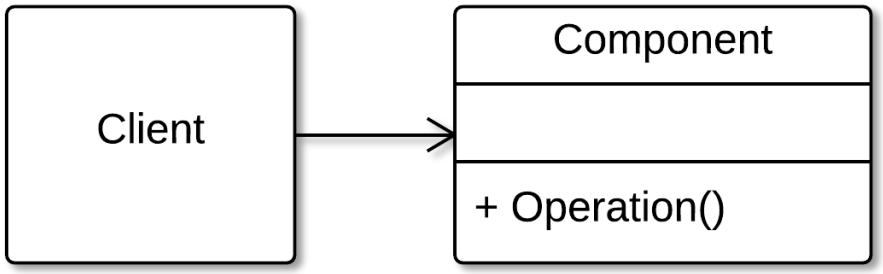
Real World Composition Pitfalls

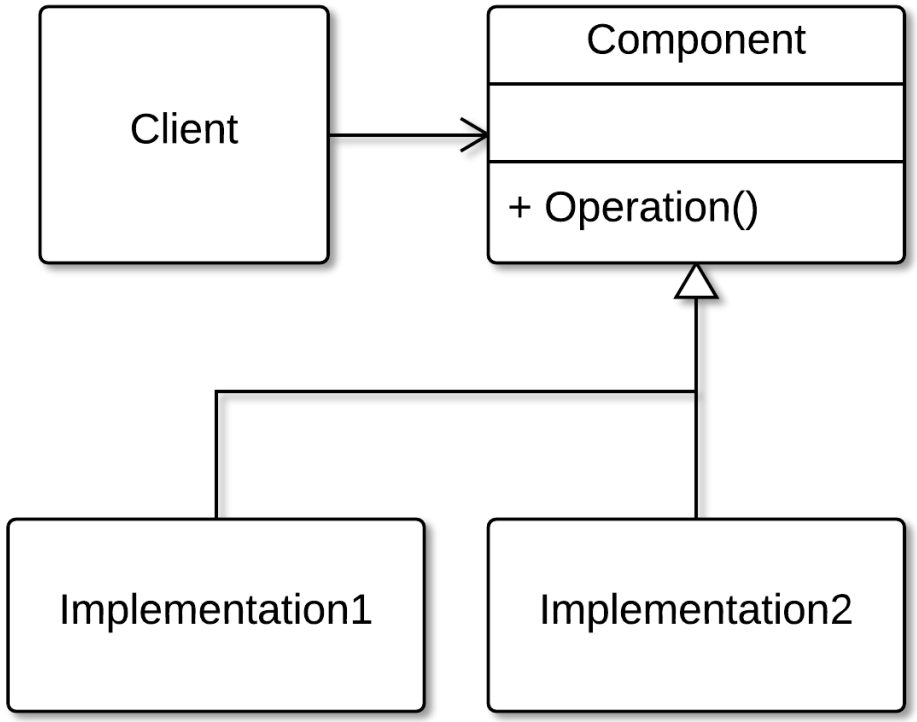
Zoran Horvat

@zoranh75 | www.codinghelmet.com

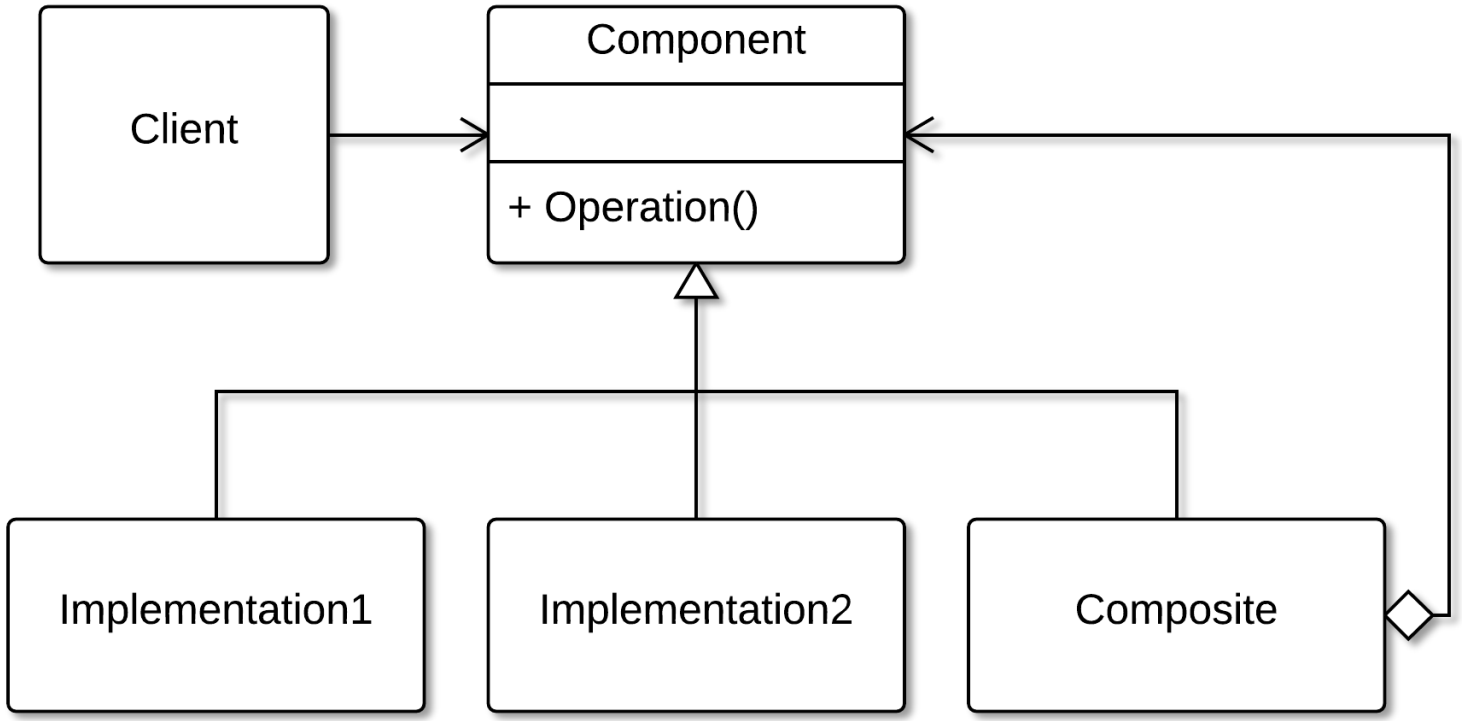
About the Composite Design Pattern



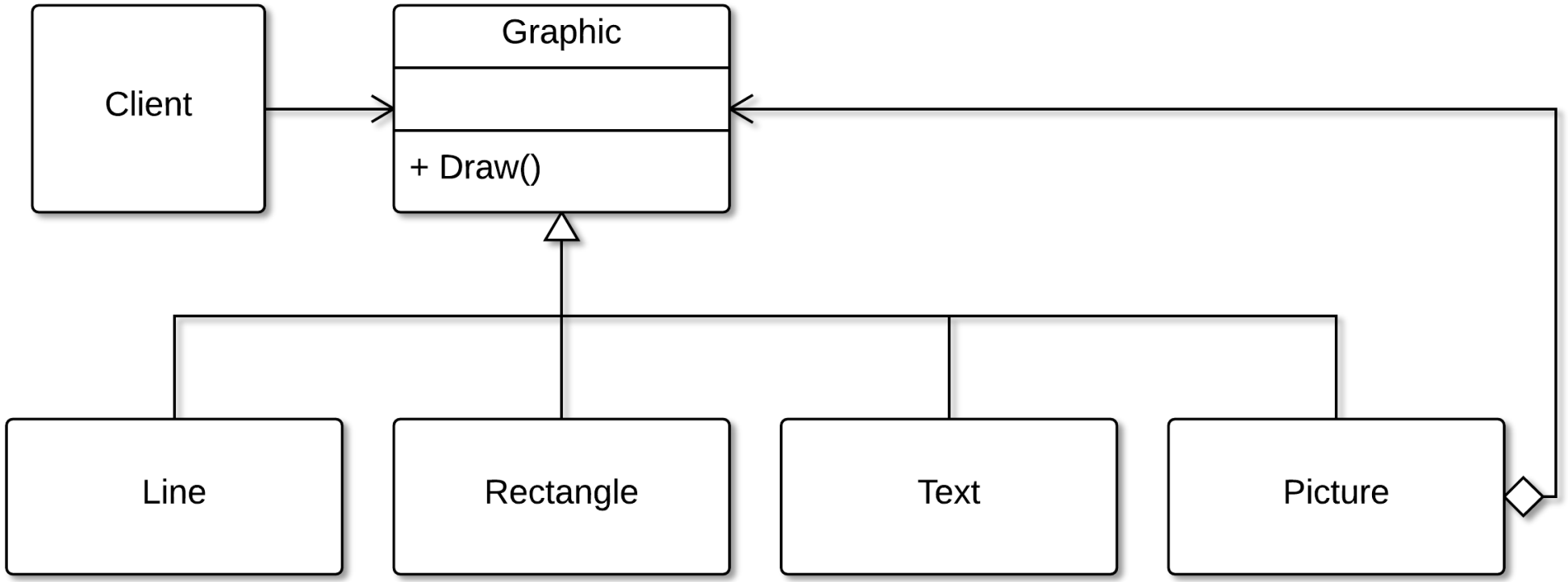
About the Composite Design Pattern



