

# Flyweight DP

4-Feb-20 11:29 PM

256

## Which is better?

- Point1
- or
- Point2

Code is in Flyweight

4-Feb-20 11:29 PM

257

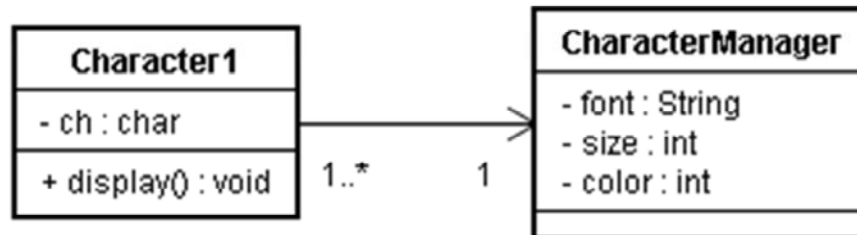
Code available in Java, C# and C++

## Problem

- We have a Character class. It stores information about one character.
  - Our document has a large number of characters.
  - Each character can have a different font and size.
  - How to reduce the memory requirements and runtime overheads?



## Solution



4-Feb-20 11:29 PM

259

Designing objects down to the lowest levels of system "granularity" provides optimal flexibility, but can be unacceptably expensive in terms of performance and memory usage.

Flyweight performs "use sharing" to support large number of fine-grained objects efficiently.

A flyweight is a shared object that can be used in multiple contexts simultaneously.

=====

The Unicode of the characters can be stored in an array.

Many of the characters have the same attributes e.g. font, font size, color, etc. We can store these common attributes for a set of characters only once.

## Usage of Flyweight DP

- String in Java
- Wrapper Classes in Java
- TimerTask
- Should Dimension be Flyweight?

4-Feb-20 11:29 PM

260

Java uses the Flyweight pattern to manage the String objects used to represent string literals.

All wrapper classes are Flyweight

TimerTask

Dimension class should have been a Flyweight

## When to apply Flyweight?

- Always.
  - Unless you have a very good reason, make your class immutable.
  - Getters should normally return immutable objects.
    - Use `java.util.collections.unmodifiableList`

This rule is applicable only if the language has automatic garbage collection.

4-Feb-20 11:29 PM

261

If you cannot make the entire class immutable, make maximum part immutable.  
Limit mutability as much as you can.  
The less mutable the better.  
Final is the new private.

For C++, `shared_ptr` will likely be needed to share immutable values.

## Assignment

- We have many user accounts. Permissions for each user are defined at group level.
  - A user can be in multiple groups.
  - Where can we use the flyweight pattern?

4-Feb-20 11:29 PM



262

OOAD classes: Different classes (Immutable), Prefer Value Objects, Rules for Concurrency,

## How many Flyweights?

- Java 8 has a new package `java.time`
  - What percentage of classes are `Immutable`?
  - If a mutable class exists, then justify its design.



## N-Queen problem

- On a  $N$  by  $N$  chess board, place  $N$  queens in such a manner that no queen threatens any other.
- Print how many different solutions are possible for  $N=8$ .

4-Feb-20 11:29 PM

264

Answer 92.

Flyweight will be used for Cell class

## Assignment

- We want to cache immutable objects.
- Create a LRU cache for immutable objects
  - Hint: Use `removeEldestEntry` function in `LinkedHashMap`

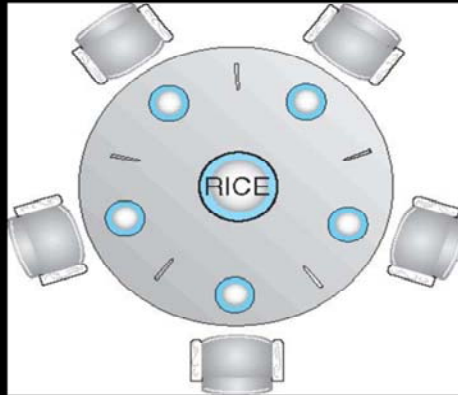
4-Feb-20 11:29 PM

265

Java Solution in Patterns project FlyweightCache.

# Danger of Mutability

- Simulate: Dining Philosophers Problem



4-Feb-20 11:29 PM

266

Philosophers spend their lives thinking and eating

Don't interact with their neighbors, occasionally try to pick up 2 chopsticks (one at a time) to eat from bowl

Need both to eat, then release both when done

In the case of 5 philosophers

Shared data

Bowl of rice (data set)

Semaphore **chopstick** [5] initialized to 1

The structure of Philosopher *i*:

```
do {  
    wait (chopstick[i] );  
    wait (chopstick[ (i + 1) % 5] );  
  
    // eat  
  
    signal (chopstick[i] );  
    signal (chopstick[ (i + 1) % 5] );  
}
```

```
        //  think  
  
    } while (TRUE);
```

What is the problem with this algorithm?

The last philosopher must take spoons in reverse order, otherwise there is a deadlock.

Moral: Rice is used only one thread. So no locking. Spoon is like a mutable object; only one thread can use it.

A large application will have some mutable objects shared. First try to make it immutable. If that is not possible, then ensure locks are acquired in same order.