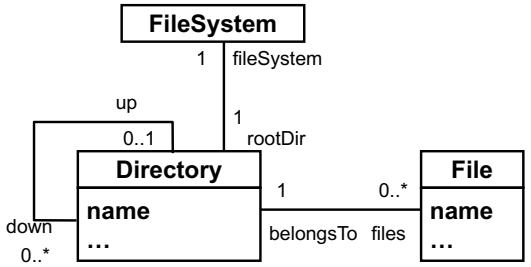
Composite Design Pattern

Composite Design Pattern

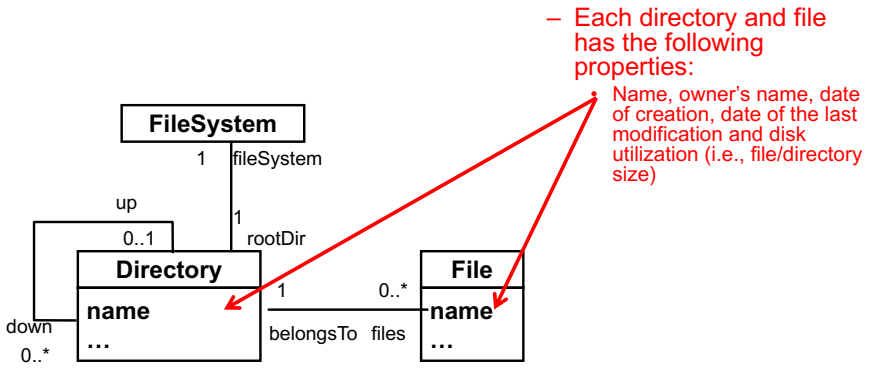
- Intent
  - Compose objects into a tree structure to represent a part-whole hierarchy.
  - Allow clients (of a tree) to treat individual objects and compositions of objects uniformly.

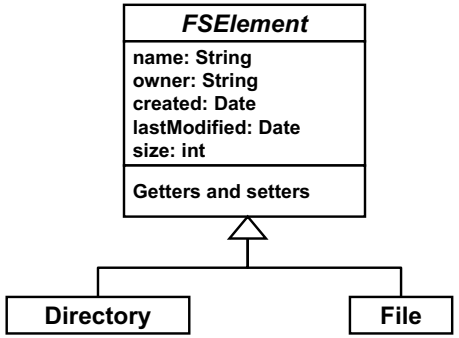
A Design Exercise: File System

- A file system consists of directories and files.
- Each file exists in a particular directory.
- Each directory can contain multiple files.
- Directories form a tree structure.
  - Every directory has its parent directory, except the root directory.
  - Each directory can have multiple sub directories.
- Each directory and file has the following properties:
  - Name, owner's name, date of creation, date of the last revision and disk utilization (i.e., file/directory size)

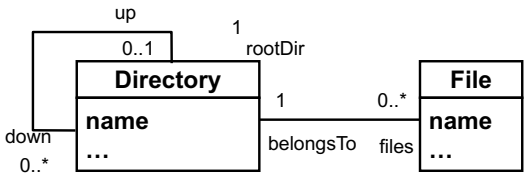


- A file system consists of directories and files.
- Each file exists in a particular directory.
- Each directory can contain multiple files.
- Directories form a tree structure.
  - Every directory has its parent directory, except the root directory.
  - Each directory can have multiple sub directories.
- Each directory and file has the following properties:
  - Name, owner's name, date of creation, date of the last revision and disk utilization (i.e., file/directory size)

