R·I·T

Rochester Institute of Technology
Golisano College of Computing and Information Sciences
Department of Information Sciences & Technology

Find Duplicate files

ISTE-121 – Homework 01

Problem this application will solve

Duplicate files on a computer system consuming extra disk space are a fact of working with a computer. Some of these files may be from backups, some could be from downloading the same file multiple times, or by several other reasons. Simply doing a find/search from the directory of interest and all its subdirectories could detect some of these files by resorting the list by filename. However, this does not tell if these files are really the same or only have the same name. It also doesn't reveal duplicate files with different names. Looking at the size would also help locate duplicates, but that would only bring us up to maybe 90% assured of duplicate. Comparing two files using a CRC (Cyclical Redundancy Check) ensures these files are the same, or as close to being 100% assured as possible. With CRC, error is 1 out of 4,294,967,295 files (4.3 billion) two files have the same CRC without being the same. Comparing the CRC *and* file length makes it almost 100%.

Experience you will develop by doing this assignment

- Use of these Packages
  - java.io
  - java.util
  - java.text
- Concepts developing
  - Report breaks
  - Formatting for printing reports in columns (padding)
  - Using other people's examples to solve problems
  - Maintenance programming
  - Recursion

- Use of these classes:
  - File
  - Comparator – to sort information
  - Collections
  - CRC32
  - BufferedInputStream
  - FileInputStream
  - BufferedWriter
  - FileWriter
  - Long (method toHexString())
  - SimpleDateFormat
  - Date (timer)
  - DecimalFormat (optional)
  - StringBuffer (optional)
  - Hash/table/map/set (optional)

Assistance for this homework

- InClass exercises this week.
- Download the CRC sample code. You may use the getCrc() method provided.
- Read this complete handout, making note of classes and functionality, sketch UML from these before starting. This will be your roadmap.
- Further questions, ask your instructor first, and then others.

The reason for the written length of this assignment isn't the complexity; it is the descriptions to help you complete it as quickly as you can.

How to find Duplicate files

Using the InClass exercises and code provided in class as a starting point, this is actually a lot easier than the previous page may make this appear.

Processing outline (all functionality must be addressed)

- Determine the starting directory, call a method passing it the starting directory.
- Get a list of the current directory's entries.
- Step through this list, when you come across a:
  - Directory – have this method call itself (recursion) with the new starting directory. Make sure you pass the full path name as an object, not string.
  - File – Add an object to the overall collection, which contains the file object along with the calculated CRC. (May be easier to calculate CRC now, than later.)
  - Hidden file, or File Doesn't exist, or other - Report file and message to the System.out.println, then ignore it.
- Print the number of files processed, see sample output
- Sort the collection by CRC value (it is a **long** at this point)
- Find just the duplicate entries, and print them. Separating each group with a blank line. See partial sample "FindDups.txt" output.
- At each major phase, print something to show your program is progressing.

Class functionality suggested:

- Classes suggested – not required:
  - TestDups.java – main() controls flow of processing
    - When no command line starting directory is specified, use "C:\Public\" as default
    - Calls the "find dups" method
    - Does ALL printing to the screen
  - FindDups.java
    - Recursively gathers files creating Info objects
    - Adds Info objects to a collection (ArrayList, TreeSet, etc.)
    - Contains method to calculate CRC for the current file object
    - Recursive method that returns a collection of Info (to TestDups)
    - Prints hidden files are ignored
  - Info.java – may be replaced by using hashtables, or other data structures
    - Simple class to hold information such as File object, and CRC value
    - Accessors for attributes
  - CrcComparator.java – Extends Comparator, used in the sort process
    - Contains `compare` method
- Output file is "FindDups.txt" – Output in fixed width columns as shown
    - CRC
    - File Length
    - Last Modified date
    - Fully qualified filename

Non-Requirements

- However, these can be extra credit, depending on amount of functionality.
    - GUI
    - Multiple sort outputs possible, choice specified on command line
    - Option to show the top N largest duplicate files (eg: top 10 biggest dups)

Requirements/Hints:

- TestDups and FindDups MUST be in separate Java files. Other classes can be in their separate files or within FindDups.
- Make sure the starting filename for the recursive method ends with a "\".
    - Check for ending "\" on the command line input.
    - Supply the "\" for the default starting directory C:\Public\
    - Remember rules about the backslash character in double quotes.

