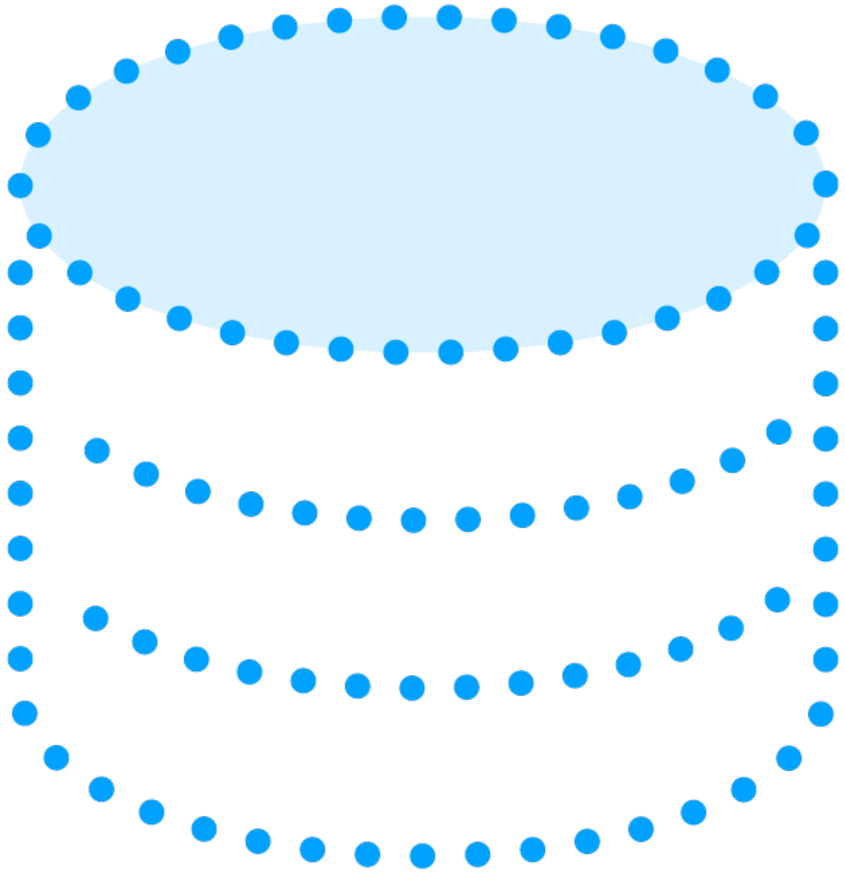
Flyweight Design Pattern

Bryan Hansen

@bh5k

Flyweight



Concepts

Efficient memory usage

Large number of similar objects

Immutable

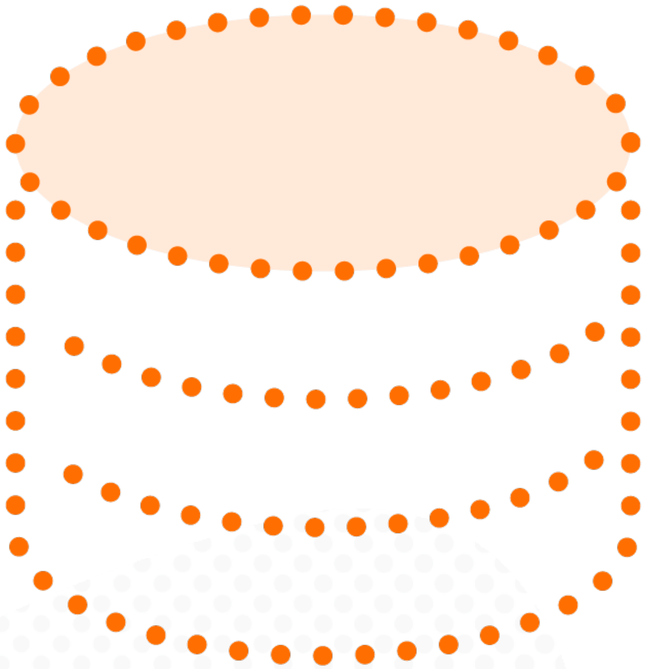
Extrinsic object state

Examples:

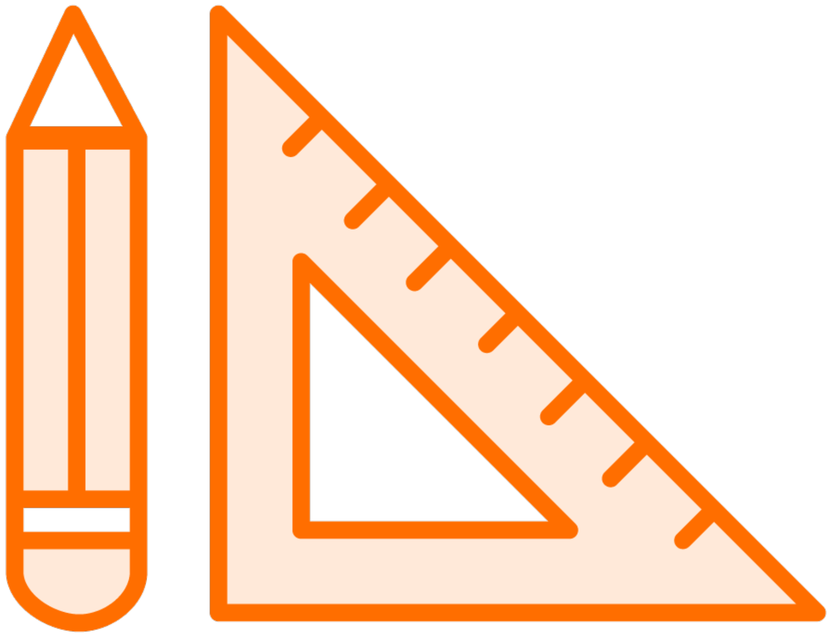
java.lang.String

java.lang.Integer#valueOf(int)

