

Enumeraciones

Estudiaremos superficialmente este tema con un ejemplo.

En Java disponemos, además de clases e interfaces, de “enumeraciones”, que son clases de las que puede instanciarse un conjunto predefinido de objetos.

```
import java.util.*;

public class Card {
    public enum Rank { DEUCE, THREE, FOUR, FIVE, SIX,
        SEVEN, EIGHT, NINE, TEN, JACK, QUEEN, KING, ACE }

    public enum Suit { CLUBS, DIAMONDS, HEARTS, SPADES }

    private final Rank rank;
    private final Suit suit;
    private Card(Rank rank, Suit suit) {
        this.rank = rank;
        this.suit = suit;
    }

    public Rank rank() { return rank; }
    public Suit suit() { return suit; }
    public String toString() { return rank + " of " + suit; }

    private static final List<Card> protoDeck = new ArrayList<Card>();

    // Initialize prototype deck
    static {
        for (Suit suit : Suit.values())
            for (Rank rank : Rank.values())
                protoDeck.add(new Card(rank, suit));
    }

    public static ArrayList<Card> newDeck() {
        return new ArrayList<Card>(protoDeck); // Return copy of prototype deck
    }
}
```

(el ejemplo muestra las enumeraciones como meras listas de identificadores, pero son realmente objetos y su definición puede “complicarse” considerablemente. Pueden estudiarse en la [documentación de Sun.](#))

Palabras reservadas en Java				
abstract	assert ^(1,4)	boolean	break	byte
case	catch	char	class	const*
continue	default	do	double	else
enum ^(5,0)	extends	final	finally	float
for	goto*	if	implements	import
instanceof	int	interface	long	native
new	package	private	protected	public
return	short	static	strictfp ^(1,2)	super
switch	synchronized	this	throw	throws
transient	try	void	volatile	while