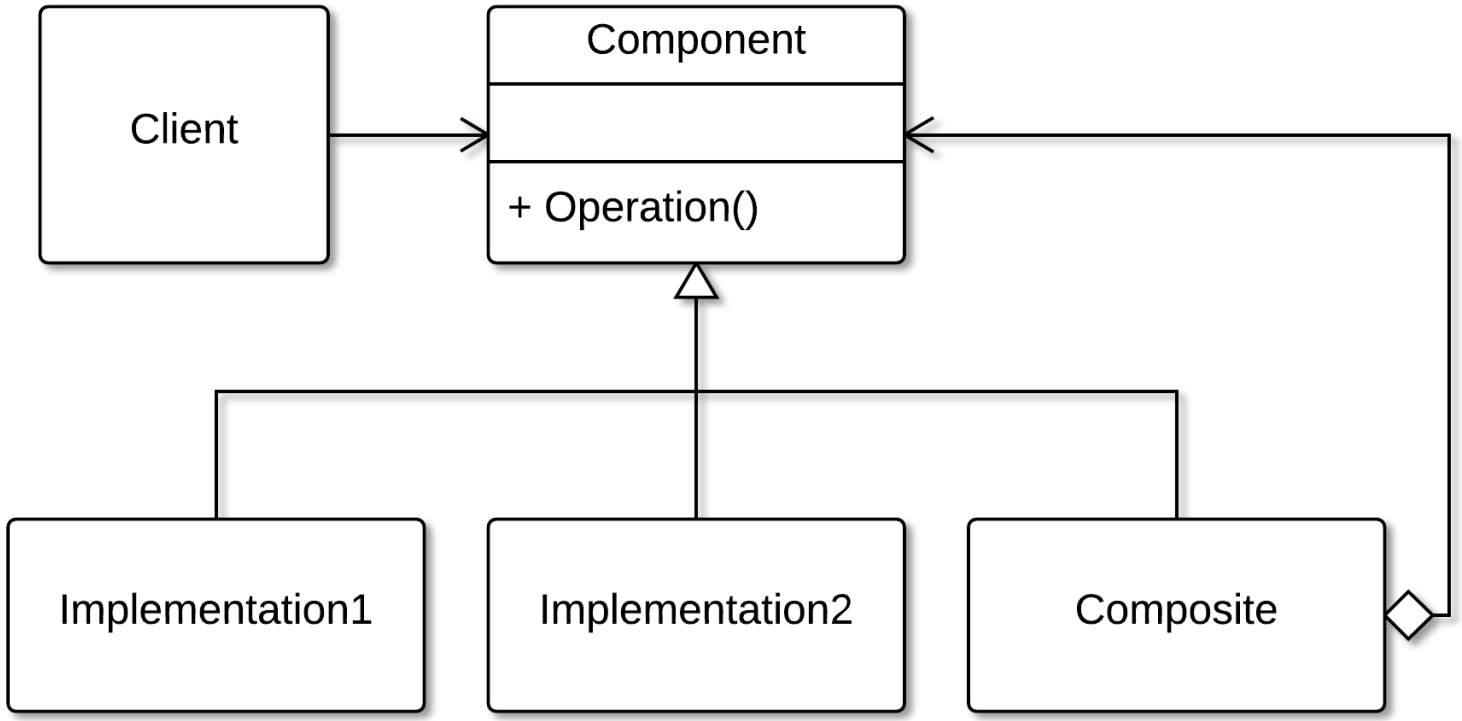
About the Composite Design Pattern



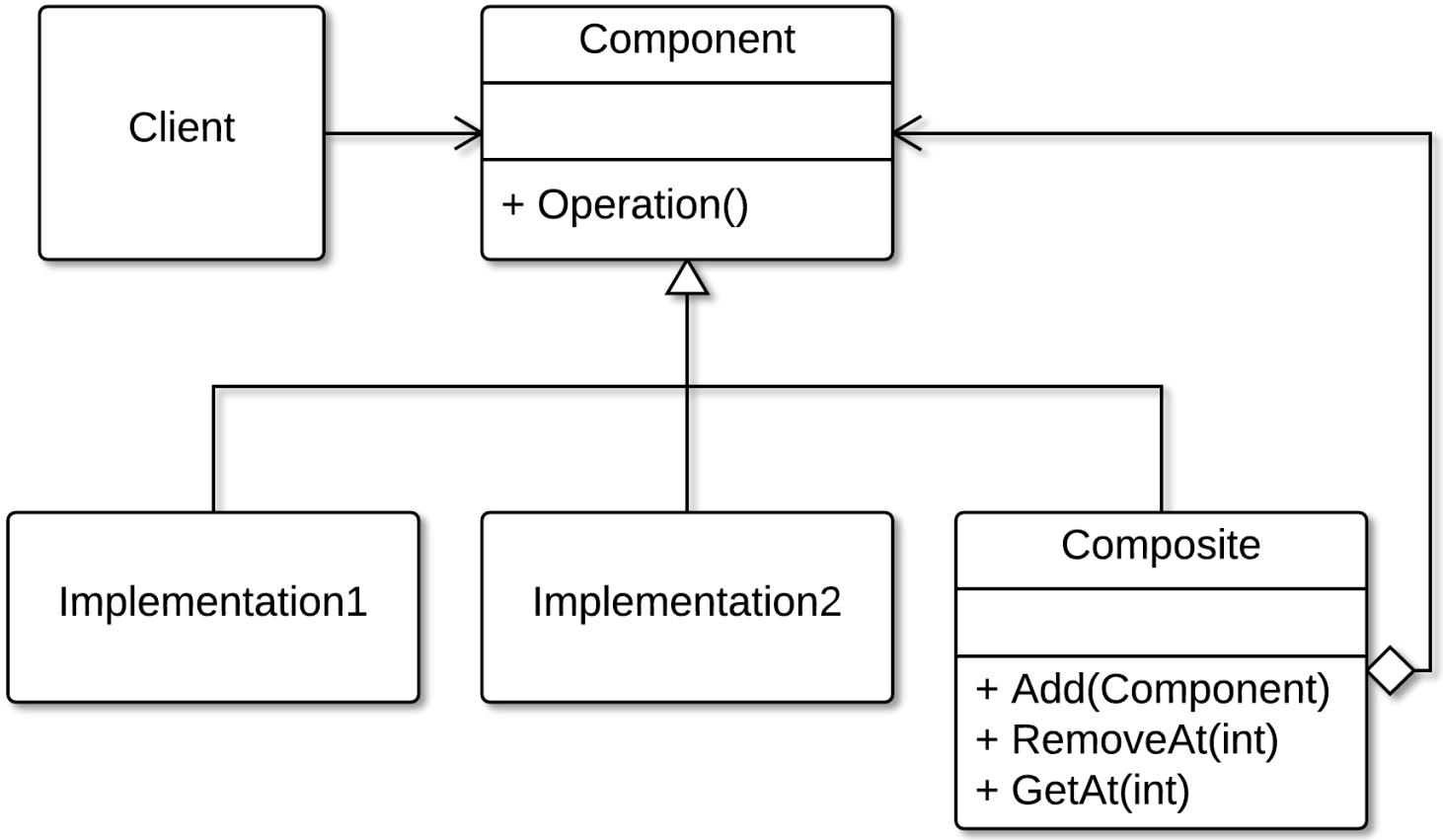
About the Composite Design Pattern



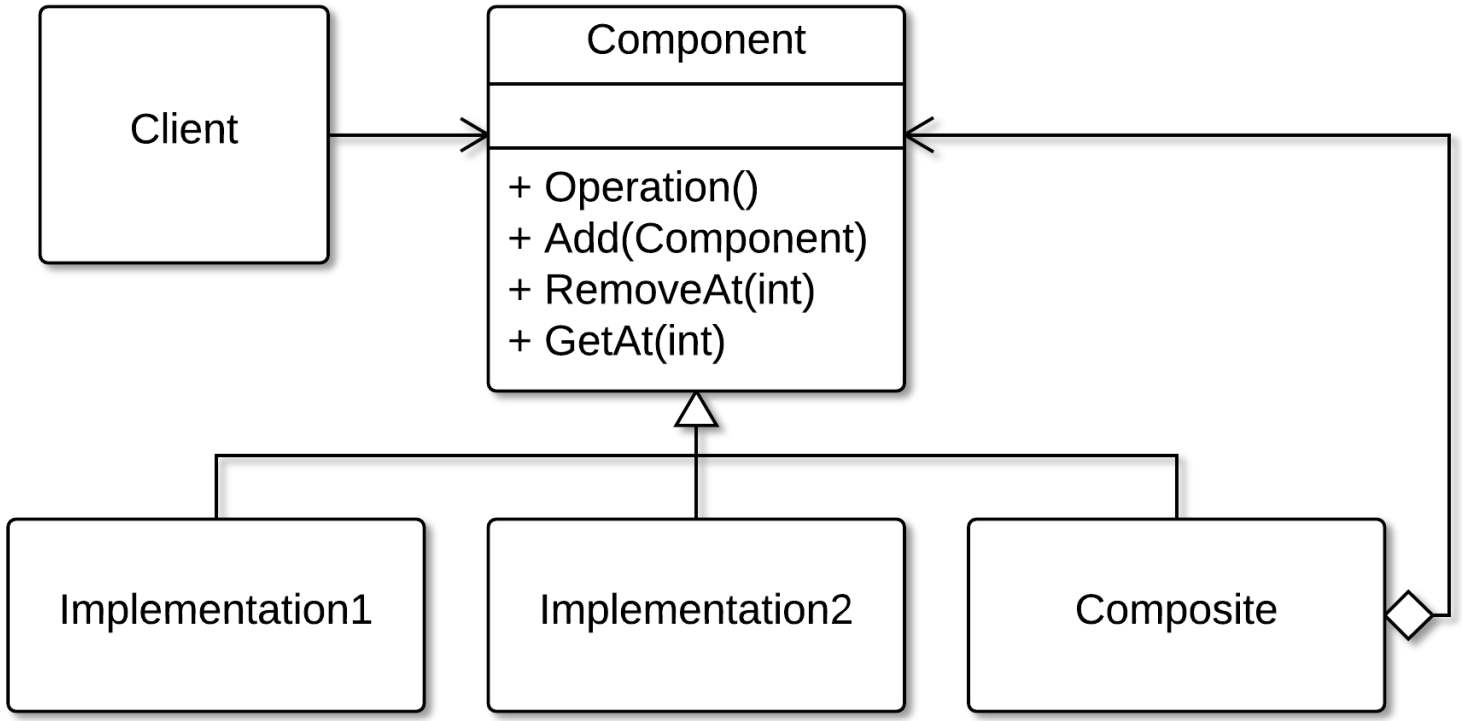
About the Composite Design Pattern

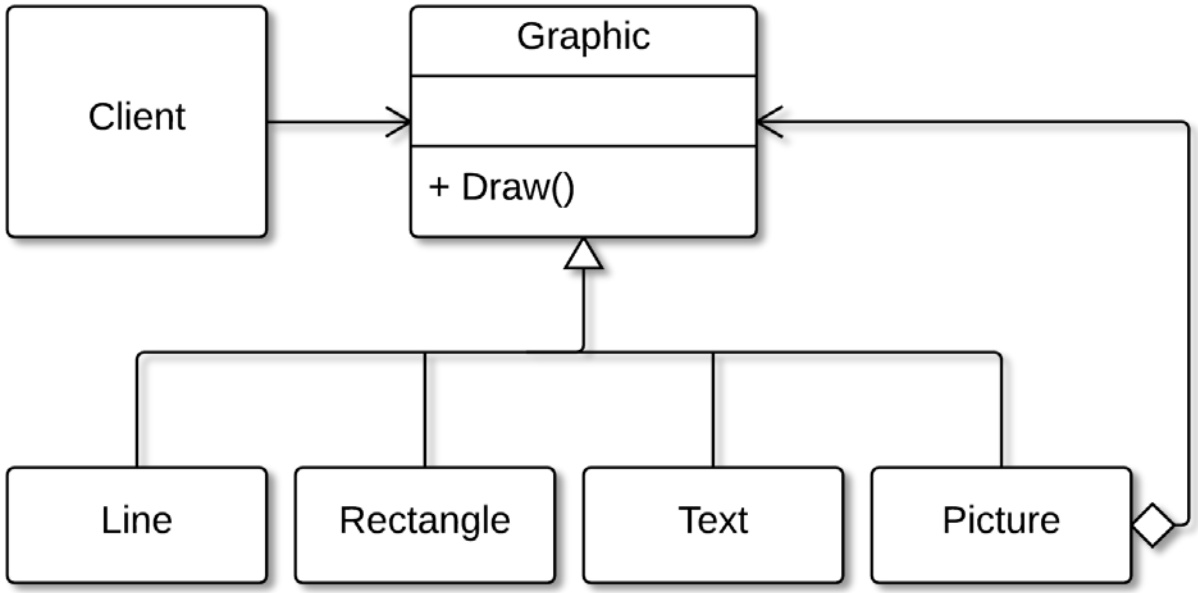


