The Composite

Composite

Creates tree structures of related objects

Works with compositions of objects or individual objects

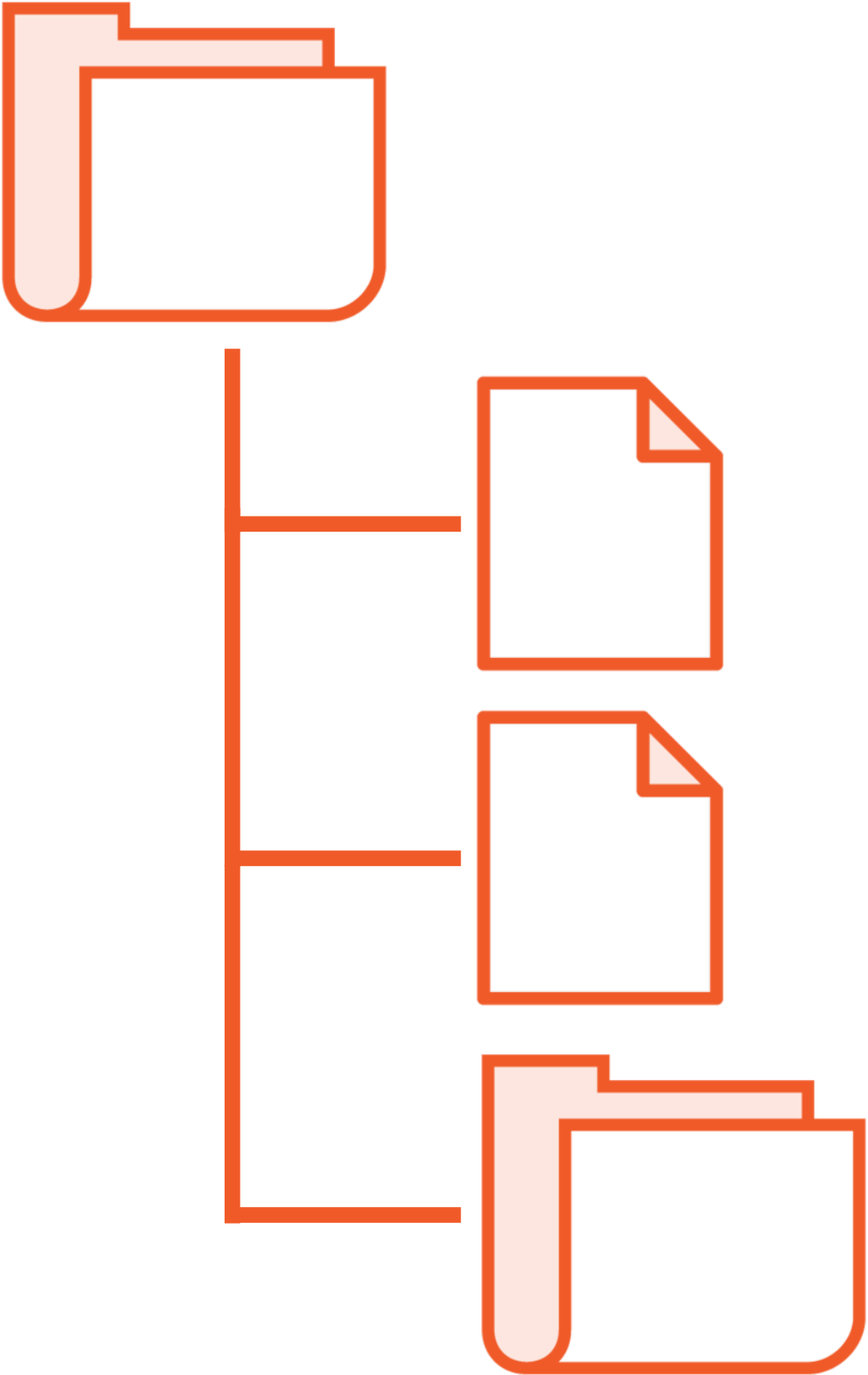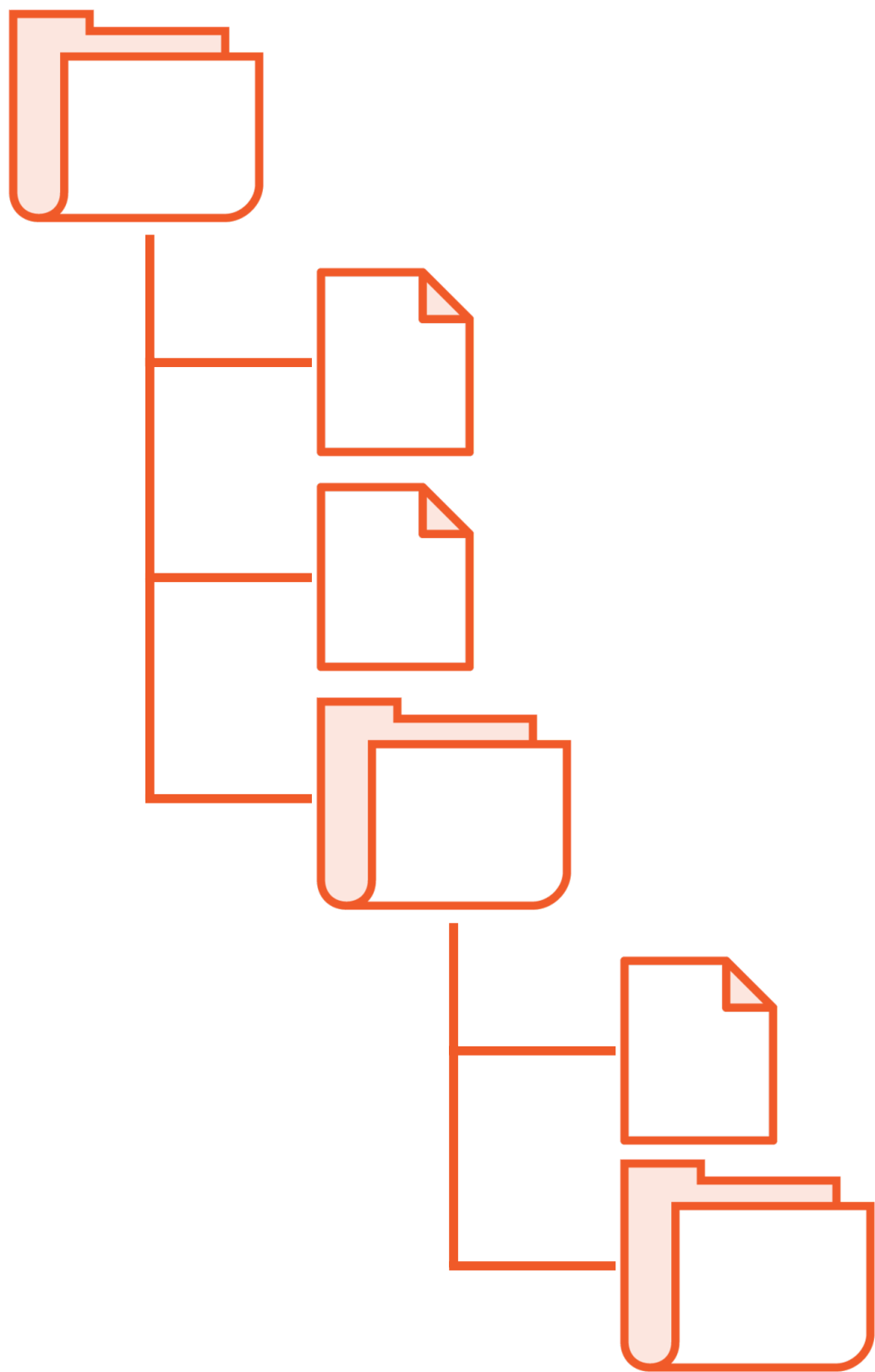Clients don't need to know the concrete types