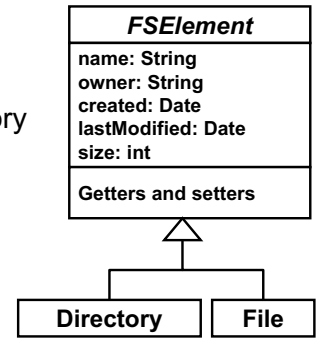




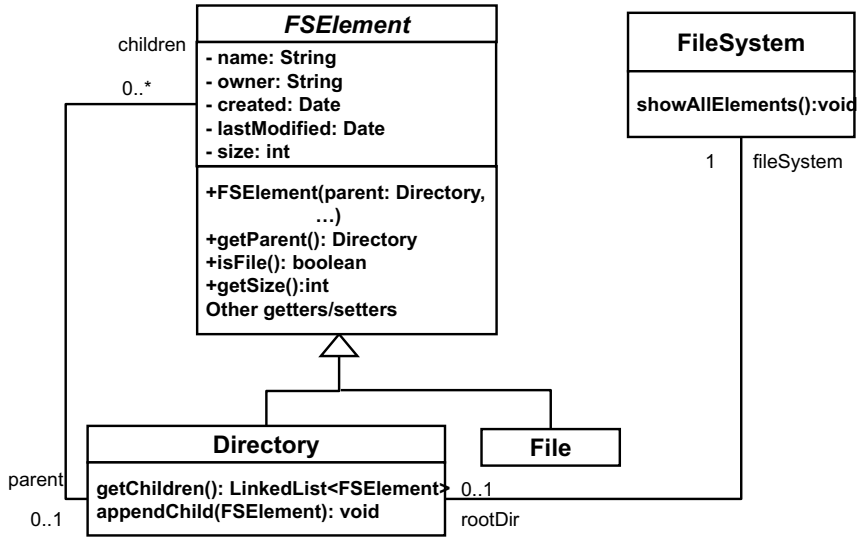
- A directory is never transformed to be a file.
- A file is never transformed to be a directory.



- How can we design directory-to-directory structures?
- How can we design file-to-directory structures?

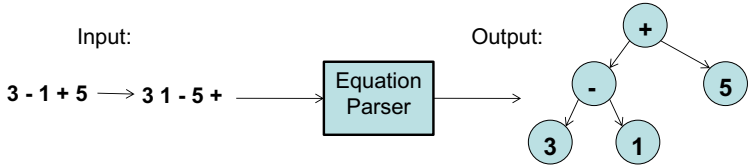


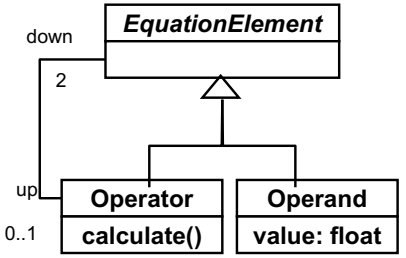
Using *Composite*…

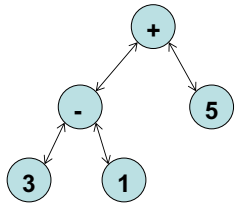


Another Example of *Composite*

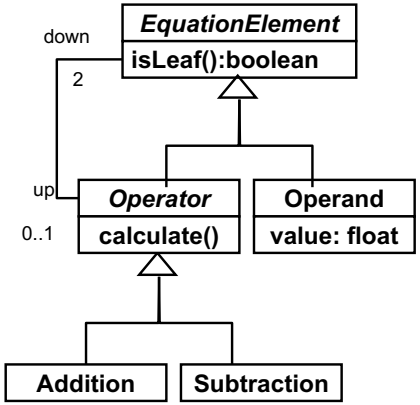
- Assume you are coding a parser to parse equations in a textual form.
  - Input: textual representation of an equation
  - Output: equivalent in-memory representation
    - Tree structure
      - Leaf nodes represent operands
      - The root and intermediate nodes represent operators.





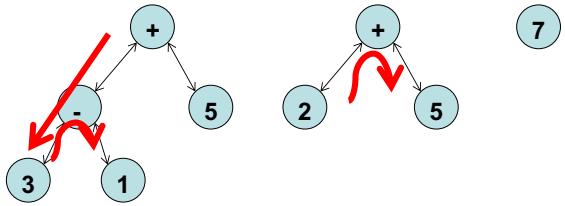


3 - 1 + 5



- Conditional statements can be eliminated in calculate()

- This parser requires a depth-first traversal policy.
  - Starts with the “deepest” and “left-most” leaf node
  - Traverse all nodes in the same layer
  - Goes up to a higher layer



