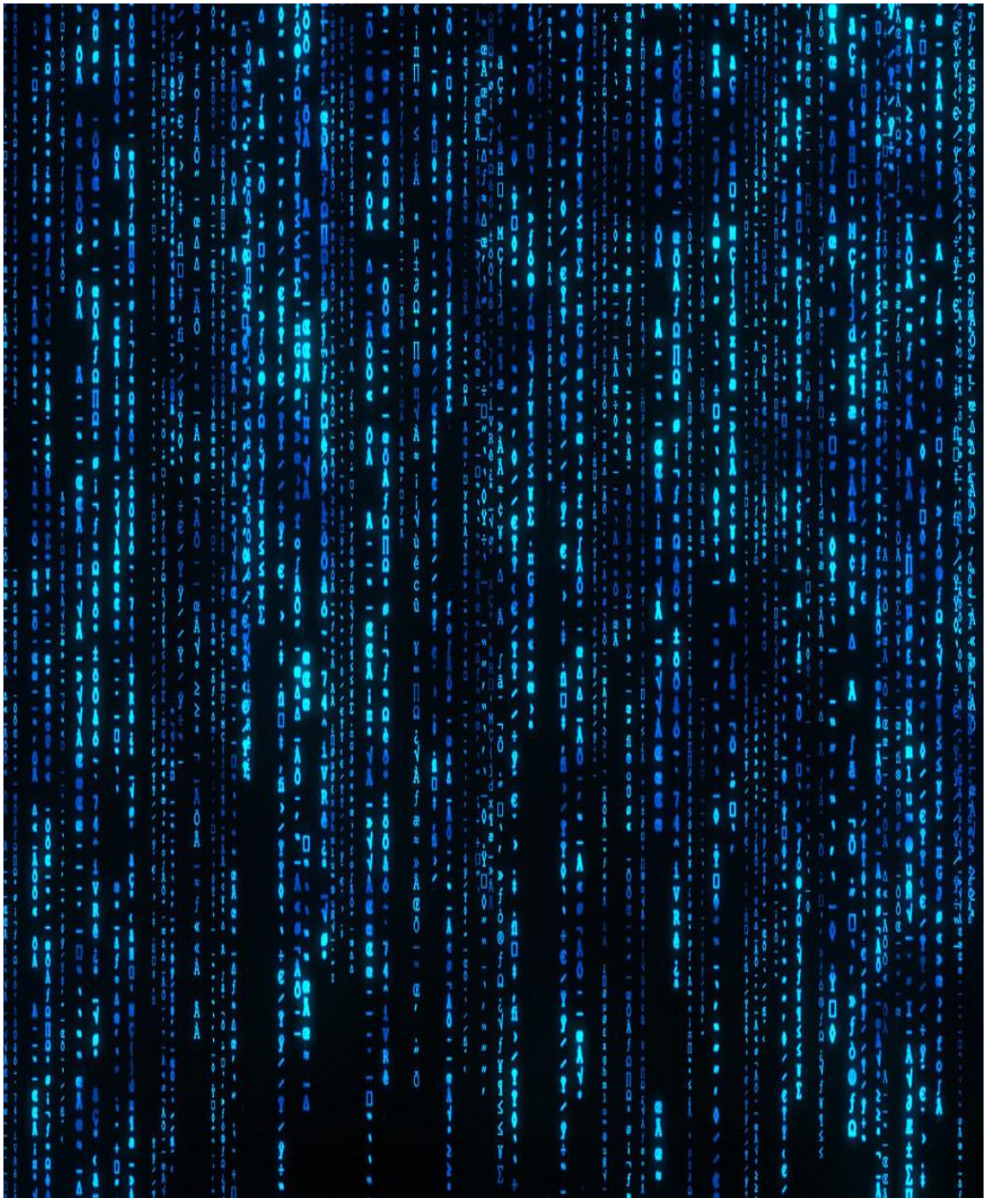


Nombre del alumno/a: **Roberto Crespo Escalona**

Nombre del módulo: **Entornos de desarrollo**

Ciclo formativo de grado Superior: **Desarrollo de aplicaciones web**

Curso: **2024-2025**

Fecha: **21/11/2024**

Profesor/a: **Jorge Tarancon Iglesias**

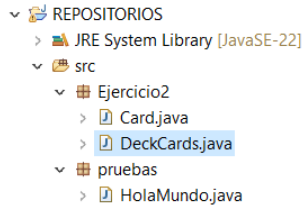
Instituto Educación Secundaria Kursaal

Algeciras.