Filesystem

Folders

Files

«interface»
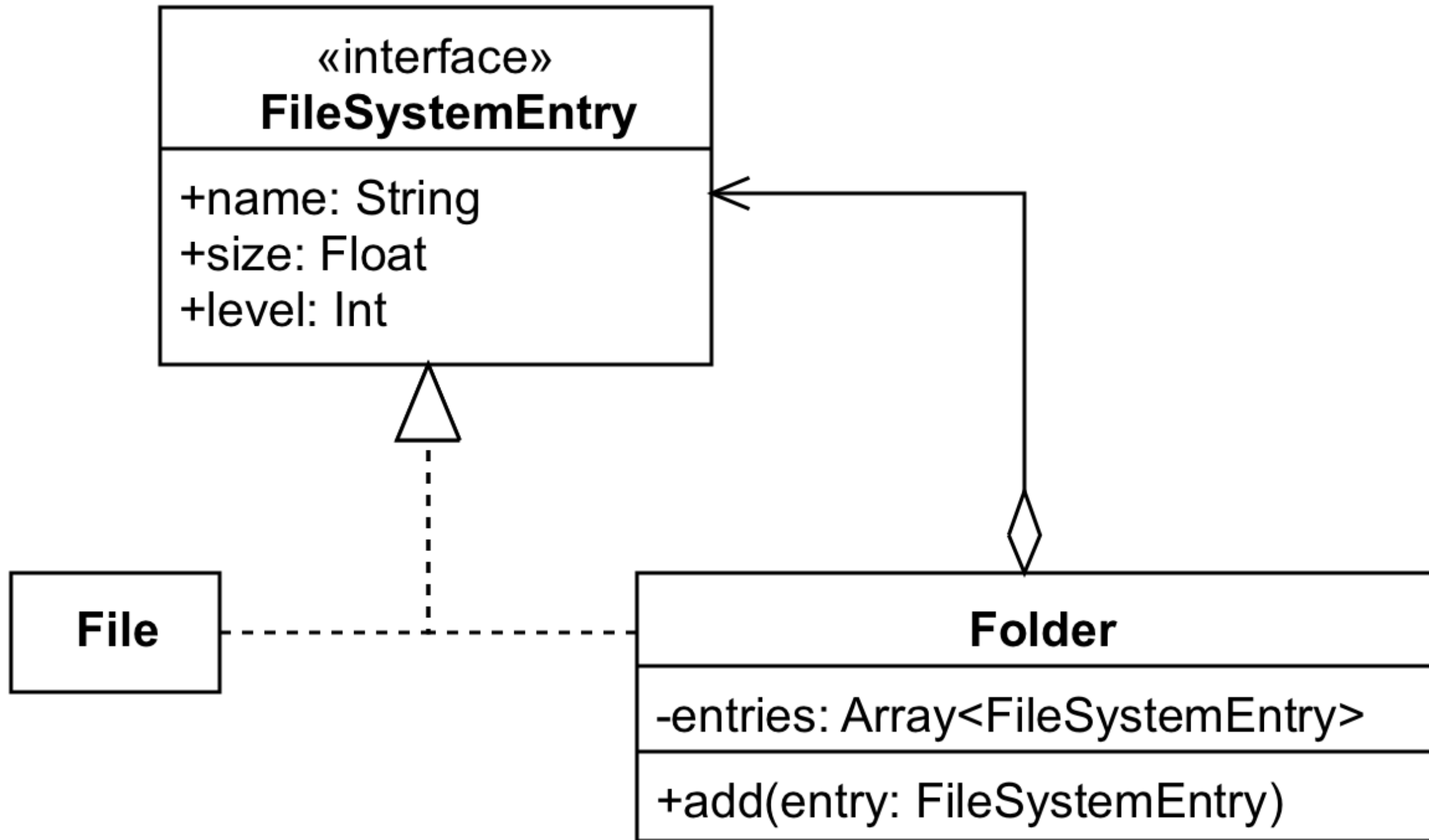**FileSystemEntry**

+name: String
+size: Float
+level: Int

File

Folder

-entries: Array<FileSystemEntry>

+add(entry: FileSystemEntry)

