

Se desea implementar un programa que maneje barajas de cartas. Una carta será una

clase que tendrá dos atributos, número (int) y palo (String).

Una baraja estará compuesta por un conjunto de cartas. Con una baraja se podrá hacer

una serie de operaciones:

- crearBaraja: será un método abstracto sin parámetros.

- barajar: cambia la posición de las cartas de forma aleatoria. Para ello utilizaremos

la instrucción Collections.shuffle(lista), siendo lista un objeto de la clase ArrayList

o LinkedList (se deberá importar java.util.Collections)

Trabajaremos con dos tipos de barajas, baraja española y baraja francesa.

La baraja española tendrá un atributo booleano que me indicará si quiero jugar con 8 y 9

en cuyo caso tendré un total de 48 cartas o sin 8 y 9 haciendo un total de 40 cartas (si

trabajamos sin 8 y 9 los números de las cartas de cada palo serán 1, 2, 3, 4, 5, 6, 7, 10, 11

y 12). Los palos de la baraja española serán OROS, COPAS, ESPADAS y BASTOS.

La baraja francesa no tendrá ningún atributo y el número de cartas para cada palo será de

13 haciendo un total de 52 cartas. Los palos serán DIAMANTES, PICAS,

CORAZONES y

TRÉBOLES.

A la hora de mostrar una carta por pantalla se indicará el número y el palo pero en el caso

de la baraja española en lugar de indicar los números 10, 11 y 12 los nombraremos como

Sota, Caballo y Rey respectivamente, y en el caso de la francesa los números 11, 12 y 13

serán Jota, Reina y Rey. En ambos casos el 1 se nombra como As.

El programa principal mostrará un menú con las siguientes opciones:

1. Nueva baraja: creará una nueva baraja preguntando al usuario el tipo de baraja y,

en el caso de la española, si debe incluir 8 y 9. Si hubiese una ya creada ésta será

reemplazada por la nueva.

2. Barajar: barajará las cartas de la baraja usando el método implementado en la

clase Baraja. En el caso de que no hubiera ninguna baraja creada lo indicará.

3. Mostrar baraja: mostrará todas las cartas de la baraja por pantalla de la forma que

se indica en el párrafo anterior. En el caso de que no hubiera ninguna baraja creada

lo indicará.

4. Imprimir baraja: guardará en un fichero de texto cuyo nombre indicará el usuario la

baraja en el mismo formato que en la opción anterior 3. En el caso de que no hubiera ninguna baraja creada lo indicará.

5. Salir

Al salir del programa se almacenará la situación actual del programa de forma que al

volver a iniciar el programa se recupere la baraja en el caso de que ésta ya hubiese sido

creada en una ejecución anterior.

Se deberá controlar todas las posibles excepciones.

## LECTOR.JAVA

```
package examen3ev22.pkg23;

import java.util.Scanner;

public class Lector {

    public static int leerInt()
    {
        Scanner sc = new Scanner(System.in);
        while (true)
        {
            try
            {
                return sc.nextInt();
            }
            catch (Exception e)
            {
                sc.nextLine();
                System.out.println("NÃºmero invÃ¡lido");
            }
        }
    }

    public static int leerInt(String msg)
    {
        Scanner sc = new Scanner(System.in);
        while (true)
        {
            try
            {
                System.out.print(msg);
                return sc.nextInt();
            }
            catch (Exception e)
            {
                sc.nextLine();
                System.out.println("NÃºmero invÃ¡lido");
            }
        }
    }

    public static String leerLinea()
    {
        Scanner sc = new Scanner(System.in);
        return sc.nextLine();
    }

    public static String leerLinea(String msg)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print(msg);
        return sc.nextLine();
    }
}
```

MAIN.JAVA

```
package examen3ev22.pkg23;

import java.io.*;

public class Examen3Ev2223 {

    public static Baraja baraja = null;

    public static String fBaraja = "baraja.bin";

    public static int menu()
    {
        int op = 0;
        while (op < 1 || op >5)
        {
            System.out.println("1. Nueva baraja");
            System.out.println("2. Barajar");
            System.out.println("3. Mostrar baraja");
            System.out.println("4. Imprimir baraja");
            System.out.println("5. Salir");
            op = Lector.LeerInt();
        }
        return op;
    }

    public static int menuBaraja()
    {
        int op = 0;
        while (op < 1 || op >2)
        {
            System.out.println("1. Baraja espa ola");
            System.out.println("2. Baraja francesa");
            op = Lector.LeerInt();
        }
        return op;
    }

    public static int menuEspa ola()
    {
        int op = 0;
        while (op < 1 || op >2)
        {
            System.out.println("1. Extendida (8 y 9)");
            System.out.println("2. No extendida");
            op = Lector.LeerInt();
        }
        return op;
    }

    public static void nuevaBaraja()
    {
        int op = menuBaraja();
        if (op==1)
        {
```

```

        boolean ext = (menuEspaniola()== 1);
        baraja = new BEspaniola(ext);
    }
    else
    {
        baraja = new BFrancesa();
    }
}

public static void barajar()
{
    if (baraja == null)
        System.out.println("No hay ninguna baraja creada");
    else
        baraja.barajar();
}

public static void mostrarBaraja()
{
    if (baraja == null)
        System.out.println("No hay ninguna baraja creada");
    else
        System.out.println(baraja);
}

public static void imprimirBaraja()
{
    if (baraja == null)
        System.out.println("No hay ninguna baraja creada");
    else
        baraja.imprimir();
}

public static void guardar()
{
    ObjectOutputStream oos = null;
    try
    {
        oos = new ObjectOutputStream(new FileOutputStream(fBaraja));
        oos.writeObject(baraja);
    }
    catch (IOException e)
    {
        System.out.println("Error guardando baraja");
    }
    finally
    {
        try
        {
            if (oos!=null)
                oos.close();
        }
        catch (IOException e)
        {
            System.out.println("Error cerrando el fichero");
        }
    }
}