Items for dropbox or paper in class

- All Java source files.
- All Class files.
- The program's output file "FindDups.txt".

```
C:\> java TestDups C:\Public\

Find Duplicate files - by Michael Floeser
Find Duplicates.  Processing: C:\Public\
C:\Public\Thumbs.db   is Hidden
Files gathered = 85
Writing output file: FindDups.txt
Duplicate files = 13
Duplicate groups= 4
Processing took: 0.3100 seconds
C:\>
```

Sample: FindDups.txt (partial file output - shrunk to fit on screen)

```
        0          0   12/03/03 02:42:22 PM  C:\Public\Edit9.java
        0          0   12/03/03 03:31:28 PM  C:\Public\Junk.java.bak

  db90573         59   01/08/99 04:18:22 PM  C:\Public\bcwdef.mbt
  db90573         59   01/08/99 04:18:22 PM  C:\Public\bcwdef.mrt

 14905204         71  11/14/03 01:27:18 PM  C:\Public\Default User\pq4mdx39.slt\Cache\This is one of 7 files.File6
 14905204         71  11/14/03 01:27:18 PM  C:\Public\Default User\pq4mdx39.slt\chrome\This is one of 7 files.File4
 14905204         71  11/14/03 01:27:18 PM  C:\Public\Default User\pq4mdx39.slt\Cache.trash\This is one of 7 files.File5
 14905204         71  11/14/03 01:27:18 PM  C:\Public\Default User\pq4mdx39.slt\This is one of 7 files.File3
 14905204         71  11/14/03 01:27:18 PM  C:\Public\Default User\This is one of 7 files.File2
 14905204         71  11/14/03 01:27:18 PM  C:\Public\PNGs\This is one of 7 files.File7
 14905204         71  11/14/03 01:27:18 PM  C:\Public\This is one of 7 files.File1

 c71c0011      4,096  08/30/02 01:53:34 PM  C:\Public\Default User\pq4mdx39.slt\Cache\_CACHE_002_
 c71c0011      4,096  08/30/02 01:53:34 PM  C:\Public\Default User\pq4mdx39.slt\Cache\_CACHE_003_
```

Your output does not required to have the CRC printed, as that is meaningless for the user. But it is good to print while writing / debugging your program to verify it is correct. Suggest creating a small subdirectory, with subdirectories of known duplicate and non-duplicate files so you know your program is working. For a valid test case, create a sets of 2, 3 and 4 duplicate files, suggest naming the sets with the 2, 3, 4 in the name.

Grade sheet for Homework #8 – Find Duplicate Files

Item (Functionality required, class names are not required)	**Max Points Allowed**	**Points Earned**
TestDups.java (contains main()) • Determines the starting directory from command line or default (C:\Public\) location. • Calls the FindDups constructor / methods that finds all files under specified directory • Print running & status summary information to console	5 5 5	
FindDups.java (separate file from TestDups.java) • Has recursive method for processing sub/directories • Calculates CRC	5 5	
Info.java (class may be eliminated by using hashtables still full credit) • Attributes (minimum list): File object, CRC • Methods: Accessors for all attributes	10	
CrcComparator.java (can be separate class, or in FindDups) • Compare method	10	
Overall: • Output has a sort order (file length, file name, date) • Print only duplicates, with blank line between groups	5 10	
FindDups.txt – Output is formatted in columns	5	
Javadocs in all files, and for all methods	10	
Follow coding standards **Making this appear as your best Java work ever!**	5	
Overall *execution and outputs* as expected	15	
Total:	**100**	
Extra credit for: (must be described in TestDups header comments <u>and</u> in a file TestDups.txt to receive credit) • GUI (Command line output still required) ○ 5 = Shows command line/default start directory Button to RUN, output sent to file. No other interaction or GUI output. ○ 20= Starting with other options, with display and save capabilities, counts also displayed. • Printing file last modified date/time • File filtering option for comparison (include or exclude) • Other sorts or features, as options • Other options on command line • Just about anything else (depends on degree of work) • Command line option to change output file • UML diagram	5 to 20 max 3 5 1 to 10 3 to 20 ?___? 4 5	
Total	???	

Comments: