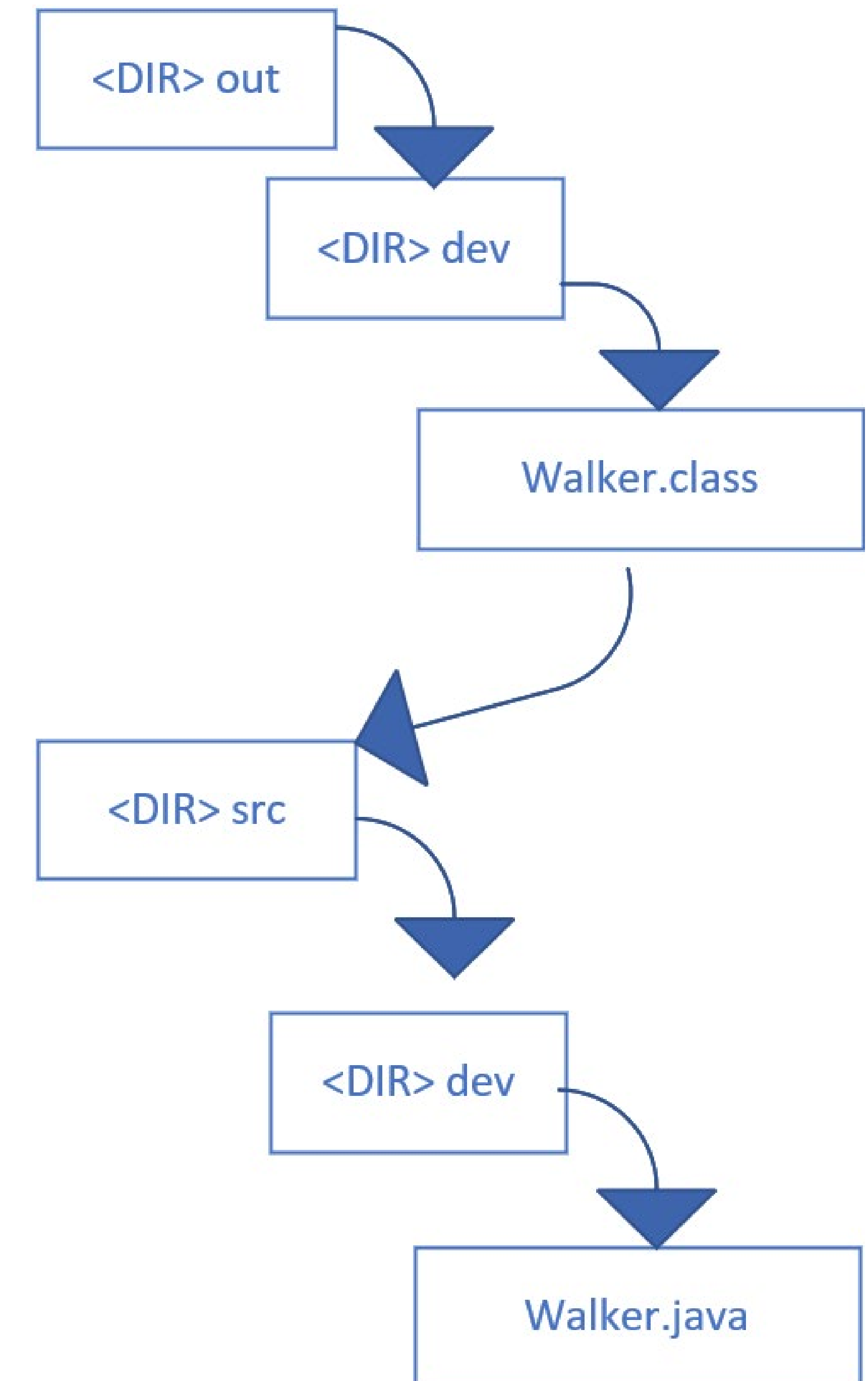


Walking the File Tree

This method walks the file tree, depth first (as does the walk method).

Depth first means, the code will recursively visit all the child elements before visiting any of a folder's siblings.

The alternative is **breadth first**, which means any dependent nodes are walked after the sibling nodes.



Why is this important?

Because it is depth first, the `Files.walkFileTree` method, provides a mechanism to accumulate information, about all the children, up to the parent.

Java provides entry points in the walk to execute operations, through a `FileVisitor` interface.

This stubs out methods you can implement, at certain events in your walk.

These events are:

- Before visiting a directory.
- After visiting a directory.
- When visiting a file.
- A failure to visit a file.

FileVisitor Interface and the SimpleFileVisitor

I've included a simple class diagram, for the FileVisitor Interface, and its simplest default implementation, SimpleFileVisitor.

You can see from these method signatures, that in most cases, you'll have access to the current path, either a directory or a file.

I didn't include the return type, which for all of these methods is the same, an enum value, as shown on the FileVisitResult enum.

In addition, you have access to basic attributes, on both the visitFile and preVisitDirectory methods.

