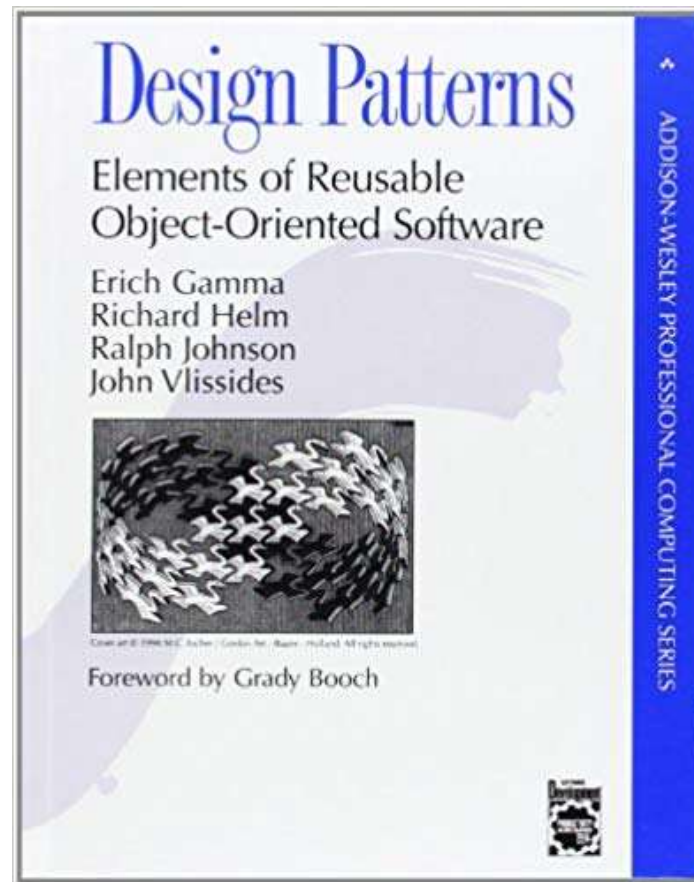


Design Patterns

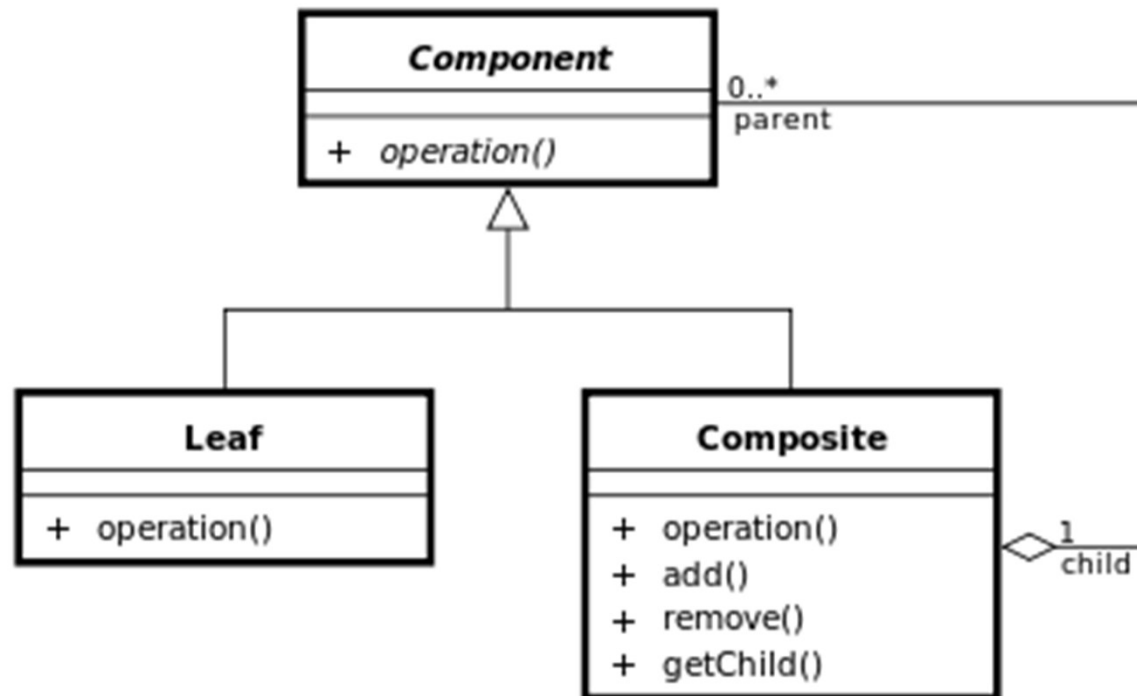
- ❑ In software development, a design pattern is a general and reusable solution to a commonly occurring problem.
- ❑ A design pattern can solve many problems by providing a framework for building an application.
- ❑ With design patterns, the design process is cleaner and more efficient.

Book from the “Gang of Four”

□ 23 patterns:



Composite Pattern



(from Wiki)

Exercise

- Design and implement the File&Folder structure using the Composite Pattern



You are tasked with designing a system for a shop selling Matryoshka dolls (also known as stacking dolls or nesting dolls), one of the iconic toys of Russia. Each Matryoshka doll consists of two essential components: the outer shell (representing the doll itself) and the inner dolls nested within (see illustrative figure below). The inner-most doll does not nest anything. The appearance of the outer shell and the inner dolls typically *look almost the same* and are *just different in their sizes*. Currently, the shop sells three models of Matryoshka dolls, *a Japanese geisha wearing a kimono, a Russian princess in a sarafan and a Vietnamese girl in Ao Dai*. These three models have different design descriptions and prices. The price of a doll model is fundamentally based on the outer shell and the number of dolls nested within. The price of the outer shell varies across the models depending on their decorations. In this case, the outer shell doll's price can be manually set by the shop. The price of an inner-layer doll is equal to 80% of the price of the outer-layer doll which directly nests it (e.g. a doll with a nesting level of 1 will be priced at 80% the price of the doll at the nesting level 0 (the outer shell) and so on).

For easy management of the dolls, each doll has an identity in the system. Besides that, the outer shell or inner dolls of each doll will be marked with a nesting level, starting from 0 for the outer shell and increased by 1 for each nesting layer.

Provide a UML diagram and declarations of all classes (including attributes and signatures of methods) that you believe will be needed to implement the management system for this shop. Please consider how to design the system so that it will be easily maintainable and extendable in the future.