1.35 MB
- 120 kB
- 25 kB
- 890 kB
  - 240 kB
  - 610 kB
    - 600 kB
    - 10 kB
  - 40 kB
    - 40 kB

```
┌─────────────────────────────────┐
│            «interface»          │
│          FileSystemEntry        │
├─────────────────────────────────┤
│ +name: String                   │
│ +size: Float                    │◄────────┐
│ +level: Int                     │         │
└─────────────────────────────────┘         │
              △                              │
              ┊                              ◇
              ┊                    ┌──────────────────────────────────────┐
┌──────────┐  ┊                    │                Folder                │
│   File   │┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄│├──────────────────────────────────────┤
└──────────┘                       │ -entries: Array<FileSystemEntry>     │
                                   ├──────────────────────────────────────┤
                                   │ +add(entry: FileSystemEntry)         │
                                   └──────────────────────────────────────┘
```
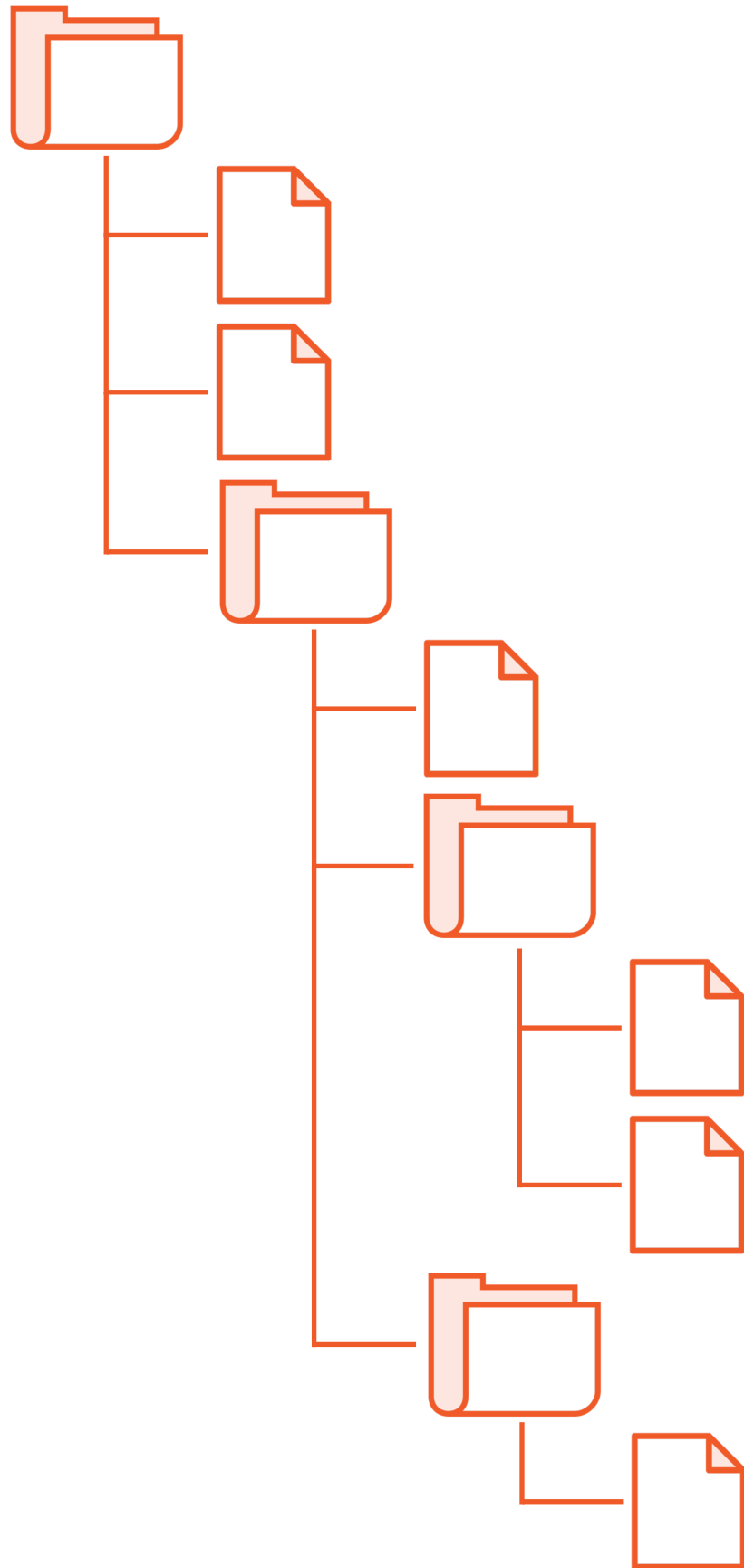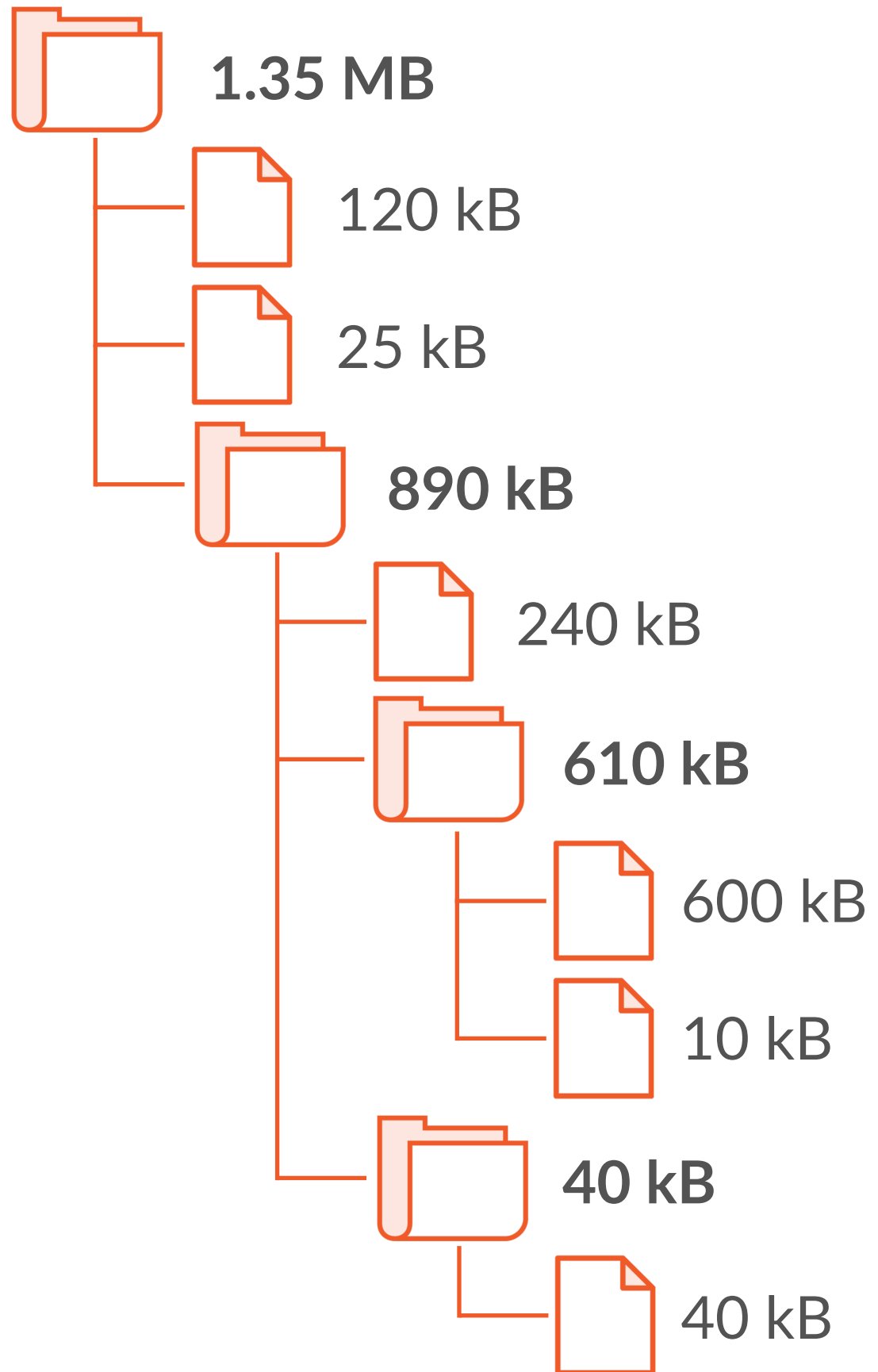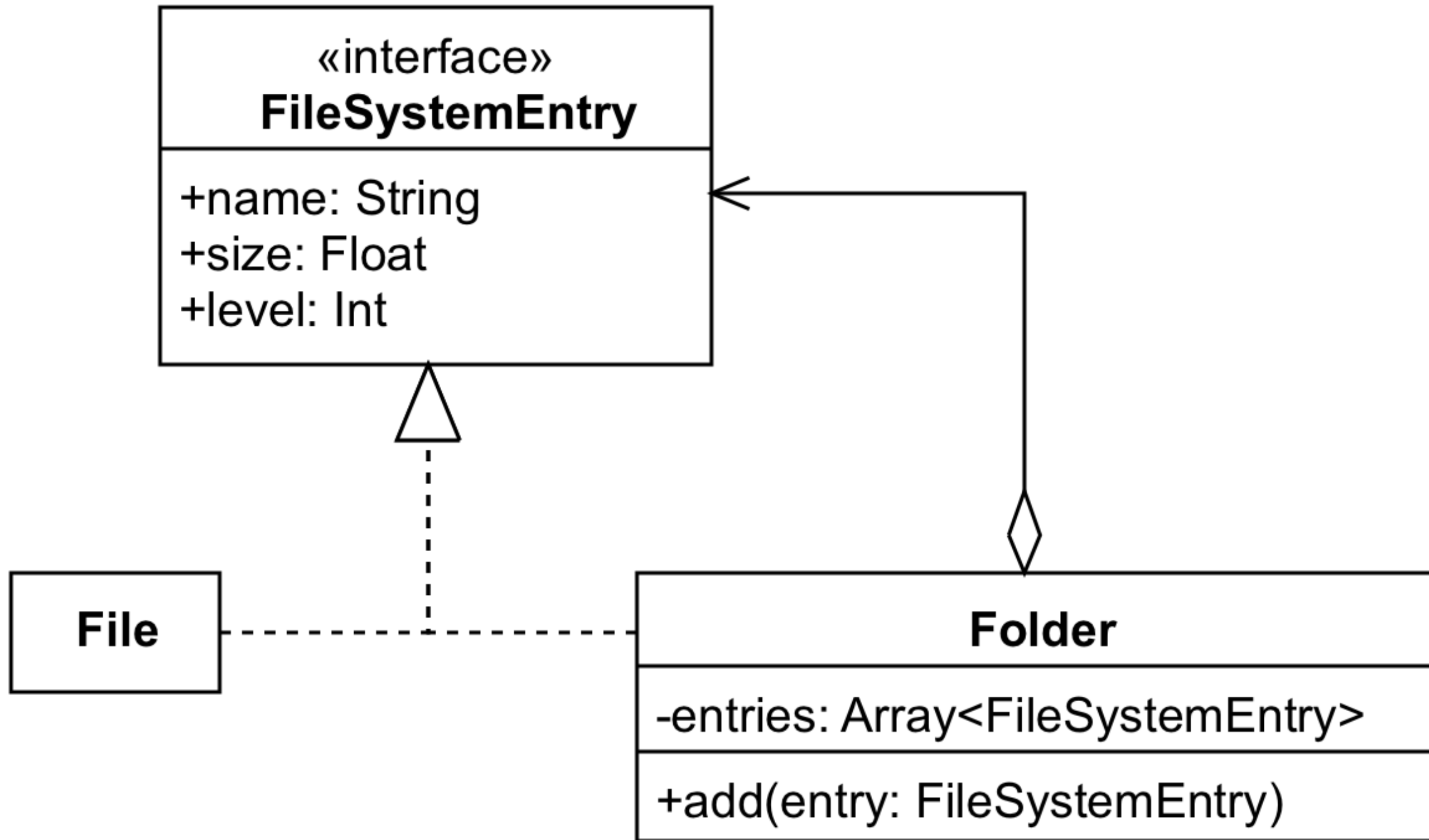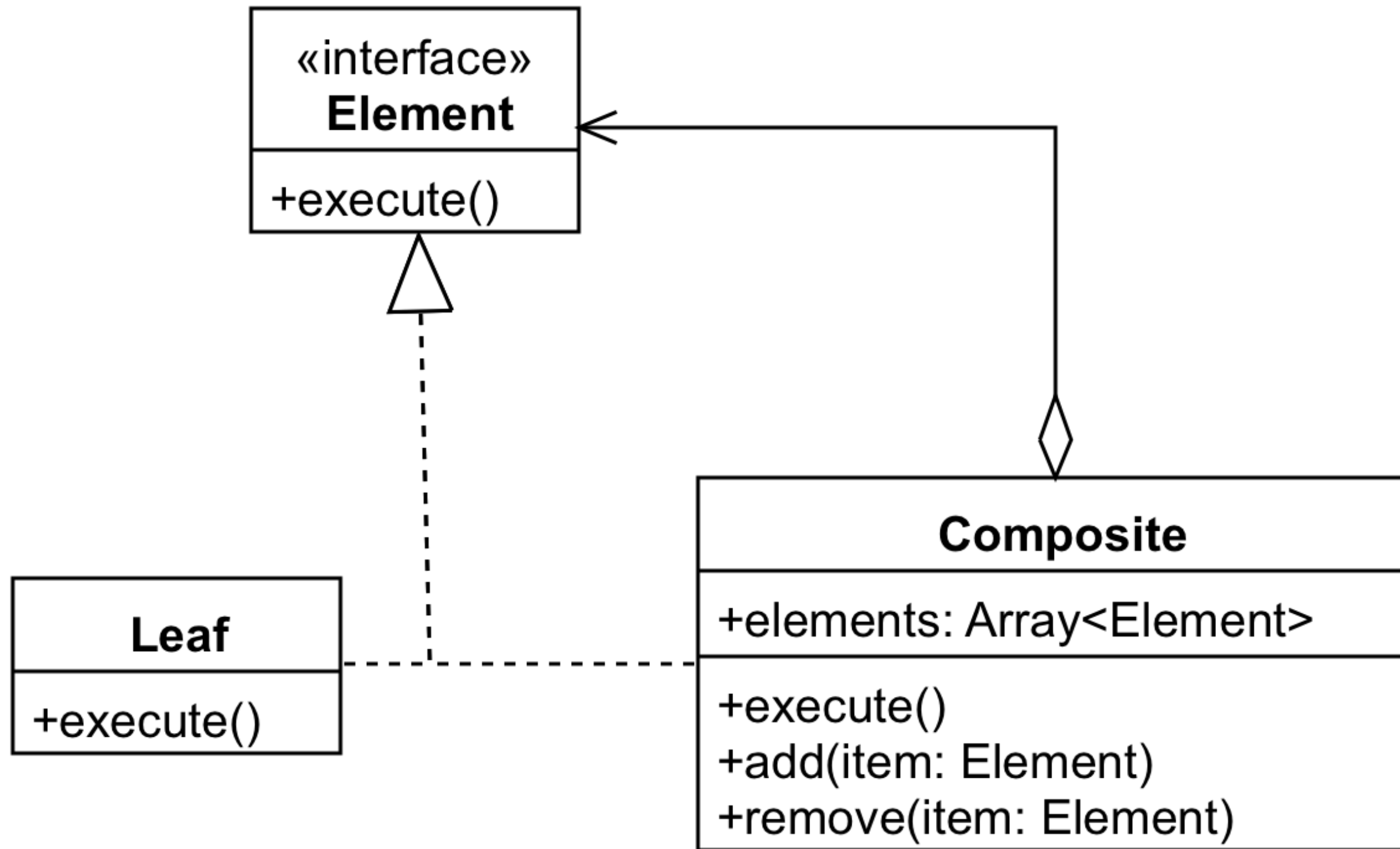
Composite