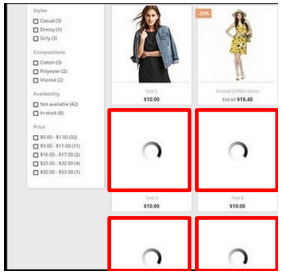
Proxy Design Pattern

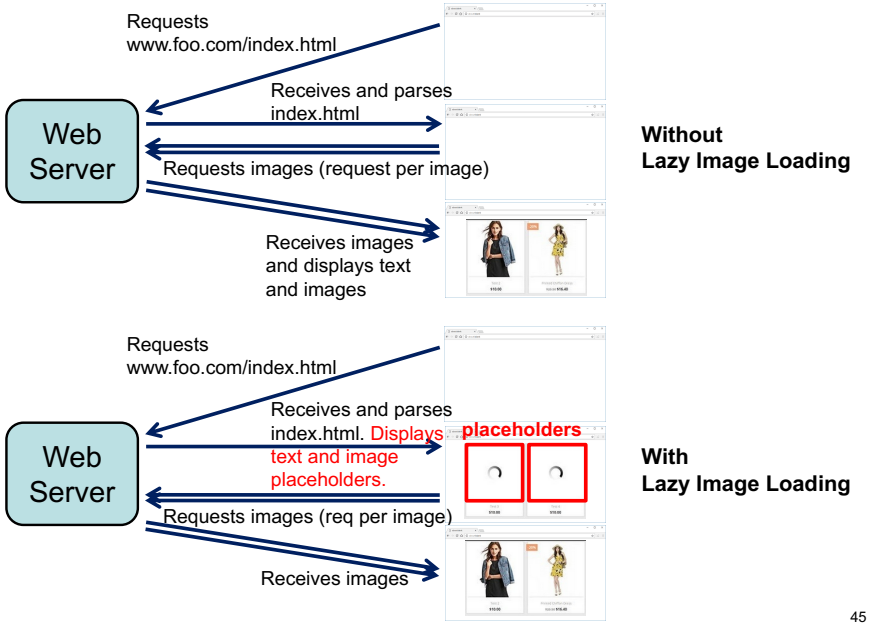
Proxy Design Pattern

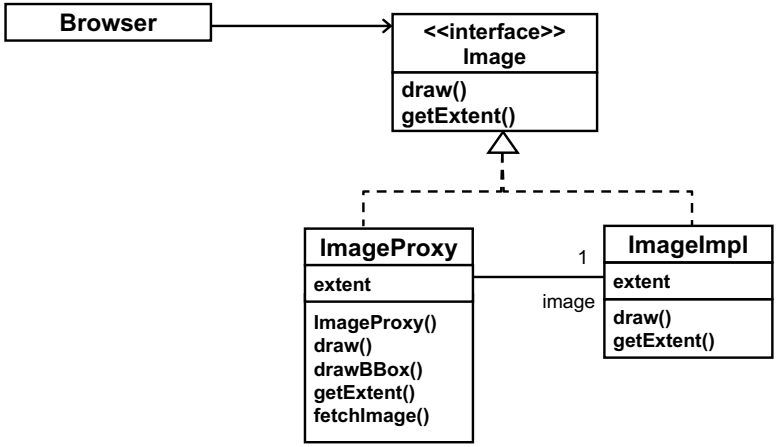
- Intent
  - Provide a surrogate or placeholder for another object to control access to it.

An Example: Lazy Image Loading in a Web Browser

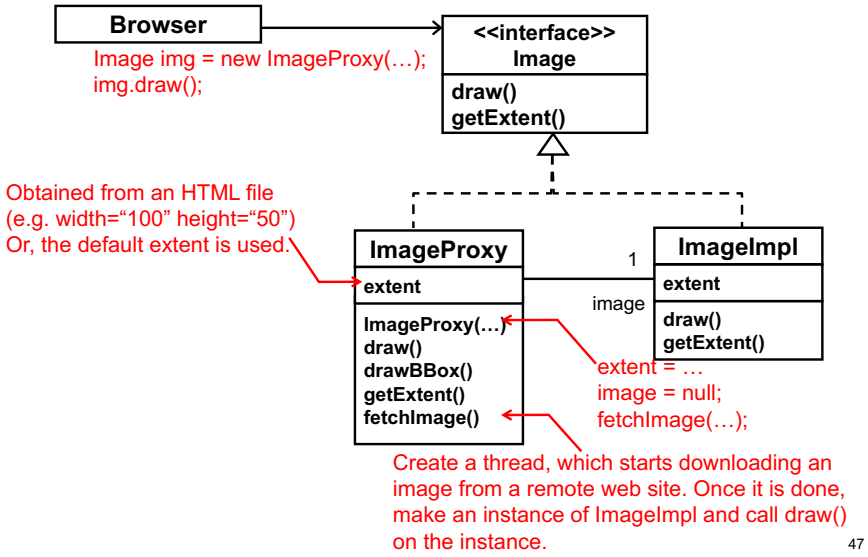
- When an HTML file contains an image(s), a browser
  - Displays a bounding box (placeholder) first for each image
    - Until it fully downloads the image.
      - Most users are not patient enough to keep watching blank browser windows until all text and images are downloaded and displayed.
  - Replaces the bounding box with the real image.