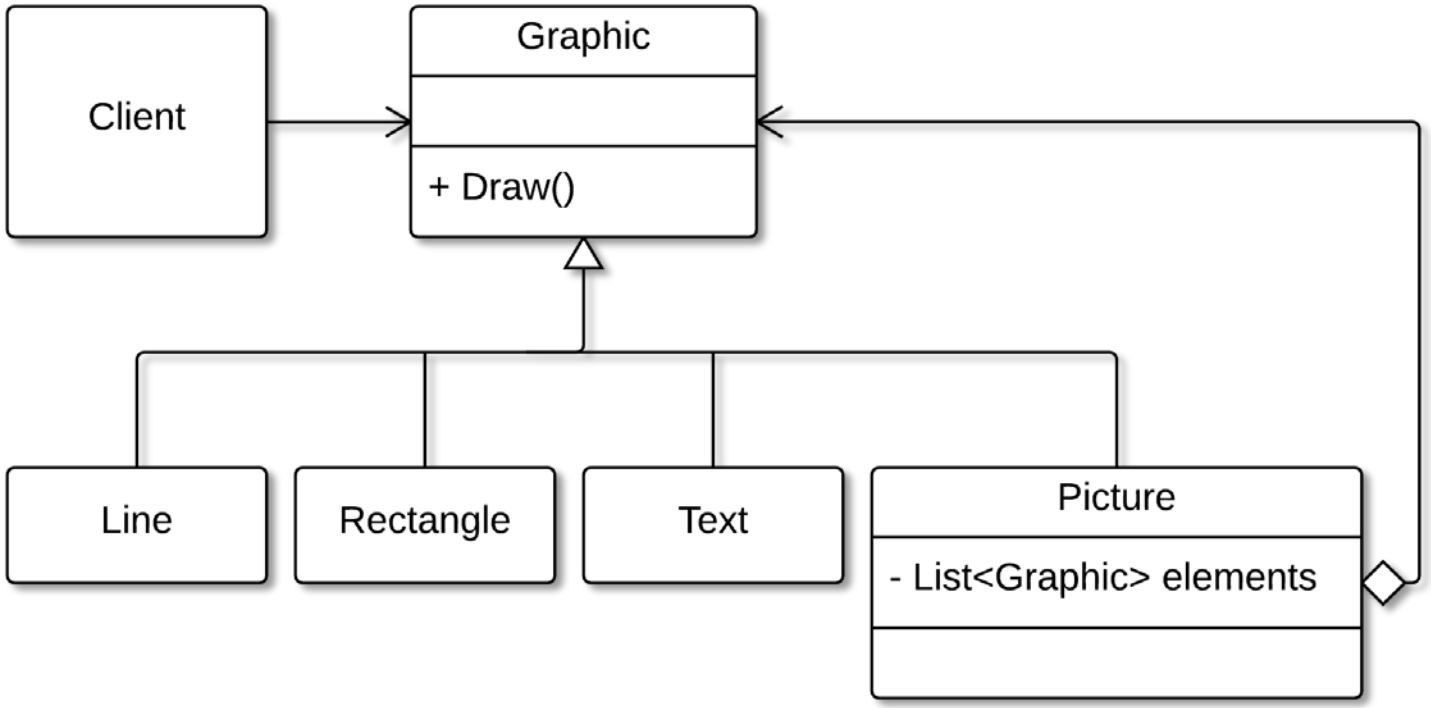
About the Composite Design Pattern

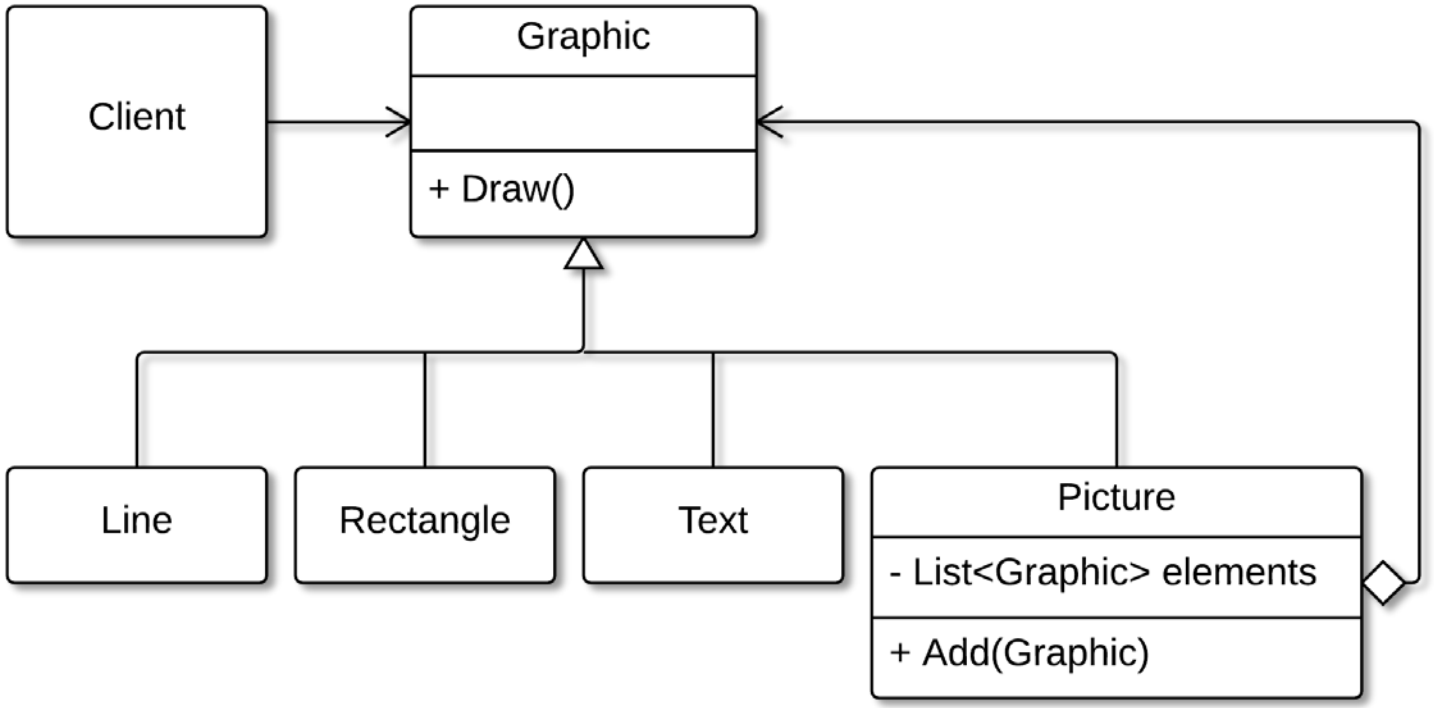


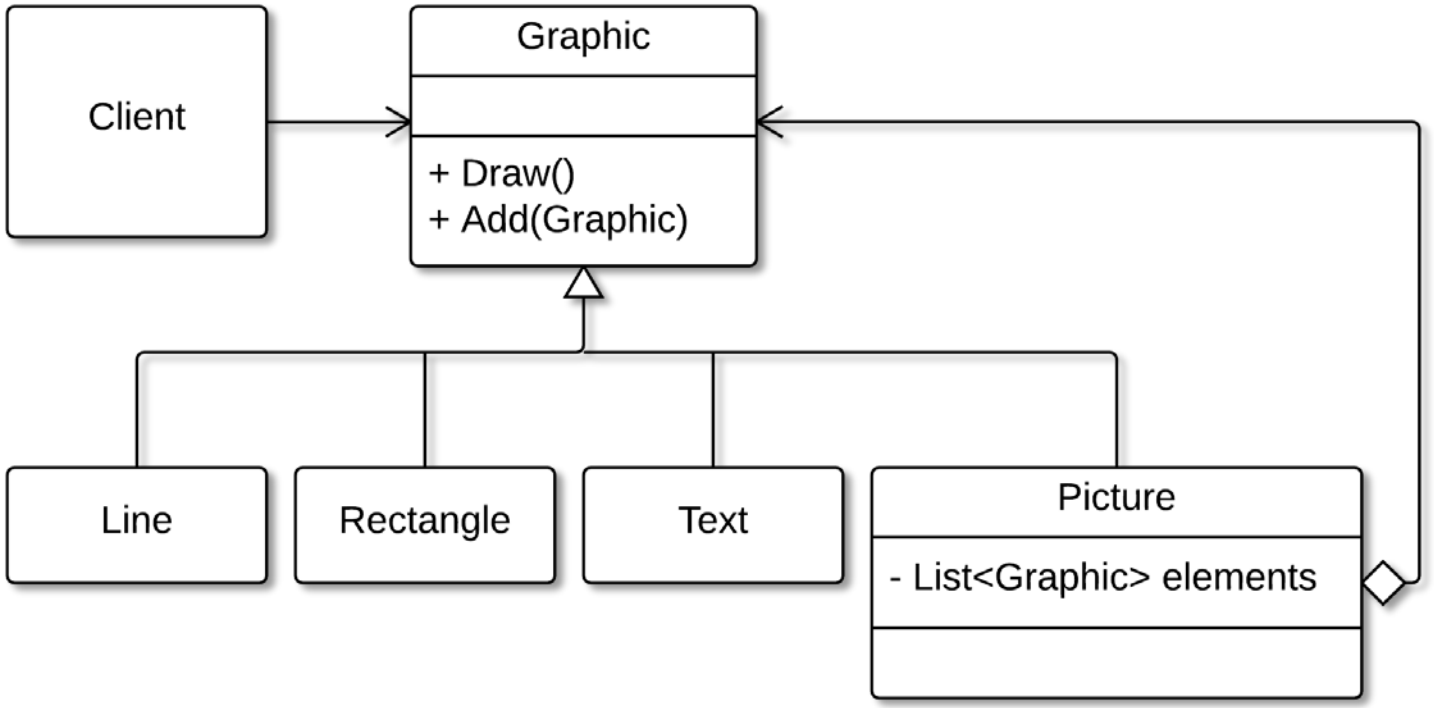
About the Composite Design Pattern

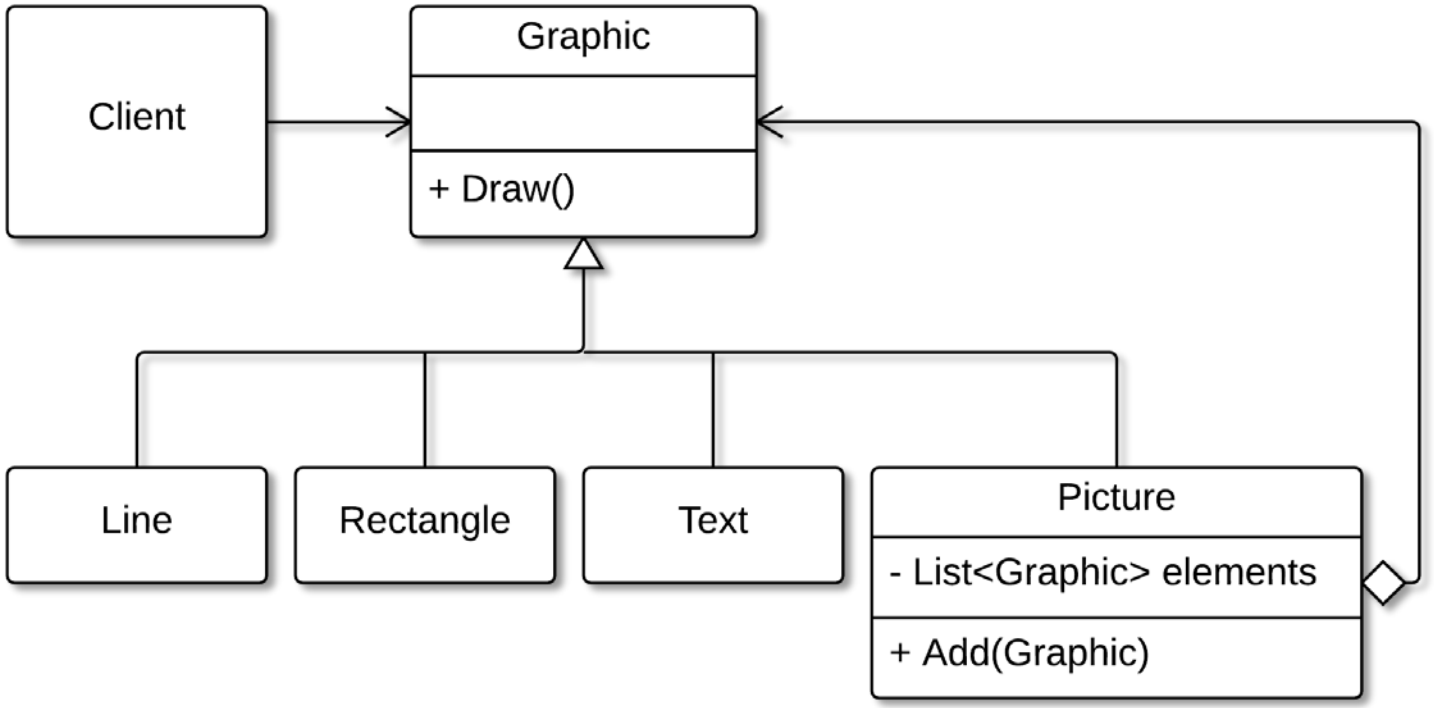


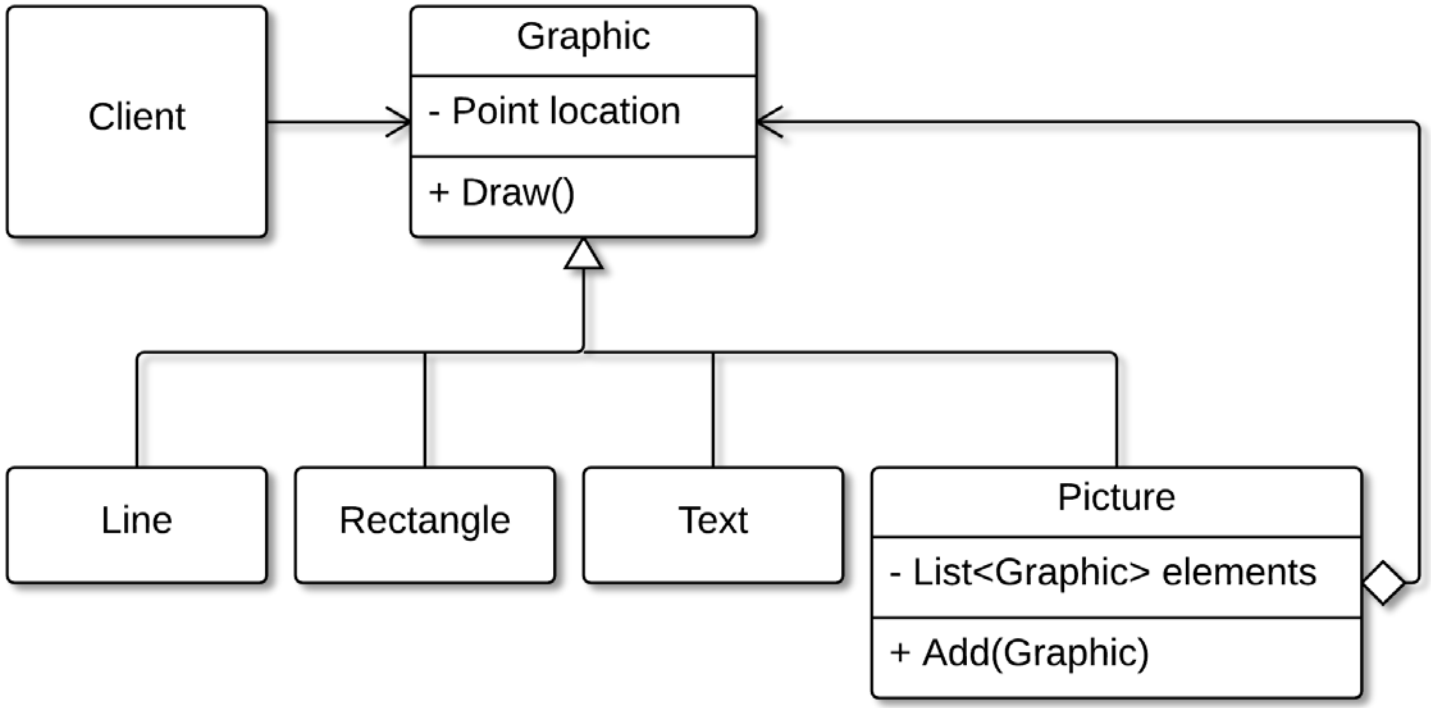


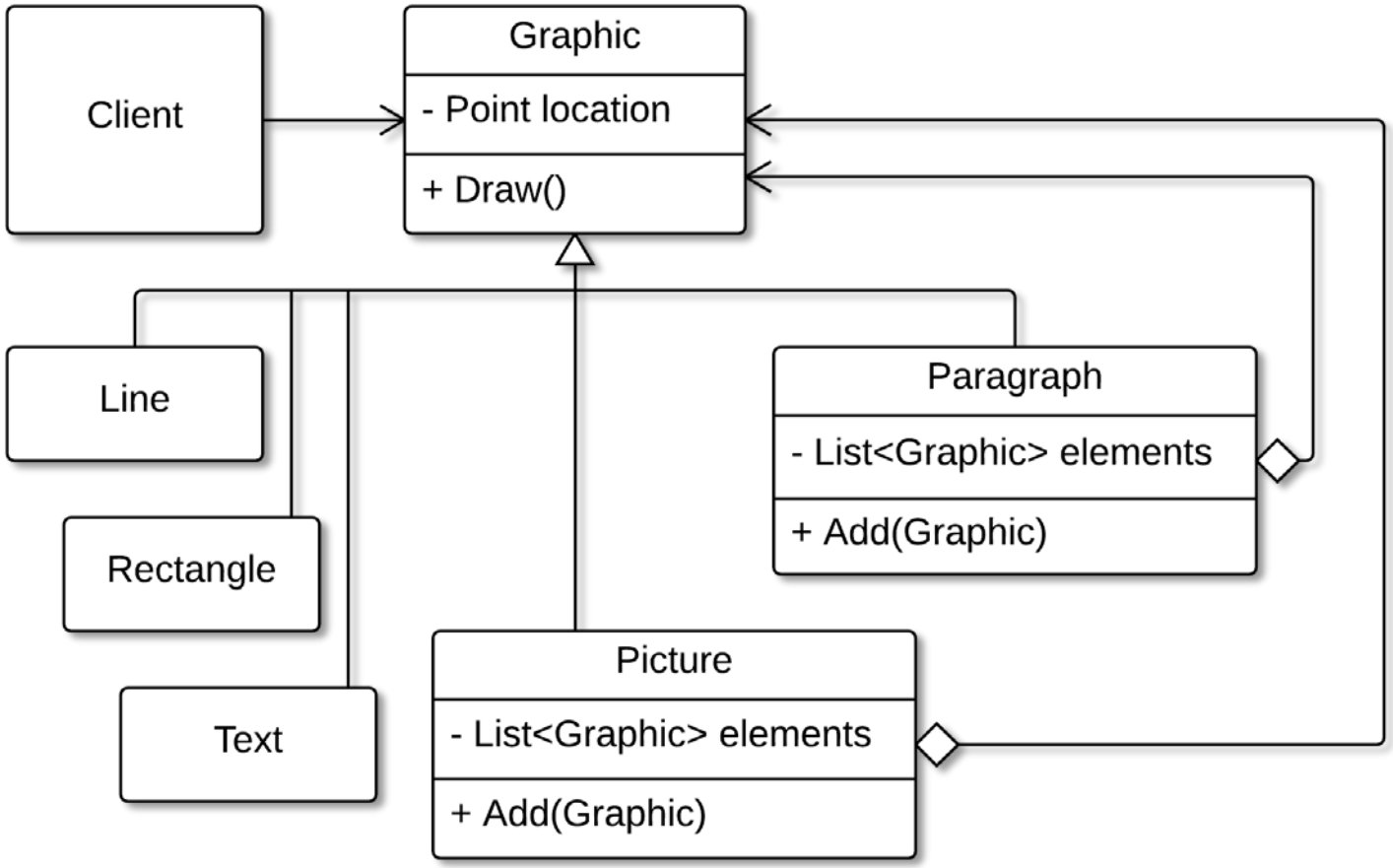