Makes it easier to work with simple and complex objects organized in tree structures.

```
«interface»
Element
+execute()
```

```
Leaf
+execute()
```

```
Composite
+elements: Array<Element>
+execute()
+add(item: Element)
+remove(item: Element)
```

Element
defines the interface

«interface»
Element

+execute()

Leaf

+execute()

Composite

+elements: Array<Element>

+execute()
+add(item: Element)
+remove(item: Element)
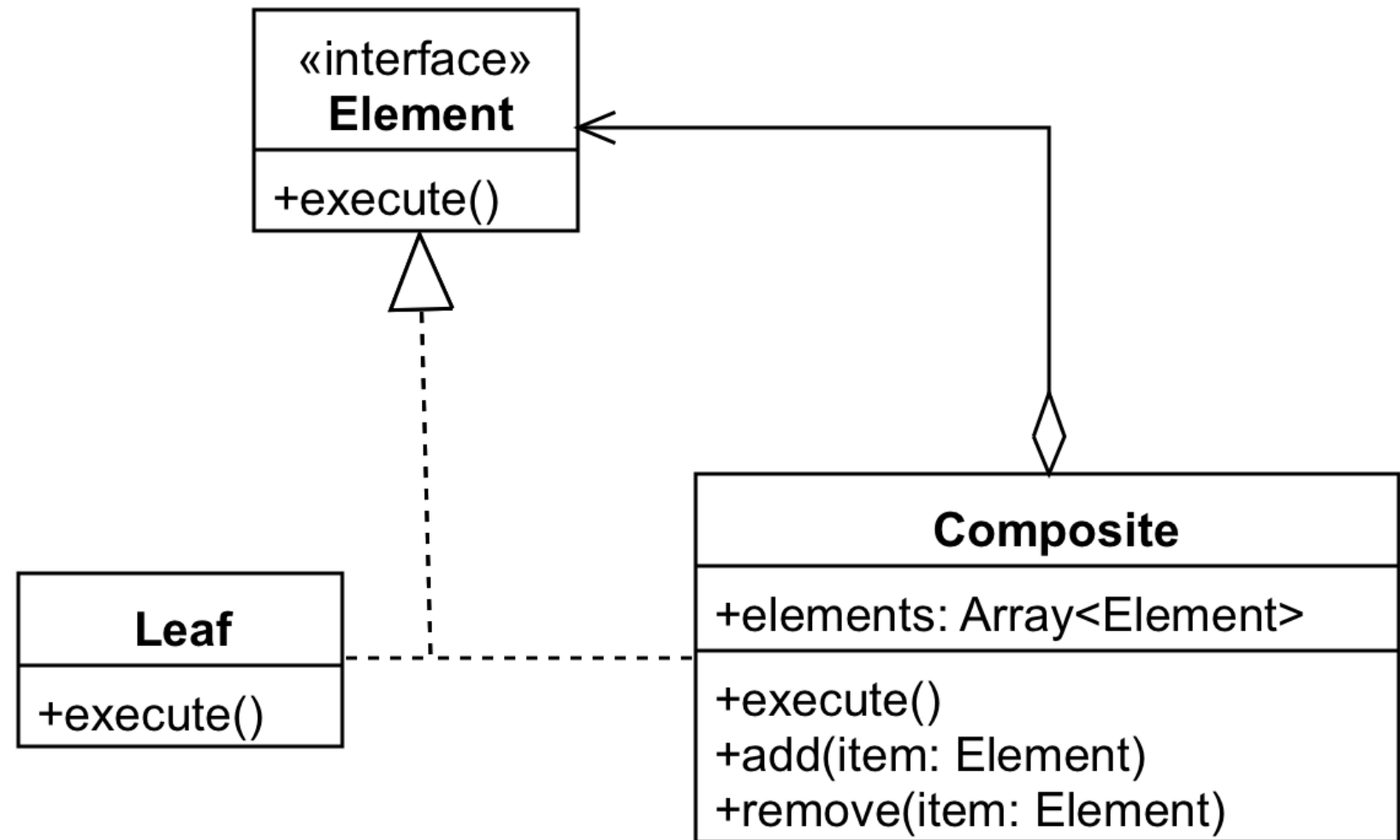
Composite stores objects of Element type

«interface»
Element

+execute()

Leaf

+execute()

Composite

+elements: Array<Element>

+execute()
+add(item: Element)
+remove(item: Element)

Leaf
is the atomic type

«interface»
Element

+execute()

Leaf

+execute()