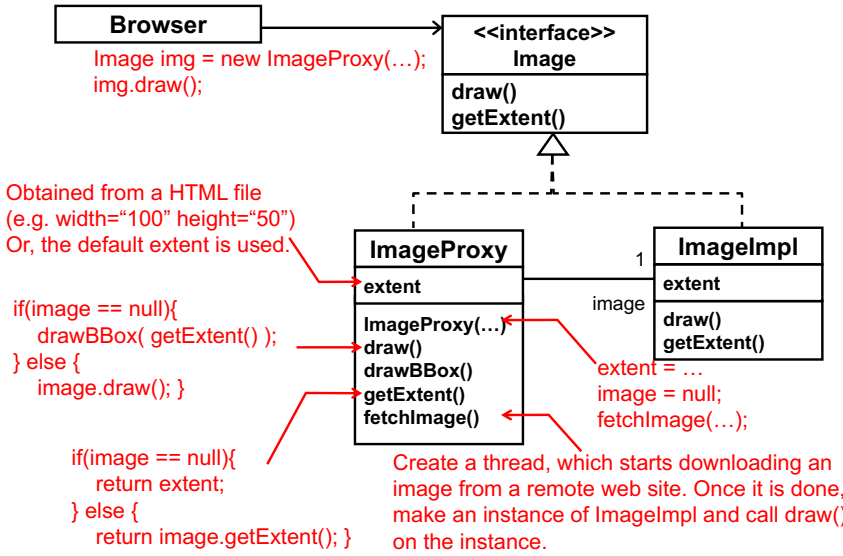






Proxy Class

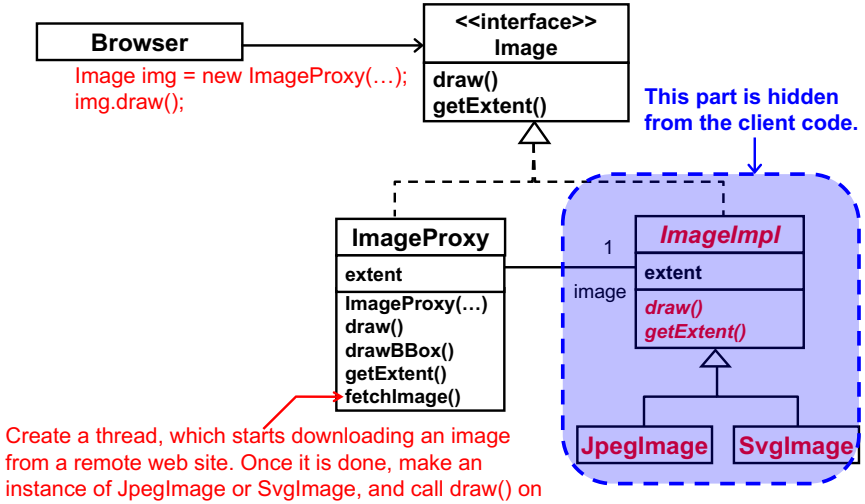




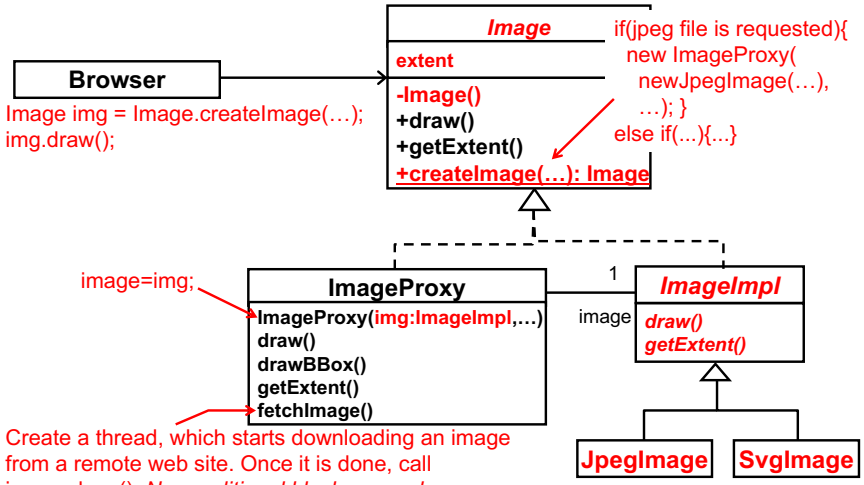
What's the Point?

