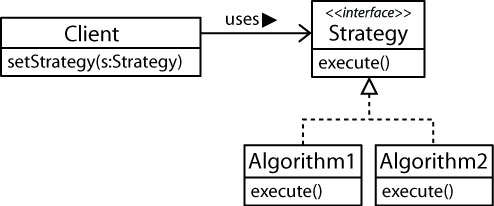
**Chapter 1 Notes – Strategy Pattern**

**Definition**

1. Intent: Define a family of algorithms, encapsulate each one, and make them interchangeable.

* By associating the client with a strategy/algorithm we allow different objects to use different algorithms and to swap algorithms at run-time.

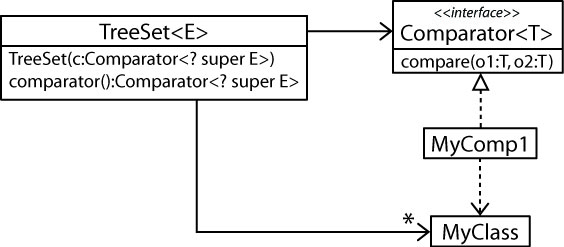


1. Design Principles:
2. Identify the aspects of your application that vary and separate them from what stays the same.
3. Program to an interface, not an implementation.
4. Favor composition over inheritance.

Describe how each of these principles is illustrated in the strategy pattern.

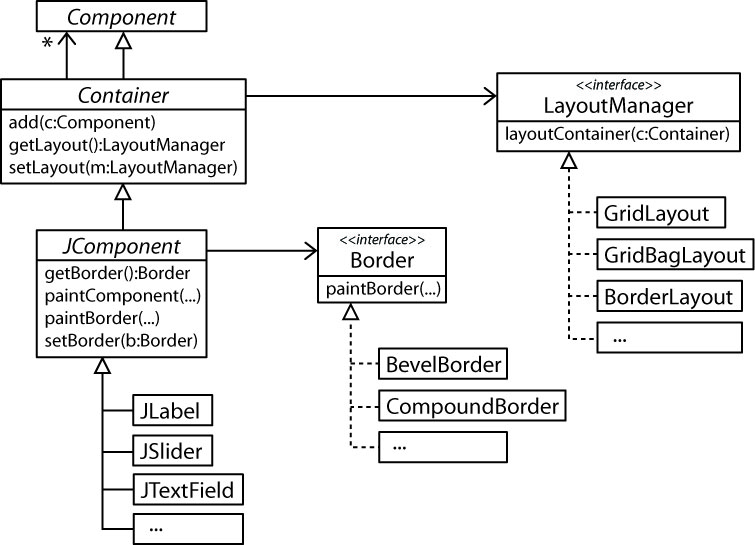
**Strategy in Java**

1. The *TreeSet* constructor can take a Comparator.

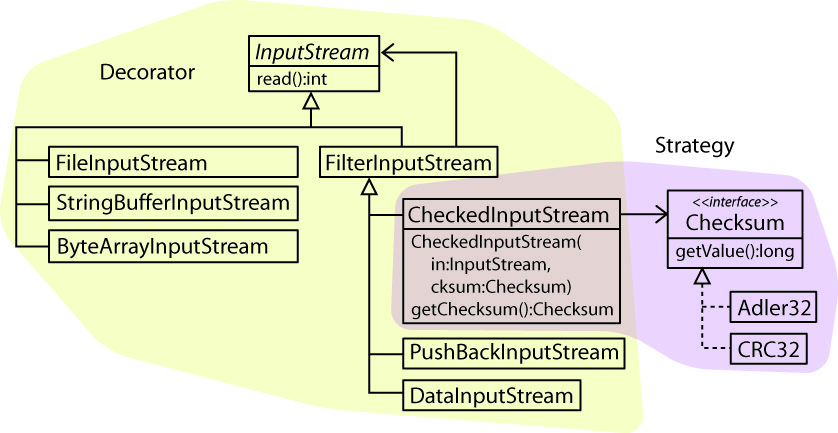


*TreeMap*, and *Collections.sort* are similar.

1. The *JComponent* class utilizes strategy with the *Border* interface while the *Container* class utilizes strategy with the *LayoutManager* interface. *JComponent* utilizes the strategy pattern in a number of other places as well.



1. A checksum is a value that can be used to verify the integrity of data and there are many different ways to calculate a checksum. Thus, the *CheckedInputStream* class utilizes the strategy pattern to associate a checksum algorithm (*Adler32* or *CRC32*) with the input stream. As we will see in Chapter 3, *CheckedInputStream* is a Decorator.



**StackExchange – Software Engineering**

1. SMS & Email Password Reset - <https://softwareengineering.stackexchange.com/questions/306836/strategy-pattern-implementation>
2. Strategy has strategies - <https://softwareengineering.stackexchange.com/questions/326179/implementing-strategy-pattern-multiple-variables>

**Other Examples**

1. Almost any type of simulation/game. For example a basketball player could have an offensive strategy (set, motion, zone), defensive strategy (man-to-man, 2-3 zone, 1-3-1 zone), and maybe even a bench strategy (sullen, rowdy, poised).
2. Chess – Pawn, Bishop, Rook, etc move differently.
3. Animals in a simulation may have a number of strategies: forage, hunt, rest, sleep, eat
4. Overtime pay – Different types of workers (Hourly, Salaried, Manager, etc) are paid overtime differently.
5. Online shopping – A shopping website suggests products in a sidebar, perhaps depending on the type of user: anonymous or registered. Suggestions could be: random, best sellers, clearance, sales items, new products, etc.
6. Online shopping – tax calculations for purchase orders depend on the state the purchaser is in.
7. Online shopping – Different types of coupons have different discounts.
8. File formats – read (or write) files in different formats.
9. A date class that can format dates in numerous ways, including a custom format.
10. A graph can be drawn as bar graph, line graph, pie chart, etc.
11. A component to generate SQL statements for different databases (Oracle, SQL Server, MySQL)
12. Compress files with different algorithms.
13. Drawing graphics with different algorithms (anti-alias, direct, others)
14. Forecasting demand (or anything) using different techniques.
15. Line-breaking algorithms.