Boolean, Byte, Character, Short, Long



Design



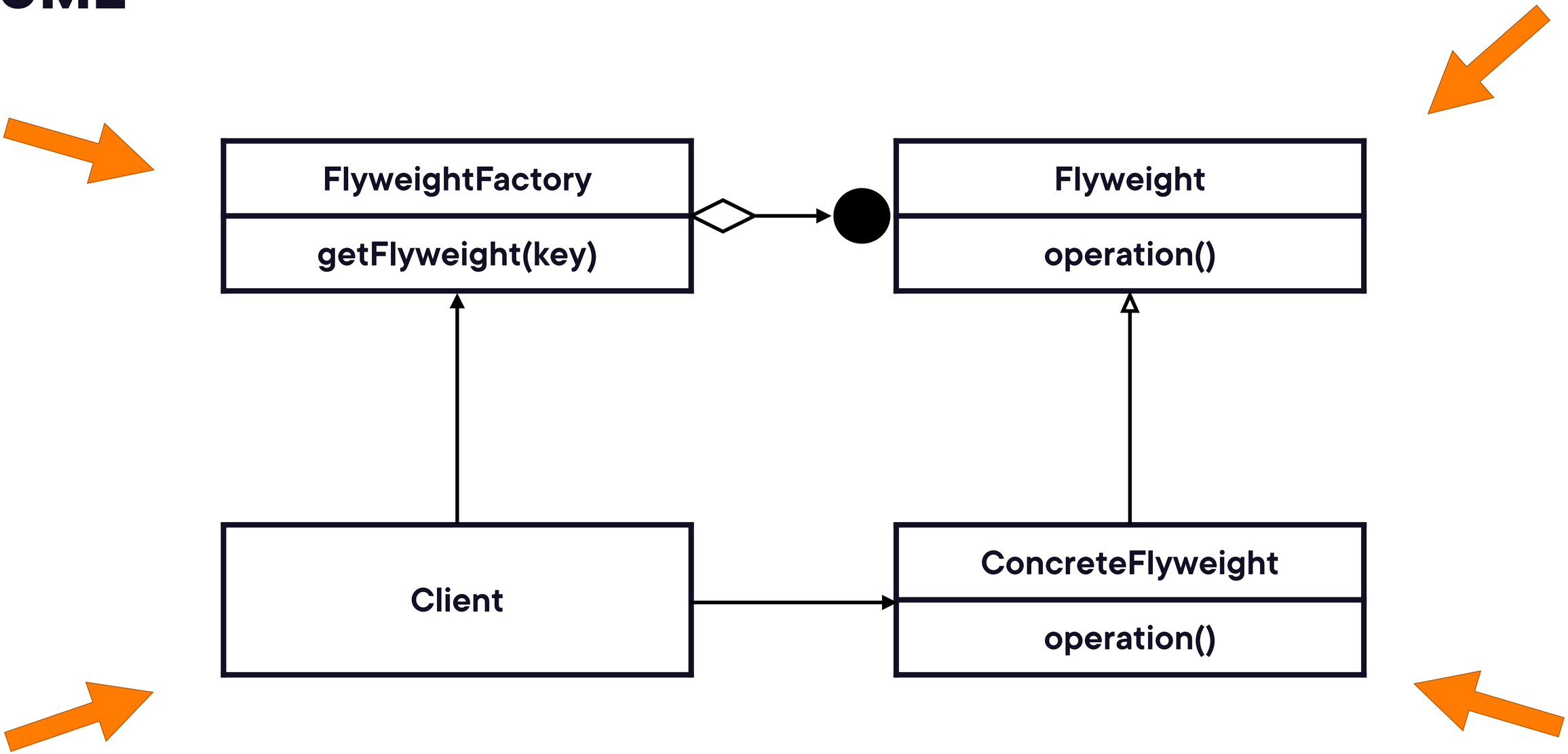
Pattern of patterns

Utilizes a Factory

Encompasses Creation and Structure

Client, Factory, Flyweight, ConcreteFlyweight

UML


FlyweightFactory
getFlyweight(key)
Flyweight
operation()
Client
ConcreteFlyweight
operation()

Everyday Example - Integer

```
Integer firstInt = Integer.valueOf(5);

Integer secondInt = Integer.valueOf(5);

Integer thirdInt = Integer.valueOf(10);

System.out.println(System.identityHashCode(firstInt));
System.out.println(System.identityHashCode(secondInt));
System.out.println(System.identityHashCode(thirdInt));
```

Exercise Flyweight

Inventory Management System

Client, Catalog, Order, Item

Pitfalls

Complex pattern

Premature optimization

Must understand Factory

Not a graphical pattern

Contrast

Flyweight	VS	Facade
Memory Optimization		Refactoring pattern
Optimization Pattern		Simplified Client
Immutable Objects		Provides a different interface

Flyweight Summary

Great for Memory Management

A little bit complex

Used frequently by core API