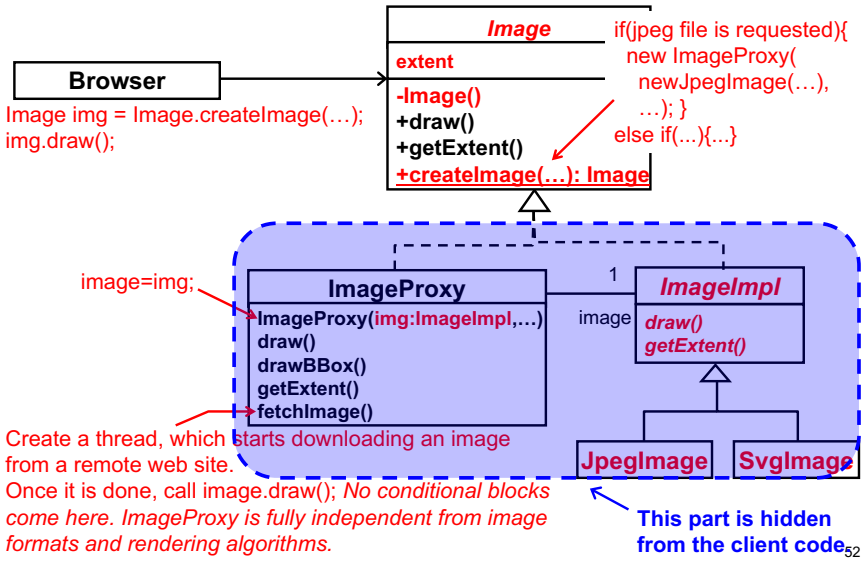
- Decouple (i.e., loosely couple) *bounding box placement* and *image rendering*.
- Why is that important?
  - Changes are expected for
    - Image formats that the browser supports.
    - Rendering algorithms
    - Communication protocols (HTTP versions)
  - Bounding box placement is independent from those changes.
  - Separate *what can change often* from *what wouldn't* to improve maintainability.

Supporting Multiple Image Formats

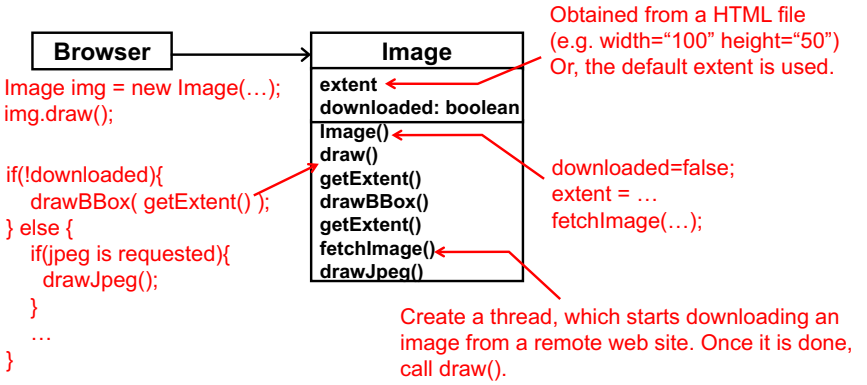


One Step Further with *Static Factory Method*





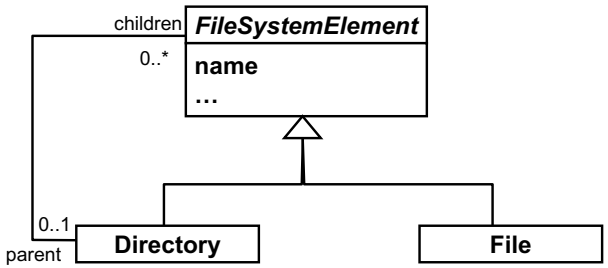
What if Everything is Integrated into a Single Class?