Composite

+elements: Array<Element>

+execute()
+add(item: Element)
+remove(item: Element)
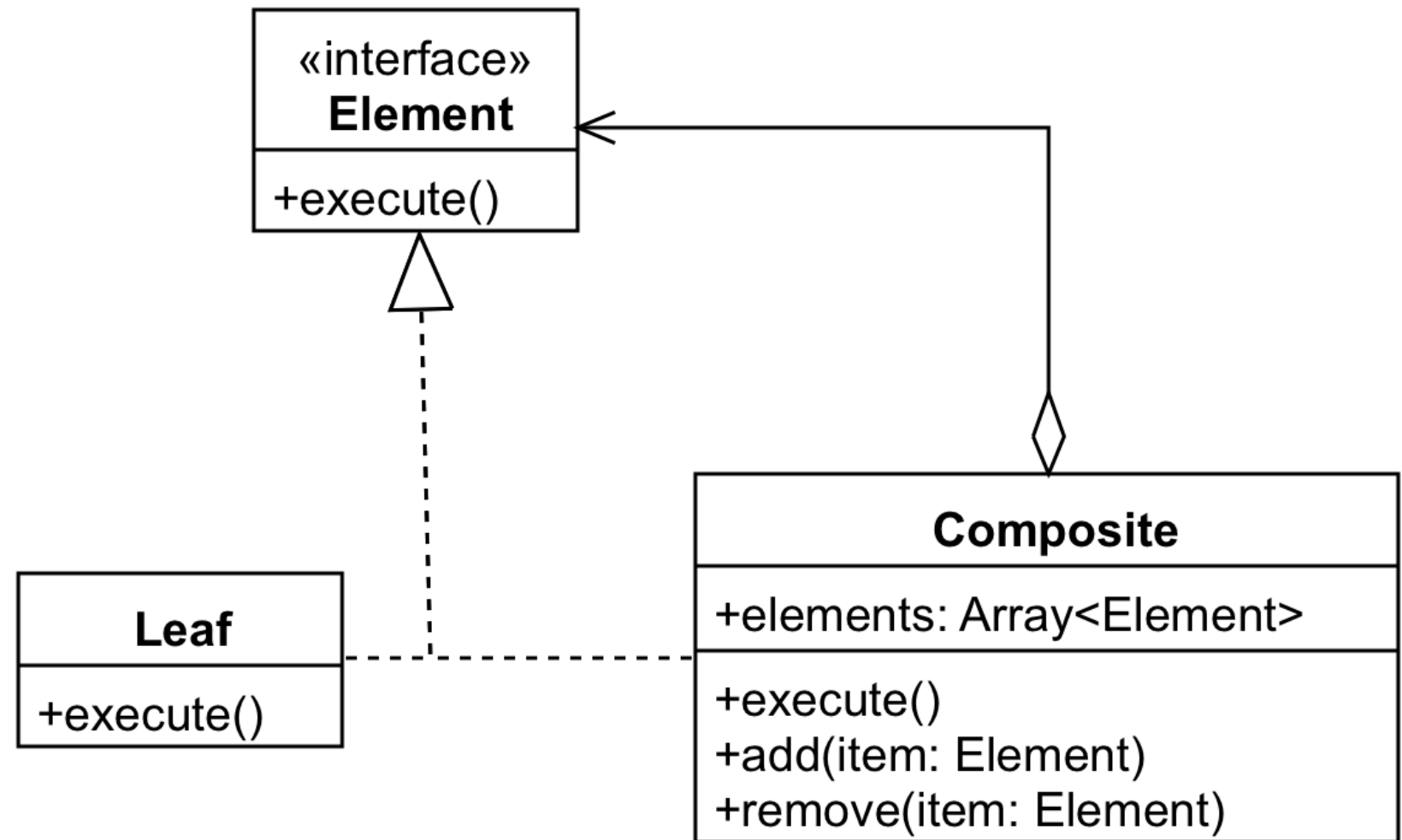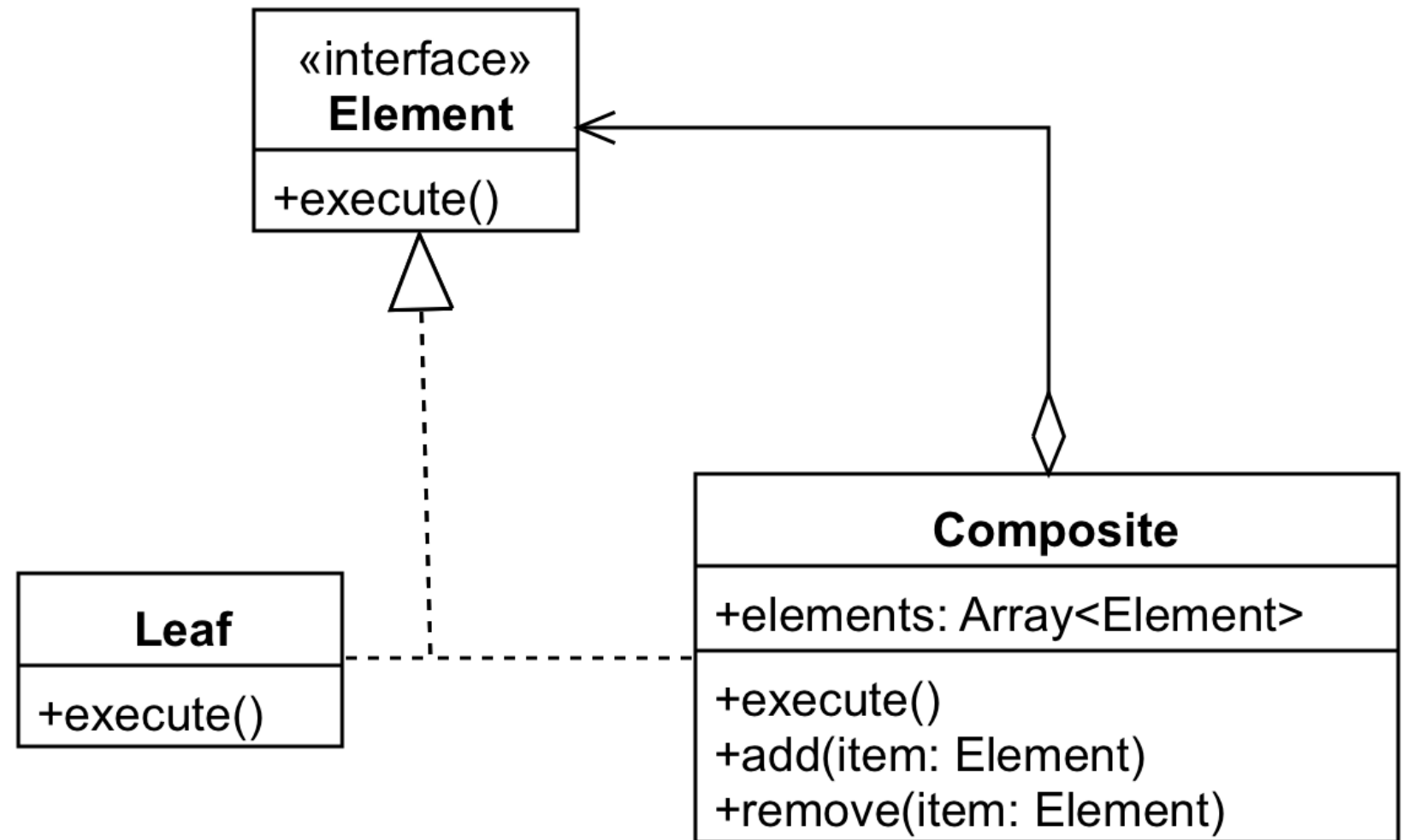
Demo

