Flyweight

Use sharing to support large number of fine-grained objects efficiently
A document has many instances of the character 'a'

Character has
  Font
  width
  Height
  Ascenders
  Descenders
  Where it is in the document

Most of these are the same for all instances of 'a'

Use one object to represent all instances of 'a'
public void testInterned() {
    String a1 = "catrat";
    String a2 = "cat";
    assertFalse(a1 == (a2+ "rat");

    String a3 = (a2 + "rat").intern();
    assertTrue(a1 == a3);
    String a4 = "cat" + "rat";
    assertTrue(a1 == a4);
    assertTrue(a3 == a4);
}

    public String intern()
    {
        Returns a canonical representation for the string object.
        A pool of strings, initially empty, is maintained privately by the class String.
Intrinsic State

Information that is independent from the object's context

The information that can be shared among many objects

So can be stored inside of the flyweight
Extrinsic State

Information that is dependent on the object's context

The information that can not be shared among objects

So has to be stored outside of the flyweight
Client

FlyweightFactory

getFlyweight(key)

Flyweight

operation(extrinsicState)

ConcreteFlyweight

intrinsicState

operation(extrinsicState)

UnsharedConcreteFlyweight

allState

Operation

flyweight

if (flyweight[key] exists)
    return existing flyweight
else
    create new flyweight
    add it to flyweight pool
    return new flyweight
The Hard Part

Separating state from the flyweight

How easy is it to identify and remove extrinsic state

Will it save space to remove extrinsic state
Example Text

Run Arrays

aaaaabaaaaaaaaaaaaaaaaaaaa

↑

↓

a b a

5 1 20
Text Example

Lexi Document Editor

Uses character objects with font information
(To support graphic elements)

"A Cat in the hat came back the very next day"

Use run array to store font information (extrinsic state)

Normal Bold Normal

22 4 18