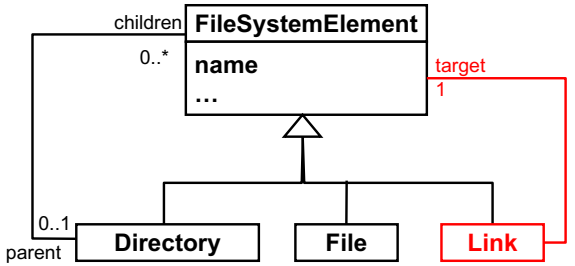
Bounding box placement, choices of image formats and image rendering are all mixed up in a single class, which will become fat and spaghetti (i.e. unmaintainable) soon.

Better design strategy: Separation of concerns

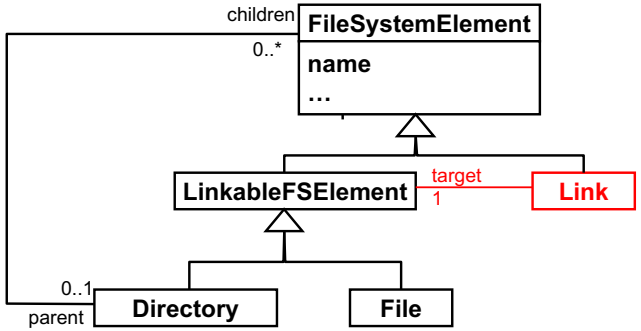
Another Example: Proxies of Files and Directories in File Systems



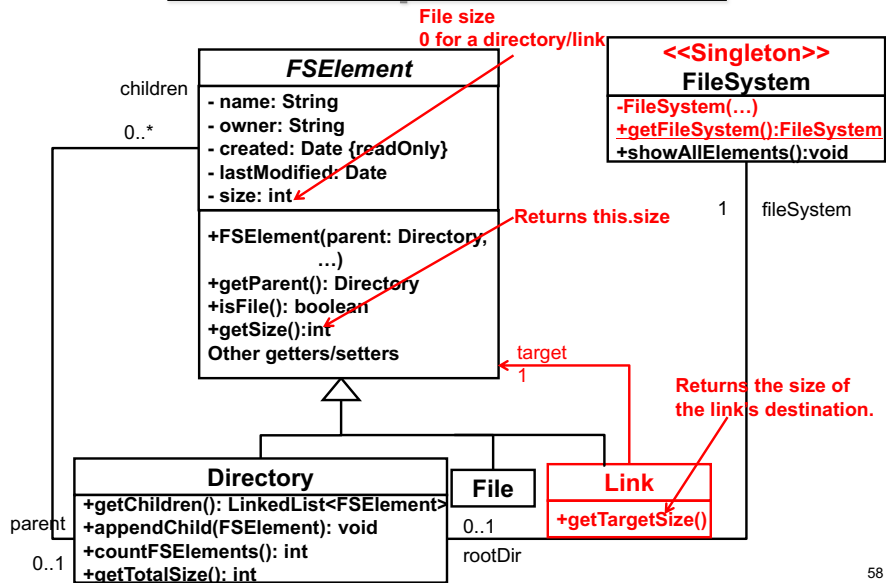
- Add a symbolic link feature
  - a.k.a. alias (Mac), shortcut (Windows)
- A link acts as a proxy of a directory or file.
- Use the Proxy design pattern.



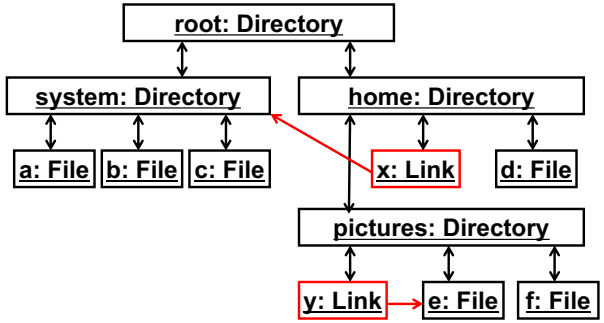
- A link acts as a proxy of a directory or file.
  - A link can act as a proxy of a link too.



HW 9: Implement this.



- If a link refers to another link, getTargetSize() goes through a chain of links until it reaches a file or directory.
- Directory
  - `LinkedList<FSElement>`: data field to reference files and subdirectories
  - `countFSElements()`: returns to total number of files and directories under a given directory.
  - `getTotalSize()`: returns the total disk consumption by all files and directories under a given directory.
- FileSystem
  - `showAllElements()`: prints out the tree structure of the entire file system.
    - You can define your own textual format.



- Use this tree structure in your test cases.
  - Set up data fields (size, owner, etc.) as you like.

- Due: April 26 (Thu) midnight