Tree Command Line App

- Lists a directory branch

- Calculates the size of the folder

Folder

File

File

Folder
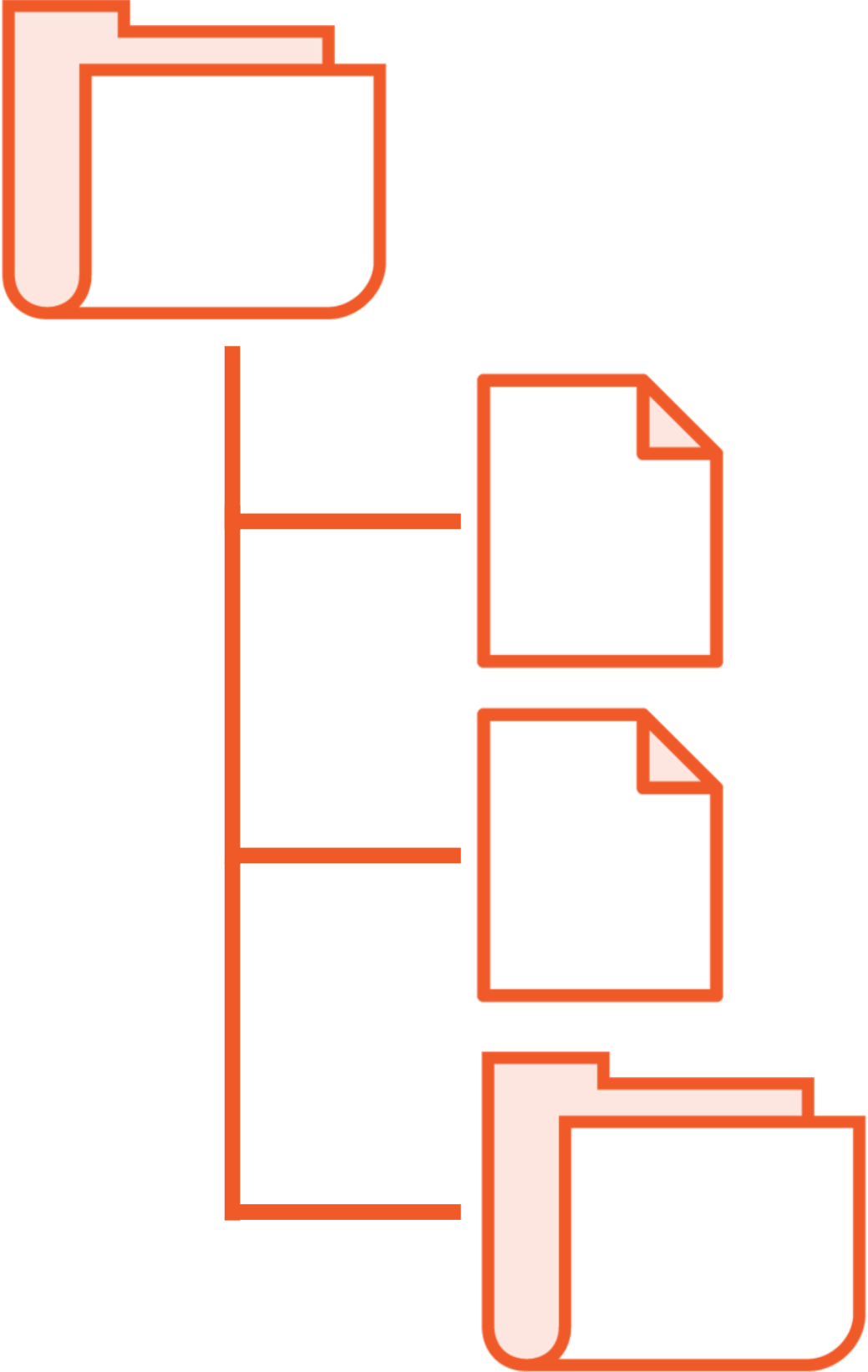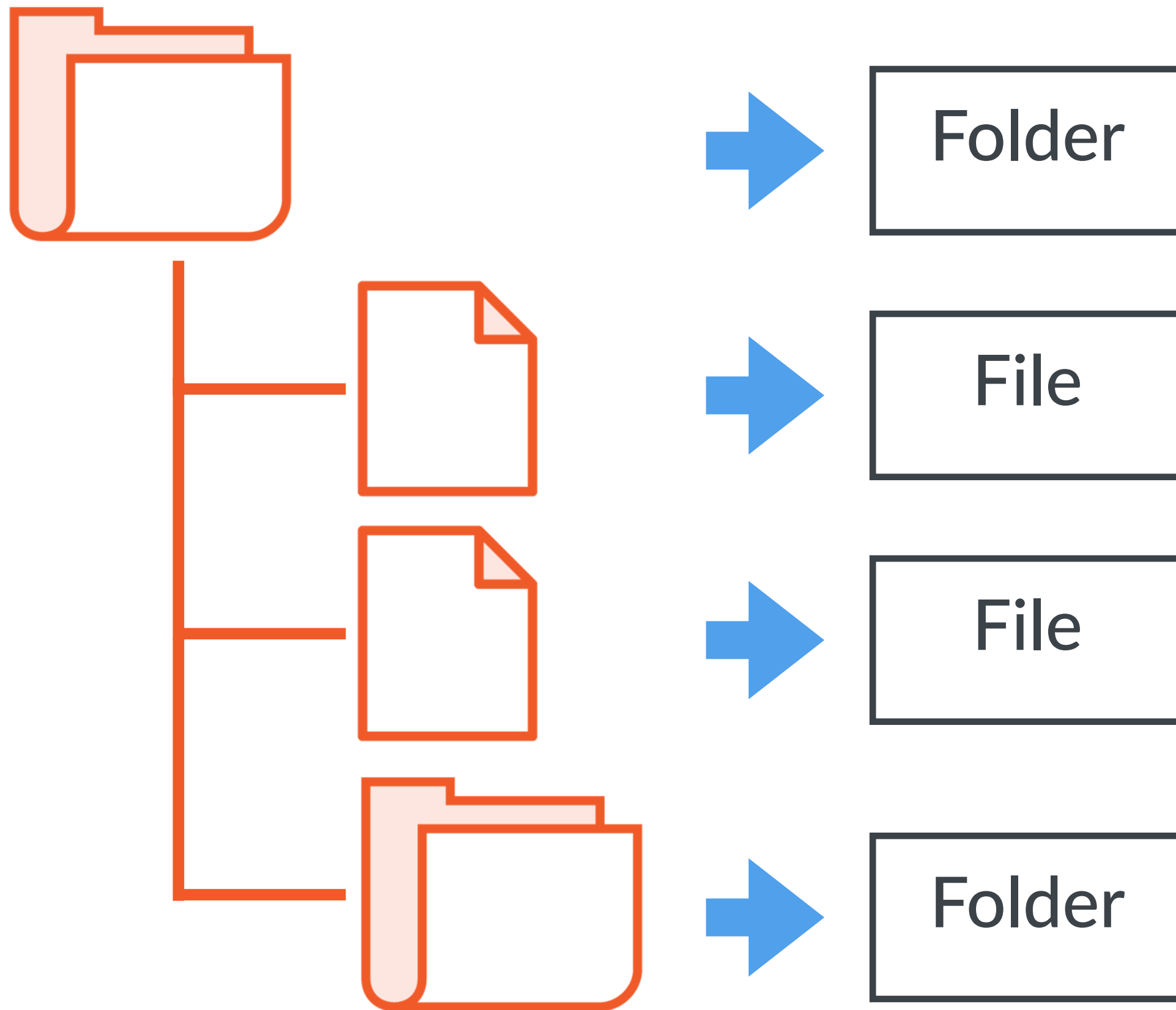
The Composite

Purpose

- Makes it easier to work with objects organized in a tree structure

- Treats simple and complex objects uniformly

- Avoids type checks and conditional logic

- Produces cleaner code