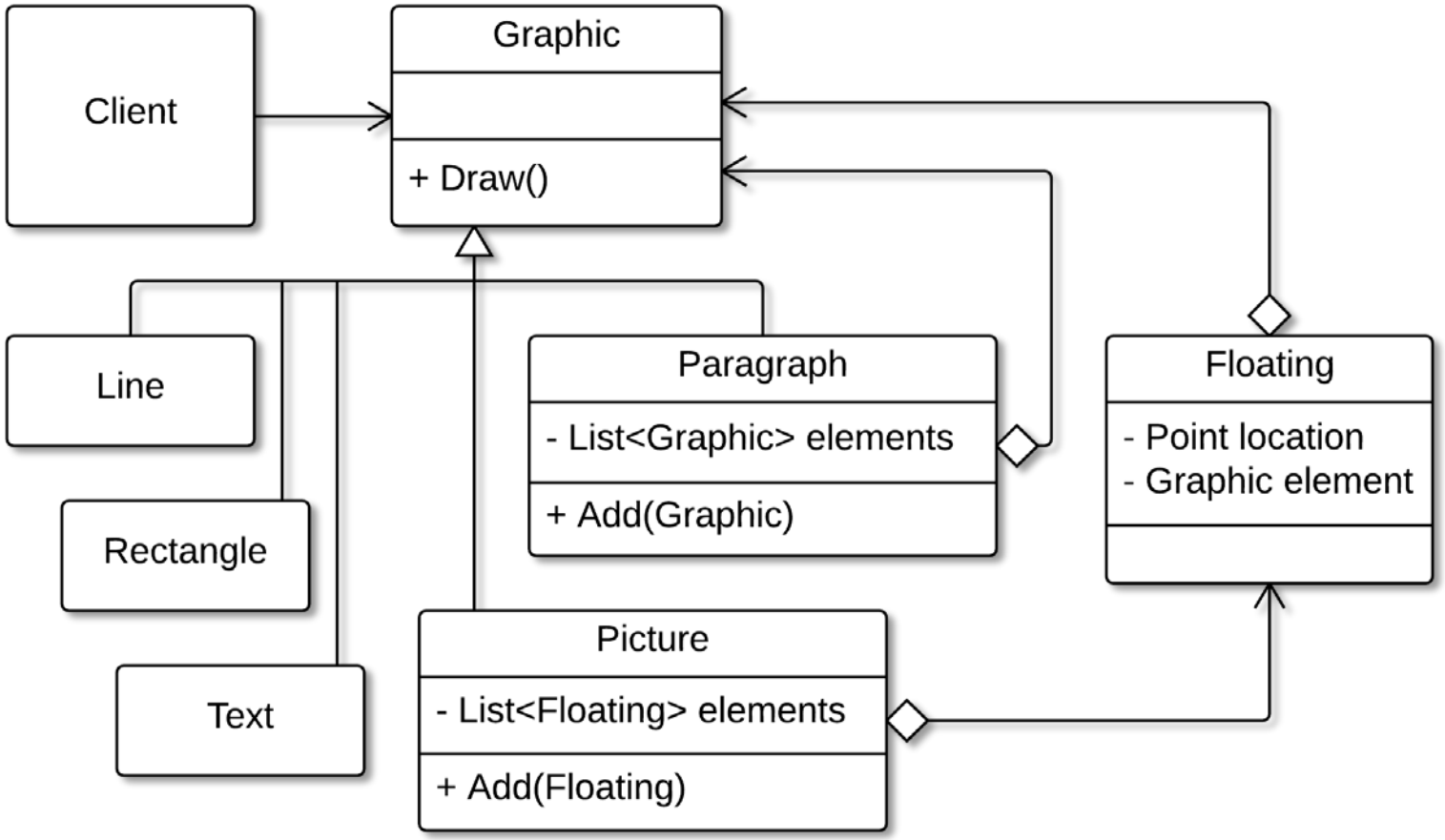


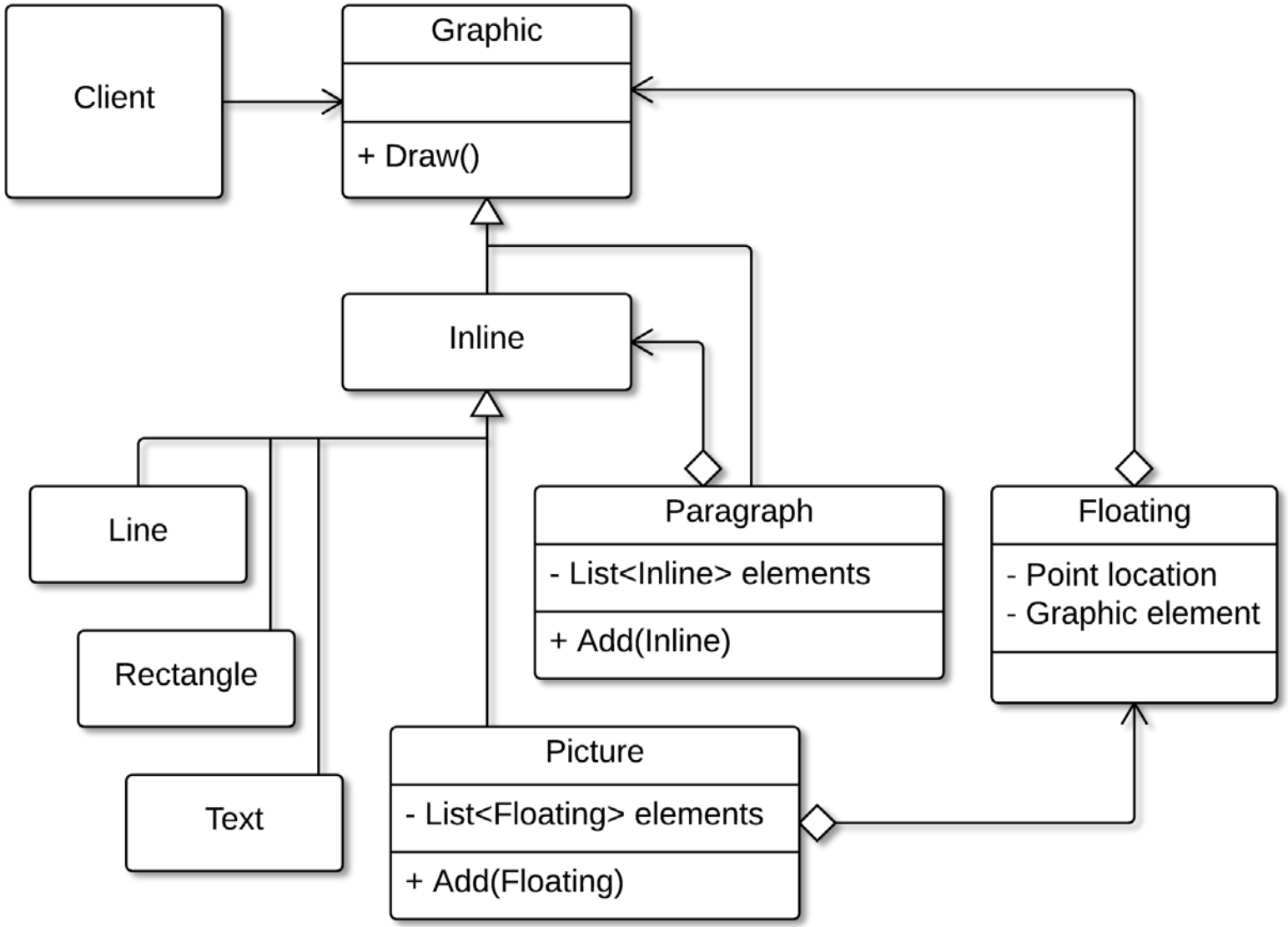


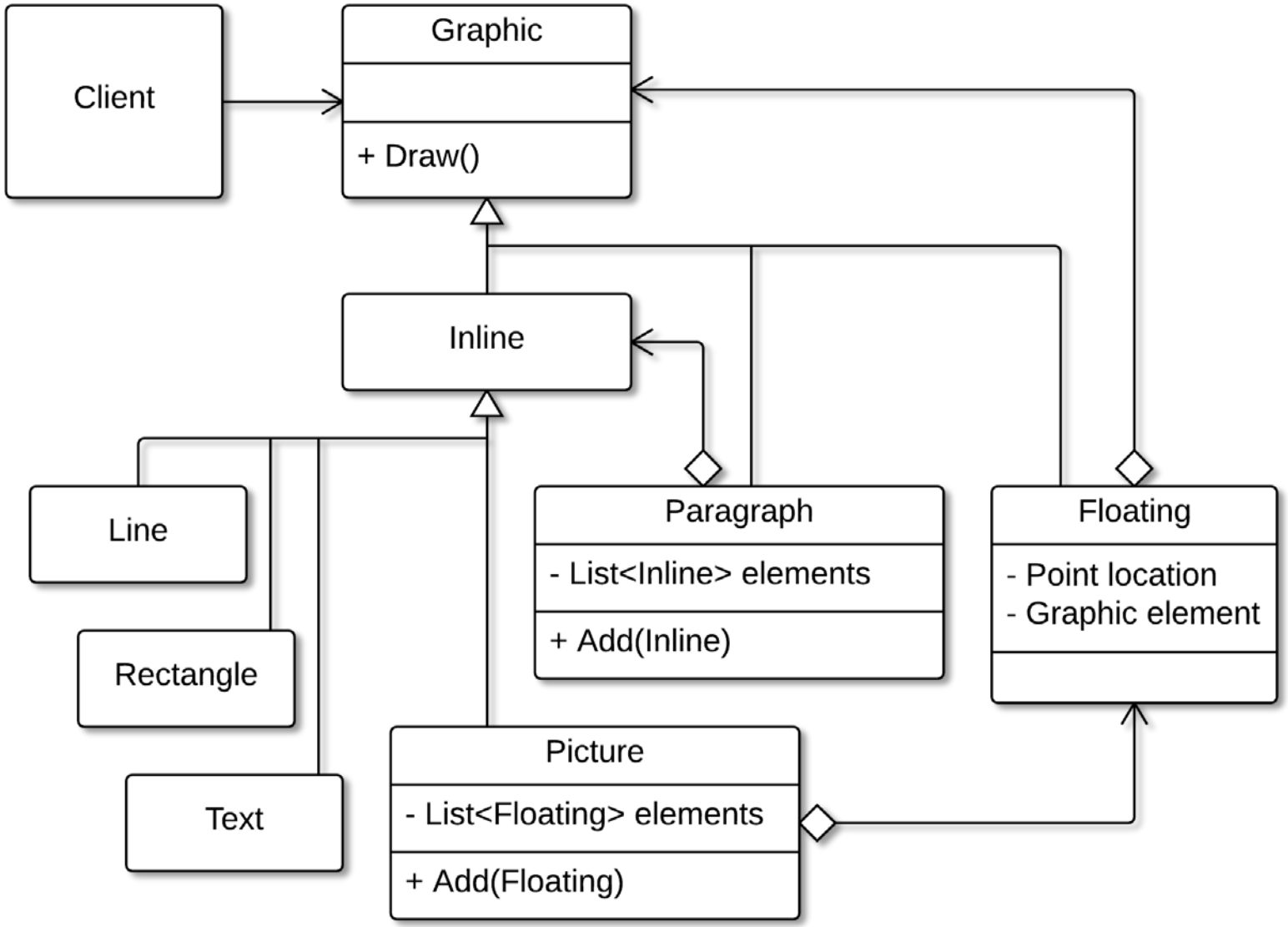


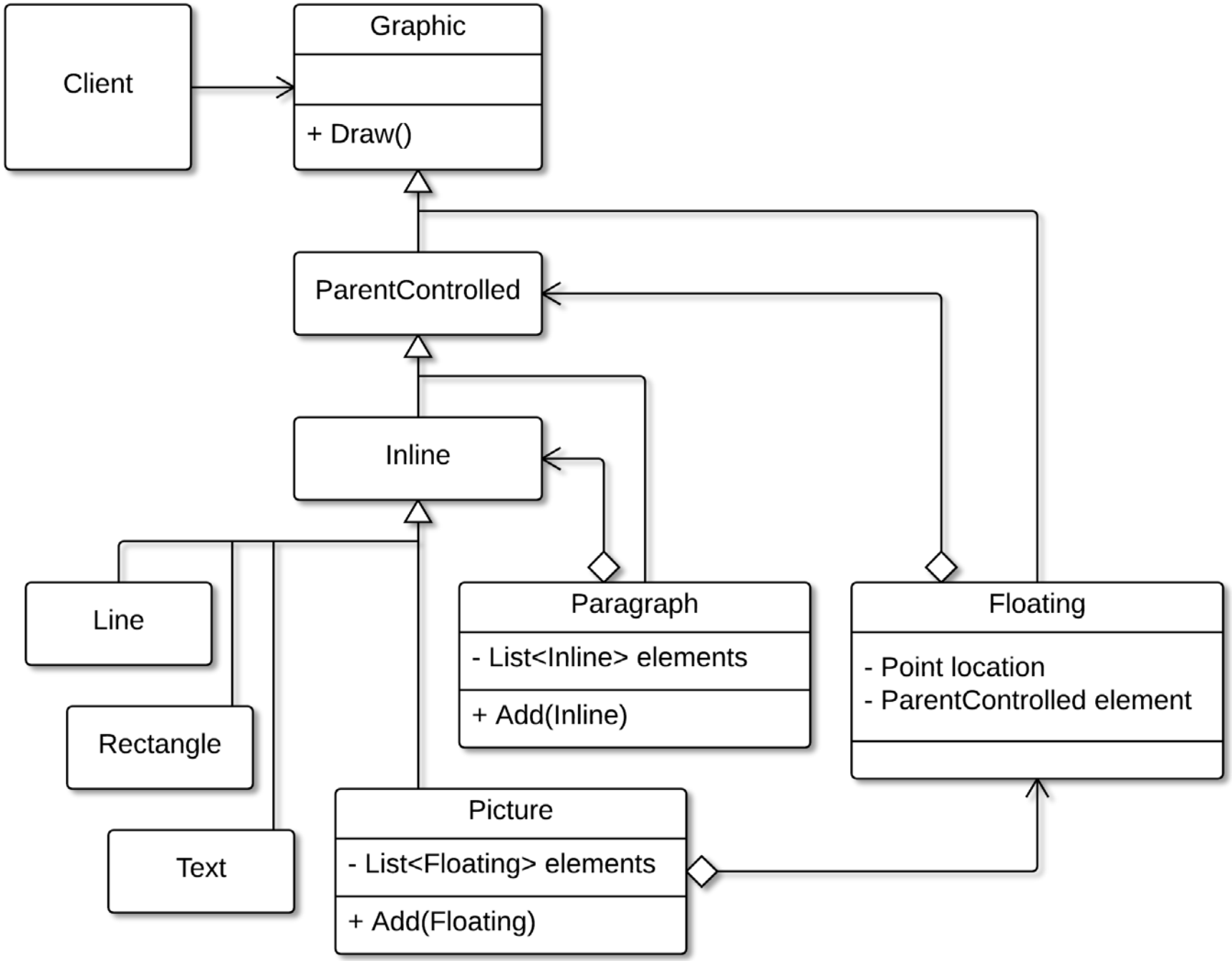


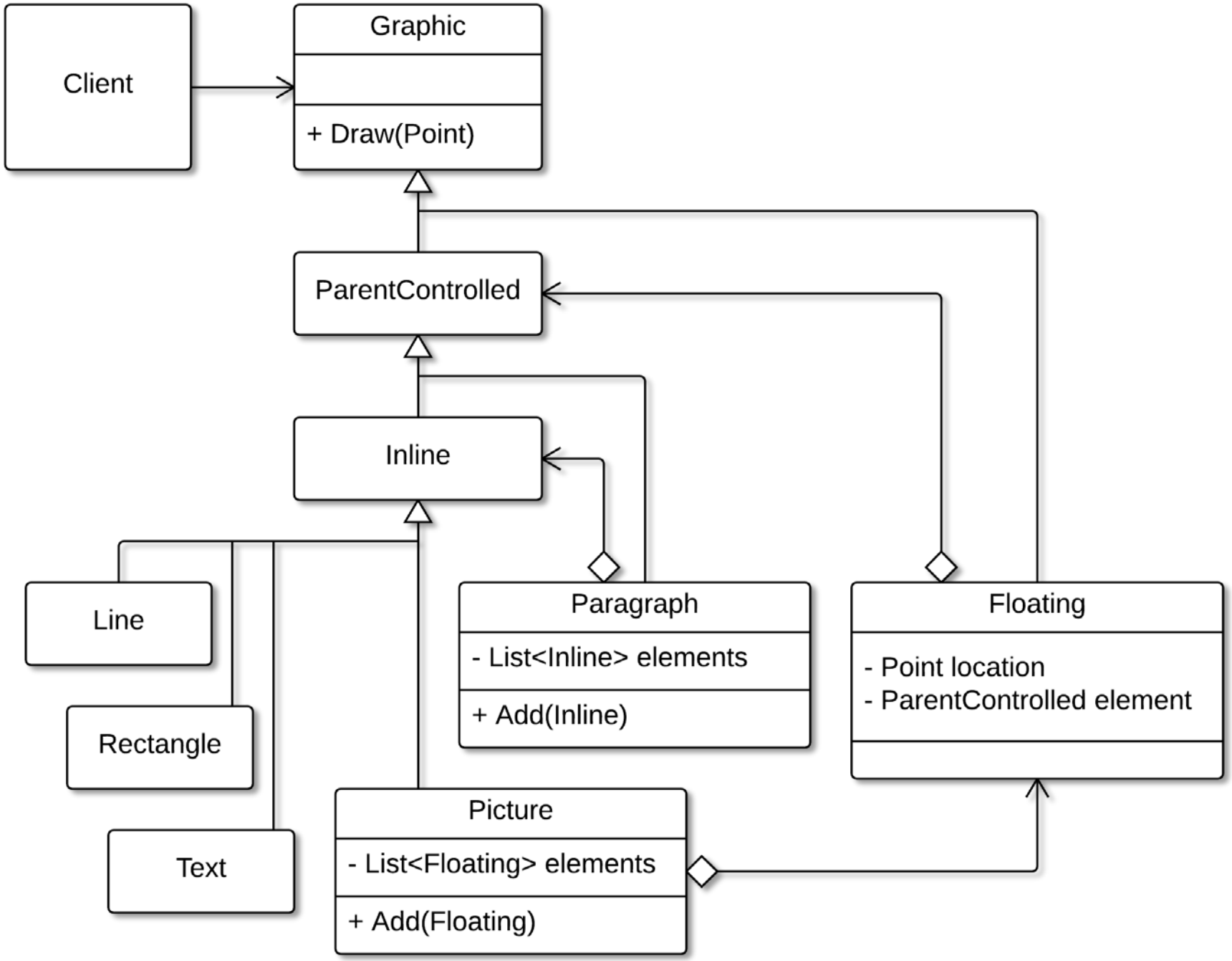


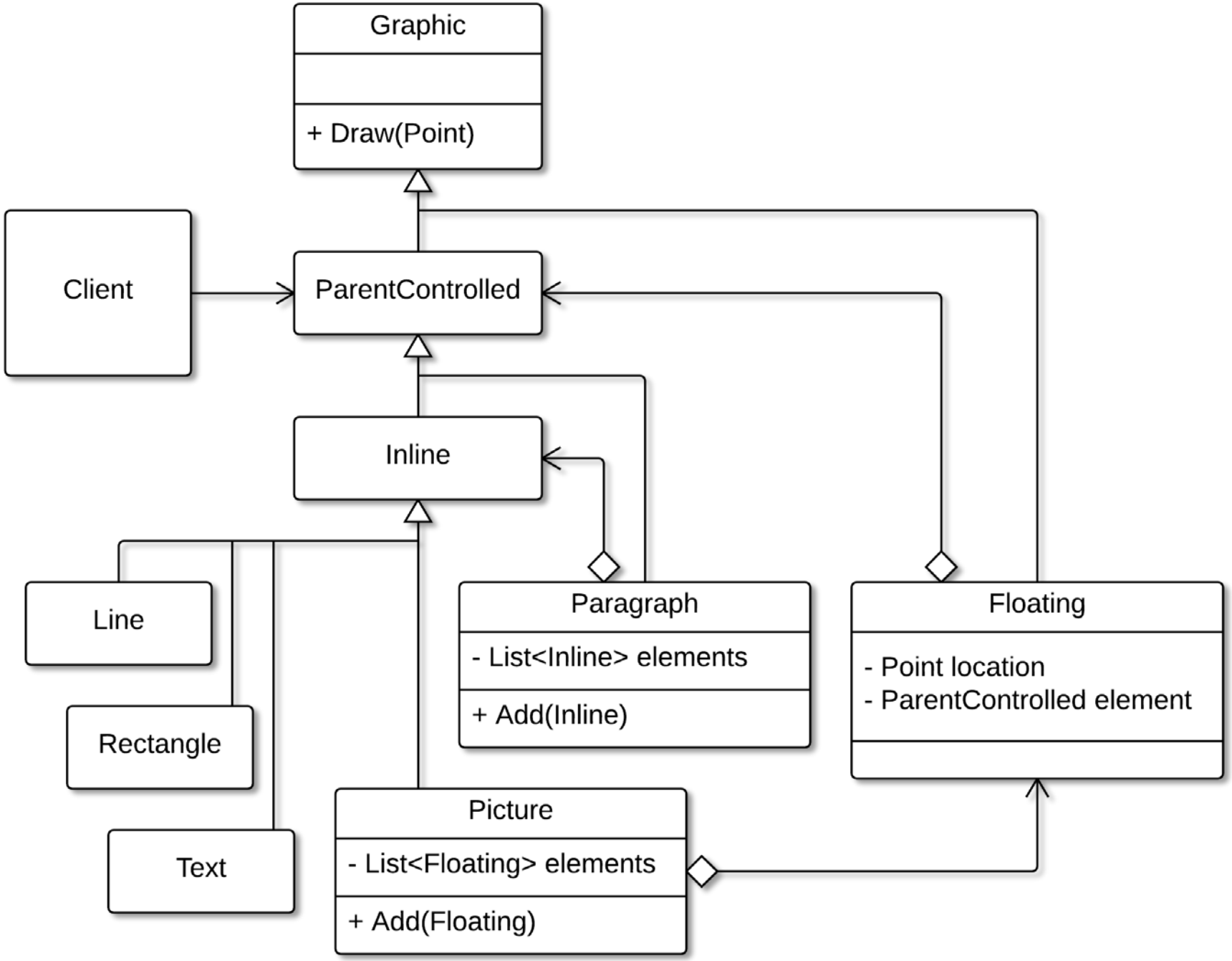


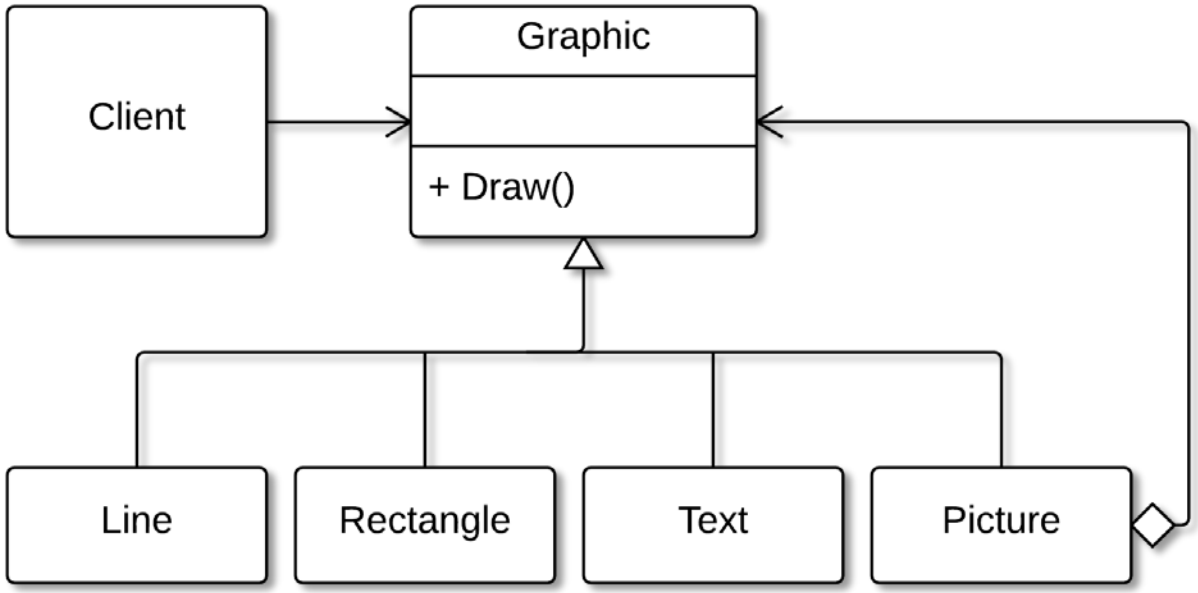


