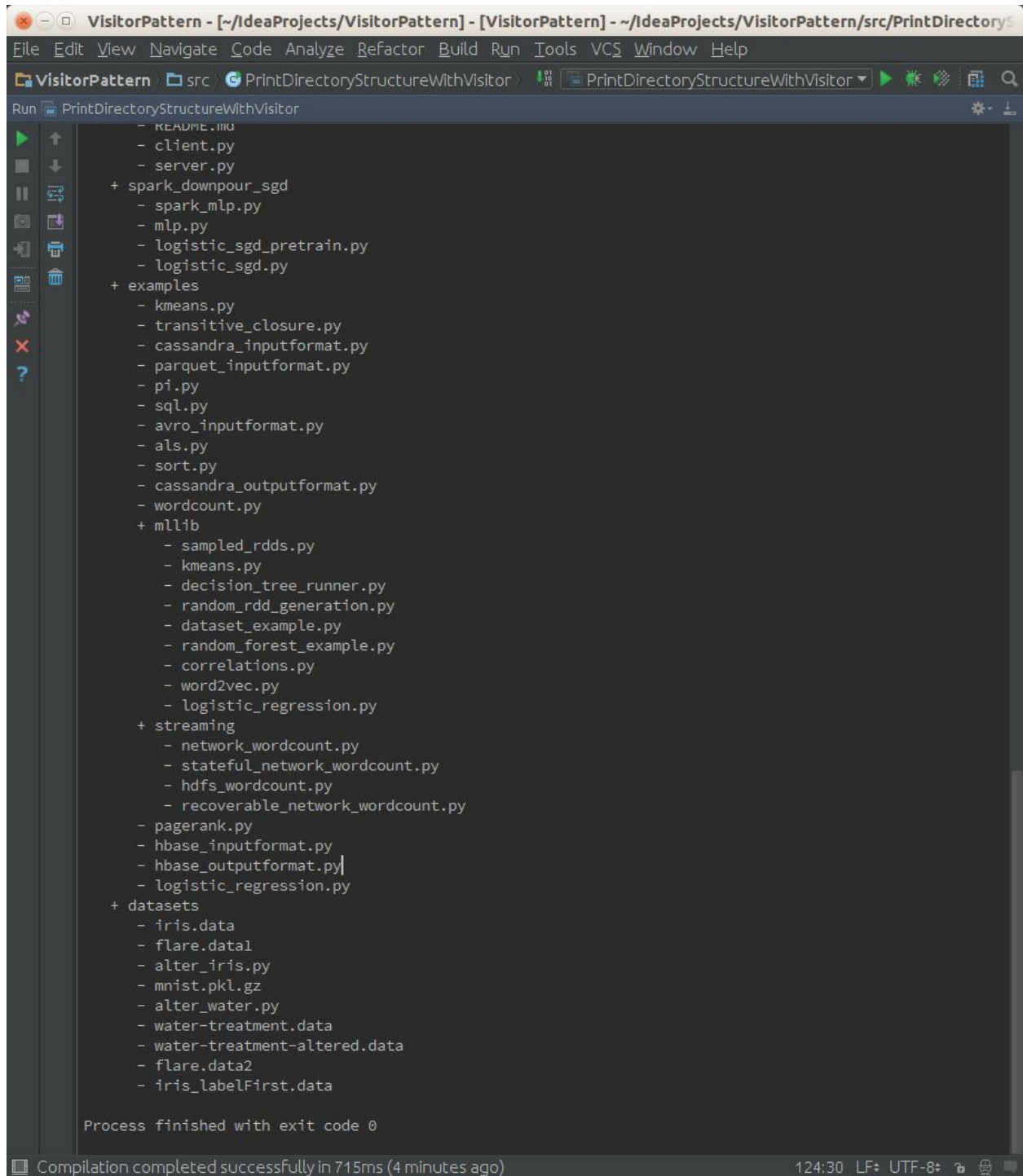
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

VisitorPattern \ src \ PrintDirectoryStructureWithVisitor \ PrintDirectoryStructureWithVisitor

Run PrintDirectoryStructureWithVisitor

```
- neural_net.py
+ .git
+ info
  - exclude
+ branches
+ logs
  + refs
    + remotes
      + origin
        - HEAD
    + heads
      - master
    - HEAD
  - config
+ hooks
  - pre-rebase.sample
  - prepare-commit-msg.sample
  - pre-applypatch.sample
  - applypatch-msg.sample
  - pre-push.sample
  - post-update.sample
  - pre-commit.sample
  - update.sample
  - commit-msg.sample
- description
+ refs
  + remotes
    + origin
      - HEAD
  + heads
    - master
  + tags
- packed-refs
- HEAD
- index
+ objects
  + info
  + pack
    - pack-66ae4ac126da81330b0e6e2be3fcd1c08278f414.pack
    - pack-66ae4ac126da81330b0e6e2be3fcd1c08278f414.idx
+ old
  - README.md
  - client.py
  - server.py
+ spark_downpour_sgd
  - spark_mlp.py
  - mlp.py
  - logistic_sgd_pretrain.py
  - logistic_sgd.py
+ examples
  - kmeans.py
  - transitive_closure.py
```

Compilation completed successfully in 715ms (4 minutes ago) 23:21 LF+ UTF-8+



GUI Screenshot Subset