```

```

public static void recuperar()
{
    ObjectInputStream ois = null;
    try
    {
        ois = new ObjectInputStream(new FileInputStream(fBaraja));
        baraja = (Baraja) ois.readObject();
    }
    catch (FileNotFoundException e)
    {
    }
    catch (Exception e)
    {
        System.out.println("Error recuperando la baraja");
    }
    finally
    {
        try
        {
            if (ois!=null)
                ois.close();
        }
        catch (IOException e)
        {
            System.out.println("Error cerrando el fichero");
        }
    }
}

public static void main(String[] args) {

    recuperar();
    int op = menu();
    while (op!=5)
    {
        switch (op)
        {
            case 1: nuevaBaraja(); break;
            case 2: barajar(); break;
            case 3: mostrarBaraja(); break;
            case 4: imprimirBaraja(); break;
        }
        op = menu();
    }
    guardar();
}
}

```

CARTA.JAV

```
package examen3ev22.pkg23;

import java.io.Serializable;

public class Carta implements Serializable{

    private int num;
    private String palo;

    public Carta(int n, String p)
    {
        num = n;
        palo = p;
    }

    public int getNum() {
        return num;
    }

    public String getPalo() {
        return palo;
    }

}
```

Baraja.java

```
package examen3ev22.pkg23;
```

```
import java.io.BufferedWriter;
```

```
import java.io.FileWriter;
```

```
import java.io.IOException;
```

```
import java.io.Serializable;
```

```
import java.util.ArrayList;
```

```
import java.util.Collections;
```

```
import java.util.Iterator;
```

```
public abstract class Baraja implements Serializable{
```

```
    protected ArrayList<Carta> baraja = new ArrayList<Carta>();
```

```
    public Baraja()
```

```
    {
```

```
    }
```

```
    public abstract void crearBaraja();
```

```
    public void barajar()
```

```
    {
```

```
        Collections.shuffle(baraja);
```

```
    }
```

```
    public abstract String nombre(int n);
```

```
    @Override
```

```
    public String toString()
```

```
    {
```

```
        String b="";
```

```

        Iterator <Carta> it = baraja.iterator();
        while (it.hasNext())
        {
            Carta c = it.next();
            b+= nombre(c.getNum())+" de "+c.getPalo()+"\n";
        }
        return b;
    }

    public void imprimir()
    {
        String fichero = Lector.leerLinea("Nombre del fichero: ");
        BufferedWriter bw = null;
        try
        {
            bw = new BufferedWriter(new FileWriter(fichero));
            bw.write(toString());
        }
        catch (IOException e)
        {
            System.out.println("Error escribiendo el fichero");
        }
        finally
        {
            try
            {
                if (bw!=null)
                    bw.close();
            }
            catch (IOException e)
            {
                System.out.println("Error cerrando el fichero");
            }
        }
    }

```



Bfrancesa.java

```
package examen3ev22.pkg23;
```

```
public class BFrancesa extends Baraja{

    private static String[] palos =
{"DIAMANTES", "PICAS", "CORAZONES", "TRÃ&BOLES"};
    private static int nCartas = 13;

    public BFrancesa()
    {
        crearBaraja();
    }

    @Override
    public void crearBaraja()
    {
        for (int i = 0; i < palos.length; i++)
        {
            for (int j = 1; j <=nCartas;j++)
            {
                baraja.add(new Carta(j,palos[i]));
            }
        }
    }

    @Override
    public String nombre(int n)
    {
        switch (n)
        {
            case 1: return "As";
            case 11: return "Jota";
            case 12: return "Reina";
            case 13: return "Rey";
            default: return n+"";
        }
    }
}
```

Bespaniola.java

**ackage** examen3ev22.pkg23;

```
public class Bespaniola extends Baraja{

    private static String[] palos = {"OROS","COPAS","ESPADAS","BASTOS"};
    private static int nCartas = 12;
    private boolean ext;

    public BEspaniola(boolean e)
    {

        ext = e;
        crearBaraja();
    }

    @Override
    public void crearBaraja()
    {
        for (int i = 0; i < palos.length; i++)
        {
            for (int j = 1; j <=nCartas;j++)
            {
                if (ext || (j!=8 && j!=9))
                {
                    baraja.add(new Carta(j,palos[i]));
                }
            }
        }
    }

    @Override
    public String nombre(int n)
    {
        switch (n)
        {
            case 1: return "As";
            case 10: return "Sota";
            case 11: return "Caballo";
            case 12: return "Rey";
            default: return n+"";
        }
    }
}
```