ÍNDICE

EJERCICIOS

1. **Cree un proyecto en Eclipse con el código del proyecto.**



Lo primero que vamos hacer es ir al entorno de eclipse, una vez dentro de eclipse hacemos click derecho en el entorno y creamos nuevo proyecto, es decir le damos a (java Project): Una vez creado el proyecto, dentro de él crearemos dos paquetes uno llamada ejercicio 2 y otro llamado pruebas. Una vez creado los dos paquetes, en el primero crearemos dos clases una llamada (Card.java) y la otra llamada (DeckCarda.java) y luego en el paquete de pruebas crearemos la clase (HolaMundo.java). En cada clase pondremos el código exigido que necesitaremos para la realización de los siguientes apartados.

```java
package Ejercicio2;

public class Card {

    public String suit;
    public String value;

    public Card (String suit, String value) {
        this.suit = suit;
        this.value = value;
    }

    public String toString () {
        return (this.suit+"-"+this.value);
    }

}
```

```java
package pruebas;

public class HolaMundo {

    public static void main(String[] args) {

        // TODO Auto-generated method stub
            System.out.println("Hola GiT!!");

    }

}
```

```java
package Ejercicio2;

import java.util.ArrayList;

public class DeckCards {

    public static void main(String[] args) {

        String[] suits = { "Spades", "Diamonds", "Club", "Heart" };
        String[] values = { "Ace", "2", "3", "4", "5", "6", "7", "8", "9", "10", "Jack", "Queen", "King" };

        ArrayList<Card> deck = new ArrayList<Card>();

        for (int i = 0; i < suits.length; i++) {
            for (int j = 0; j < values.length; j++) {
                Card card = new Card(suits[i], values[j]);
                deck.add(card);
            }
        }

        for (int i = 0; i < deck.size(); i++) {
            int j = (int) Math.floor(Math.random() * i);
            Card tmp = deck.get(i);
            deck.set(i, deck.get(j));
            deck.set(j, tmp);
        }

        for (int i = 0; i < 5; i++) {
            System.out.println(deck.get(i));
        }
```

2. **Analice la estructura de archivos del proyecto, identificando cuales son ejecutables.**

Como podemos ver en el proyecto de hola mundo lo que nos aparece es la función de imprimir para luego mostrar en pantalla, ¿y que es lo que va a mostrar? Pues una vez lo ejecutemos los que va mostrar es el texto que hay dentro, es decir el Hola Git!

El proyecto de DeckCards.java, en un programa para la reacción de un mazo de cartas y al ejecutarlo de dará el resultado de las cartas de ese mazo. Y el proyecto Card.java, que se ejecuta pero no muestra un resultado en sí mismo, si no que los manda al proyecto DeckCards.java.

3. **Explique con sus palabras la funcionalidad de cada archivo. (Hacer una traza puede ayudar a comprender el funcionamiento).**

Pues como hemos dicho anteriormente, la funcionalidad del proyecto Hola Mundo es sacar por pantalla un mensaje que en este caso muestra Hola Git!, se realiza en un único método que es el main, que es el método que suele venir por defecto en el entorno java y la función que saca la información que es System.out.println("Hola Git!");

En el proyecto de Card.java. Este proyecto se encarga de ver si la carta del mazo contiene suit, es decir ya sea (Spades, Diamonds, Club, Heart) y luego ver si contiene value, es decir puede almacenar (Ace, 2, 3, 4, 5, 6,7, 8, 9, 10, Jack, Queen, King), reliza la creación de las cartas del mazo mientras se ejecuta y hace llamar a el proyecto Deckcards.java recibiendo así el suit y el value.

El proyecto de Deckcards.java lo que hace es que realiza la creación del mazo y luego muestra el resultado. Tenemos la función Arraylist que es donde se almacenaran las cartas del mazo, para la creación de estos se han utilizado bucles (for), uno para la creación del mazo en si con todos los palos, otro para barajar las cartas (mezclar) y otro para que nos muestre 5 cartas con los diferentes palos.

4. **Identifique las librerías Java utilizadas. Explique qué hace cada uno de los métodos de dichas librerías, indicando: datos de entrada, datos de salida y función que realiza. Consulte la documentación oficial para ello. Java API.**

Comandos utilizados:

En Deckcards.java:

Como utilidad de java tenemos la función Arraylist, ¿Qué es? Pues es una matriz una matriz de tamaño variable que se puede encontrar en el paquete de java.util. La diferencia entre una matriz incorporada y una Arraylist en java es que el tamaño de una matriz no se puede modificar, si desea agregar o eliminar elementos a/ de una matriz debe crear uno nuevo. ¿Y para qué sirve? Pues sirve para almacenar datos en memoria de forma similar que los Arrays. Con la ventaja de que el número de elementos que almacena, lo hace de forma dinámica, es decir, no es necesario declarar su tamaño como pasa con los Arrays.

Tenemos la función length que permite contar el número de carácter de una cadena (de un String) incluido todos los espacios y que además devuelve el número. En este caso nuestras variables de carácter String son suits y values que están nombradas como string.

