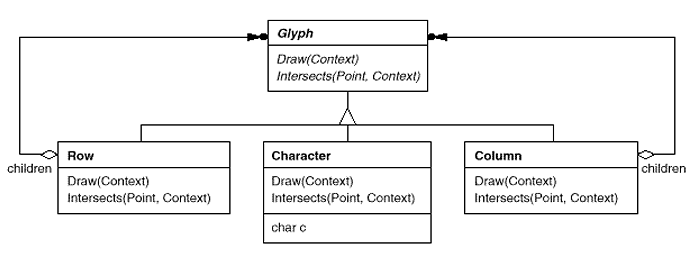
Flyweight Design Pattern

GOF : **Use sharing to support large numbers of fine-grained objects efficiently.**

# **Class Diagram**



As suggested by Joshua Bloch, you can achieve Flyweight Design using static factory method. Java language

uses Boolean where the **valueOf()** method properly uses Flyweight Design and returns the similar Boolean object

without creating repeatedly. For the convenience, the code structure is given below.

public final class Boolean implements java.io.Serializable,Comparable<Boolean>

{

public static final Boolean TRUE = new Boolean(true);

public static final Boolean FALSE = new Boolean(false);

// Use of Flyweight Design Pattern

public static Boolean valueOf(boolean b)

{

return (b ? TRUE : FALSE);

}

}

The class **java.util.EnumSet (Item 32),** introduced in release 1.5, has no public constructors, only static factories. They return one of two implementations, depending on the size of the underlying enum type: if it has sixty-four or fewer elements, as most enum types do, the static factories return a **RegularEnumSet** instance, which is backed by a single long; if the enum type has sixty-five or more elements, the factories return a **JumboEnumSet** instance, backed by a long array.

# **FlyWeight Pattern Example**

Let us consider we have three kinds of Cards like MASTER,VISA,ELECTRON. We have to choose the card based on the bin of the card without recreating the objects.

public final class Card

{

private String bin = null;

private final static Card MASTER = new Card("1111");

private final static Card VISA = new Card("2222");

private final static Card ELECTRON = new Card("3333");

private final static Card INVALID = new Card("4444");

private Card(String bin)

{

this.bin = bin;

}

public static Card valueOf( String cardNo )

{

Card card = null;

if( cardNo.startsWith("1111")) card = MASTER;

else if( cardNo.startsWith("2222")) card = VISA;

else if( cardNo.startsWith("3333")) card = ELECTRON;

else card = INVALID;

return card;

}

public String getTransactionDetails()

{

return "Transaction Details for "+toString();

}

@Override

public String toString()

{

String cardType = null;

if( bin.startsWith("1111")) cardType = "MASTER";

else if( bin.startsWith("2222")) cardType = "VISA";

else if( bin.startsWith("3333")) cardType = "ELECTRON";

else cardType = "INVALID";

return cardType;

}

}

package com.type2;

public class TestCard

{

public static void main(String[] args)

{

String cardNo = "3333 1111 2222 4444";

Card card = Card.valueOf(cardNo);

System.out.println("Card Type ::: "+card.toString());

System.out.println(card.getTransactionDetails());

}

}