

UT2 PD1

EJERCICIO 1

- a)** Variable de Clase: x
tiene static
- b)** Variable de Instancia: y

Salida del código:

```
a.y = 5
b.y = 6
a.x = 2
b.x = 2
IdentifyMiParts.x = 2
```

EJERCICIO 2

Carta.java

```
/**
 * Clase Carta
 *
 * @author Magela
 */
public class Carta {

    /**
     * Contiene un String con el palo de la carta
     */
    public String palo;
```

```

/**
 * Contiene el numero de la carta
 */
public int numero;

/**
 * Asigna un palo a la carta
 * @param palo
 */
public void Palo(String palo) {
    this.palo = palo;
}

/**
 * Asigna un numero a la carta
 * @param numero
 */
public void Numero(int numero) {
    this.numero = numero;
}

/**
 * Metodo que retorna un String para imprimir con el Numero y Palo de una
 * carta
 *
 * @return Numero y Palo de la carta
 */
public String MostrarCarta() {

    return "Numero: " + String.valueOf(numero) + ", Palo: " + palo;
}
}

```

Ej2_cartas.java

```
public class Ej2_cartas {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here

        Mazo ma = new Mazo(); // Creo un mazo de cartas

        int i;

        for (i = 0; i < 48; i++) {
            System.out.println(ma.mazo[i].MostrarCarta()); // Imprimo el mazo
        }

    }
}
```

Mazo.java

```
package ej2_cartas;

/**
 * Clase Mazo, se generan 48 cartas y se guardan en el mazo el mazo es un array
 * de cartas
 *
 * @author Magela
 */
public class Mazo {
```

```

Carta[] mazo = new Carta[49]; // mazo

/**
 * Constructor, cuando se crea un mazo se crean 48 cartas del 12 de los
 * cuatro palos
 */
public Mazo() {
    int j;
    int cont = 0;

    // Creacion cartas Espada
    for (j = 1; j <= 12; j++) {
        mazo[cont] = new Carta();
        mazo[cont].Numero(j);
        mazo[cont].Palo("Espada");
        cont++;
    }

    // Creacion cartas Basto
    for (j = 1; j <= 12; j++) {
        mazo[cont] = new Carta();
        mazo[cont].Numero(j);
        mazo[cont].Palo("Basto");
        cont++;
    }

    // Creacion cartas Oro
    for (j = 1; j <= 12; j++) {

```

```
mazo[cont] = new Carta();

mazo[cont].Numero(j);

mazo[cont].Palo("Oro");

cont++;

}

// Creacion cartas Copa
for (j = 1; j <= 12; j++) {
    mazo[cont] = new Carta();
    mazo[cont].Numero(j);
    mazo[cont].Palo("Copa");
    cont++;
}
}
}
```

EJERCICIO 3

1)

```
public class SomethingIsWrong {
    public static void main(String[] args) {
        Rectangle myRect;
        myRect.width = 40;
        myRect.height = 50;
        System.out.println("myRect's area is " + myRect.area());
    }
}
```

Siendo Rectangle una clase definida, en el main se crea un objeto Rectangle pero no se inicializa.

Tendríamos que tener esto:

```
Rectangle myRect = new Rectangle();
```

2) Código corregido:

```
public class SomethingIsWrong {
    public static void main(String[] args) {
        Rectangle myRect = new Rectangle();
        myRect.width = 40;
        myRect.height = 50;
        System.out.println("myRect's area is " + myRect.area());
    }
}

public class Rectangle {
    public float width;
    public float height;

    public float area() {
        return width*height;
    }
}
```

EJERCICIO 4

1)

```
String[] students = new String[10];
String studentName = "Peter Parker";
students[0] = studentName;
studentName = null;
```

En el código existen una referencia a cada objeto.

Primero se crea el primer objeto `students` y la referencia apunta al objeto. Después se crea el segundo objeto `studentName` y su referencia apunta al objeto. Luego, la primera referencia apunta al segundo objeto, por lo que el primer objeto se queda sin referencia, es eliminado por el Garbage Collector y se libera memoria.

[Referencia de funcionamiento de Garbage Collector:

<https://srpoder.wordpress.com/2008/10/09/memoria-y-garbage-collection-en-java/>]

2) Java para destruir objetos creados que no están siendo referenciados y están perdidos en memoria ocupando esta utiliza el Garbage Collector, lo que hace es ejecutar `System.gc()`; eliminando en memoria objetos “perdidos”.

3)

```
NumberHolder nh = new NumberHolder();

System.out.println(" anInt = "+nh.anInt);

System.out.println(" aFloat = "+nh.aFloat);


nh.anInt = 3;

nh.aFloat = 5;


System.out.println(" anInt = "+nh.anInt);

System.out.println(" aFloat = "+nh.aFloat);
```

Resultado:

```
anInt = 0
aFloat = 0.0
anInt = 3
aFloat = 5.0
```