Tenemos el meto Size que se encarga de realizar un seguimiento de los elementos. Usado también en los Arraylist, por ejemplo cuando creas un Arraylist sin ningún elemento el método size va a devolver 0 pero este valor va cambiando a medida que agregamos o eliminamos elementos.

Tenemos la clase Math que se encarga de proporcionar un conjunto de métodos estáticos para realizar varios cálculos matemáticos diferentes, podemos ver en el código que se le implementa floor que hace como valor decimal y que devuelve el valor entero más grande, menor o igual que el decimal especificado. La función Random devuelve un resultado aleatorio situado entre los valores específicos.

En Card.java:

Para qué sirve el comando "this". Pues para hacer referencia a la variable de instancia de la clase actual. Para invocar el método de la clase actual y para invocar el constructor de la clase actual.

Tenemos el método toString que devuelve una representación en una cadena de texto del objeto en el que los has empleado. Es útil para cuando se quiere devolver un resumen del estado interno del objeto.

En Hola Mundo:

La única función ejecutable que tenemos es System.out.println("Hola Git"); que se trata nada más y nada menos que de imprimir por pantalla el argumento que le das.

5. **Comente el código y súbalo a su cuenta de GitHub. Cree un proyecto nuevo público y adjunto su URL:**

```java
public class HolaMundo {

    // PROGRAMA QUE IMPRIME UN TEXTO Y LO MUESTRA POR EL MONITOR:

    public static void main(String[] args) {

        // TODO Auto-generated method stub
            System.out.println("Hola GiT!!"); // LA FUNCIÓN SISTEM.OUT. PRINTLN LO QUE HACE ES IMPRIMIR
                                              // POR PANTALLA EL VALOR QUE LE DAS, EN ESTE CASO TENEMOS UN
                                              //UN TEXTO QUE PONE "HOLA GIR". ENTONCES MOSTRARÁ POR EL MONITOR
                                              // HOLA GIT.

    }

}
```

```java
package Ejercicio2;

public class Card {

    // ES UN PROGRAMA QUE SE EJECUTA PERO QUE NO MUESTRA EN SI UN RESULTADO, POR QUE LO MANDA AL EL PROGRAMA
    // DECKCARDS.JAVA.

    public String suit; // AQUI SE DEFINE SI LAS CARTA DEL MAZO CONTIENE ( SPADE, DIAMANTE, CLUB, O HEARD)
    public String value; // Y AQUI SE DIFINE SI LA CARTA CONTIENE ( ACEM 2,3,4,5,6,7,8,9,10, QUEEN O KING)

    // ES DECIR SE CREAN LAS CARTAS DEL MAZO MIENTRAS LO EJECUTAMOS
    // Y LLEGAN A DECKCARDS.JAVA CON EL SUIT Y EL VALUE.

    public Card (String suit, String value) {
        this.suit = suit;
        this.value = value;
    }

    // VA A DEVOLVER EL RESULTADO EN UNA CADENA DE TEXTO.

    public String toString () {
        return (this.suit+"-"+this.value);
    }

}
```

```java
public class DeckCards {

    // ES UN PROGRAMA QUE SE ENCARGA DE LA CREACIÓN DEL MAZO DE LAS CARTAS QUE HAN SUDIO CREADAS EN LA CLASE
    // CARD.JAVA.

    public static void main(String[] args) {

        // SON LOS VALORES QUE SE LES DAN A LAS CARTAS.

        String[] suits = { "Spades", "Diamonds", "Club", "Heart" };
        String[] values = { "Ace", "2", "3", "4", "5", "6", "7", "8", "9", "10", "Jack", "Queen", "King" };

        ArrayList<Card> deck = new ArrayList<Card>(); // EN ESTA FUNCIÓN SE IRAN ALMACENANDO LAS CARTAS DEL
                                                      // MAZO QUE CREAREMOS MENDIAN BUCLES.

        for (int i = 0; i < suits.length; i++) { // BUCLE QUE SE ENCARGA DE CREAR EL MAZO DE CARTAS.
            for (int j = 0; j < values.length; j++) {
                Card card = new Card(suits[i], values[j]);
                deck.add(card);
            }
        }

        for (int i = 0; i < deck.size(); i++) { // BUCLE QUE SE ENCARGARA DE  BARAJAR  LAS CARTAS DEL MAZO.
            int j = (int) Math.floor(Math.random() * i);
            Card tmp = deck.get(i);
            deck.set(i, deck.get(j));
            deck.set(j, tmp);
        }

        for (int i = 0; i < 5; i++) { // NOS DARA COMO RESULTADO 5 CARTAS ALEATORIAMENTE.
            System.out.println(deck.get(i));
```