Composite

1. Treating individual and aggregate objects uniformly.
2. Objects use other objects’ fileds/methods through inheritance and composition.
3. Composition lets us make compound objects
   1. E.g. a mathematical expression composed of simple expressions; or
   2. A shape group made of several different shapes
4. Composite design pattern is used to treat both single (scalar) and composite objects uniformly
   1. Foo and List<Foo> have common APIs
5. Composite: A mechanism for treating individual (scalar) objects and compositions of objects in a uniform manner.

Geometric Shapes

1. We will have an object that can contain many children of the same object and perform group operations like print.
2. Example given with circle , square and graphicObject. (Written in code)

Summary

1. Objects can use other objects via inheritance /composition
2. Some composed and singular objects need similar/identical behaviors
3. Composite design pattern lets us treat both types of objects uniformly
4. Java supports container iteration with the Iterable<T> interface
5. A single object can masquerade as a collection by returning a single-element collection containing only this.