

# Composite Pattern

- Composite is a structural pattern that allows you to compose objects into a tree and work with them as if they were individual objects
- It is used when we can structure our data into a tree (ex. Company organizational structure, package with smaller packages inside, etc)
- With the composite pattern, we don't have to worry about the tree structure, as objects delegate tasks to their subtree.
- However, sometimes, if tree elements are too different, the composite pattern may not work well

# Composite Pattern

- Composite pattern is implemented with the following:
  - An interface defining common behavior between elements in the tree
  - Composite classes that links several elements together
  - Leaf elements that execute core functionality
- When the interface calls “execute”, the operation is propagated down the tree to the leaves.
- Most of the time the leaves will do the majority of the work.