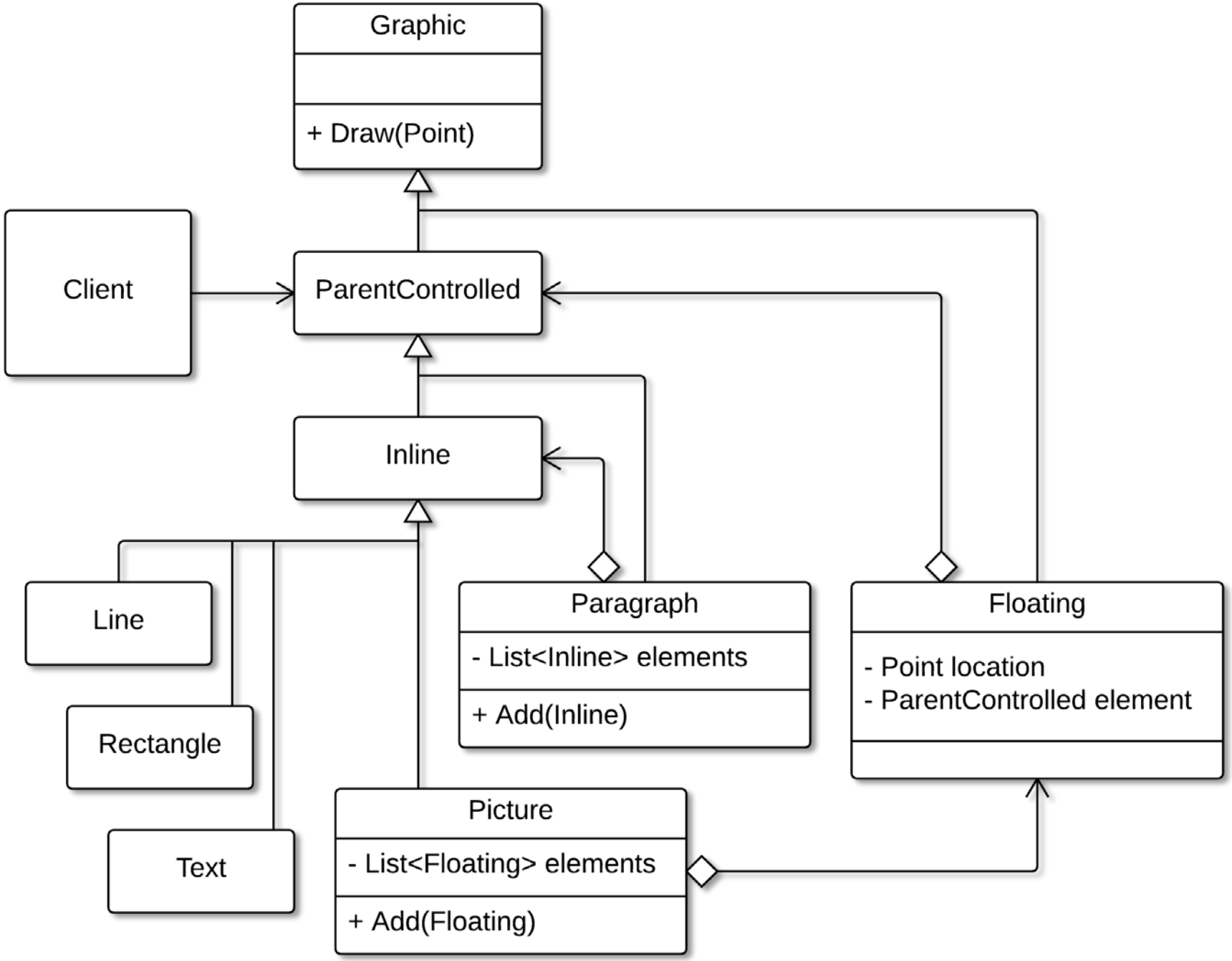




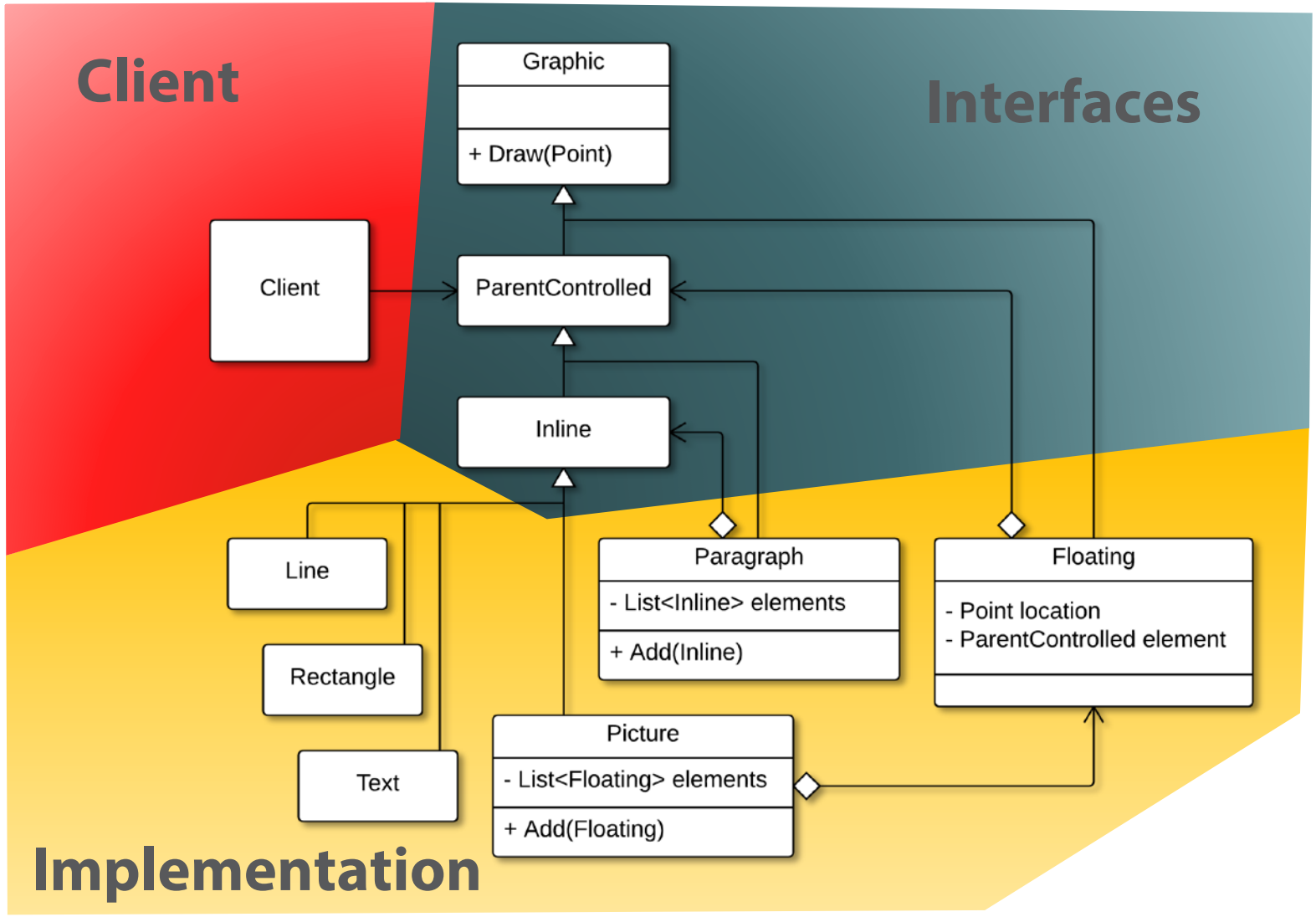








Abstraction vs. Implementation



Summary

- Example from the Gang of Four book
  - Class diagram provided as an example is over-simplified
  - Really applicable solution requires much larger number of classes
- Larger set of classes requires orchestration
  - It is the composite element that orchestrates execution
- Composite design pattern was not the goal of the exercise
  - The goal was to let the client control elements that are sometimes complex
  - Composite design pattern came to the rescue
  - Implementation was tailored to particular client's needs

Summary

- When was the best time to apply the Composite design pattern?
  - When we know what the client wants as the result
- Principal designs are not the same as those we implement
  - General solution needs to be adapted to particular project
  - Principal designs ignore real-world problems
- Real projects cannot ignore real-world problems
  - Application of a design pattern is often overly complicated
  - This may lead to new ideas that will simplify the design
    - A different, supporting design pattern may come to the rescue