

Informe de Laboratorio 03

Tema: Arreglos de Objetos

Nota

Estudiante	Escuela	Asignatura
Mikhail Gabino Velasque Arcos mvelasquea@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Laboratorio FUNDAMENTOS DE LA PROGRAMACION Semestre: II Código: 20214260

Laboratorio	Tema	Duración
03	Arreglos y Objetos	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - B	Del 20 Setiembre 2023	Al 27 Setiembre 2023

1. Actividad

- Cree un Proyecto llamado Laboratorio3 Usted deberá agregar las clases Nave.java y DemoBatalla.java. Analice, complete y pruebe el Código de la clase DemoBatalla
- Solucionar la Actividad 4 de la Práctica 1 pero usando arreglo de objetos
- Solucionar la Actividad 5 de la Práctica 1 pero usando arreglos de objetos

2. SOLUCIONARIO

- Se hace el uso de arreglo de objetos como el de nave.java y DemoBatalla.java para completar la actividad como se muestra en la siguiente seccion

2.1. CODIGO FUENTE

- Se crea la clase Nave.
- Se crea la clase principal:

Listing 1: Creando la clase Nave

```
vim Nave.java  
vim DemoBatalla.java
```

Listing 2: Creando la clase Nave

```
public class Nave {
    private String nombre;
    private int fila;
    private String columna;
    private boolean estado;
    private int puntos;
    // Metodos mutadores
    public void setNombre( String n){
        nombre = n;
    }
    public void setFila(int f){
        fila = f;
    }
    public void setColumna(String c){
        columna = c;
    }
    public void setEstado(boolean e){
        estado = e;
    }
    public void setPuntos(int p){
        puntos = p;
    }
    // Metodos accesoires
    public String getNombre(){
        return nombre;
    }
    public int getFila(){
        return fila;
    }
    public String getColumna(){
        return columna;
    }
    public boolean getEstado(){
        return estado;
    }
    public int getPuntos(){
        return puntos;
    }
}
```

Listing 3: Creando la clase principal de DemoBatalla.java

```
public class DemoBatalla {
    public static void main(String [] args){
        Nave [] misNaves = new Nave[10];
        Scanner sc = new Scanner(System.in);
        String nomb, col;
        int fil, punt;
        boolean est;
        for (int i = 0; i < misNaves.length; i++) {
            System.out.println("Nave " + (i+1));
            System.out.print("Nombre: ");
            nomb = sc.next();
            System.out.println("Fila ");
```

```
fil = sc.nextInt();
System.out.print("Columna: ");
col = sc.nextInt();
System.out.print("Estado: ");
est = sc.nextBoolean();
System.out.print("Puntos: ");
punt = sc.nextInt();
misNaves[i] = new Nave(); //Se crea un objeto Nave y se asigna su referencia a misNaves
misNaves[i].setNombre(nomb);
misNaves[i].setFila(fil);
misNaves[i].setColumna(col);
misNaves[i].setEstado(est);
misNaves[i].setPuntos(punt);
}
System.out.println("\nNaves creadas:");
mostrarNaves(misNaves);
mostrarPorNombre(misNaves);
mostrarPorPuntos(misNaves);
System.out.println("\nNave con mayor nmero de puntos: " + mostrarMayorPuntos(misNaves));
}

// Mtodo para mostrar todas las naves
public static void mostrarNaves(Nave [] flota){
    for (int i = 0; i < flota.length; i++) {
        System.out.println("NOMBRE: " +flota[i].getNombre());
        System.out.println("FILA: " +flota[i].getFila());
        System.out.println("COLUMNA: " +flota[i].getColumna());
        if (flota[i].getEstado()==true) {
            System.out.println("vivo");
        } else {
            System.out.println("muerto");
        }
        System.out.println("PUNTOS DE VIDA: "+flota[i].getPuntos());
    }
}

// Mtodo para mostrar todas las naves de un nombre que se pide por teclado
public static void mostrarPorNombre(Nave [] flota){
    String nombre;
    System.out.println("INGRESAR EL NOMBRE DEL QUE SE BUSCA");
    Scanner objeto= new Scanner(System.in);
    nombre=objeto.nextLine();
    for (int i = 0; i < flota.length; i++) {
        if (flota[i].getNombre().equals(nombre)) {
            System.out.println("COINCIDENCIA CON: "+nombre);
            System.out.println("NOMBRE: " +flota[i].getNombre());
            System.out.println("FILA: " +flota[i].getFila());
            System.out.println("COLUMNA: " +flota[i].getColumna());
            if (flota[i].getEstado()==true) {
                System.out.println("vivo");
            } else {
                System.out.println("esta muerto");
            }
            System.out.println("PUNTOS DE VIDA: "+flota[i].getPuntos());
            break;
        }else{
            System.out.println("---");
        }
    }
}

// Mtodo para mostrar todas las naves con un nmero de puntos inferior o igual
```

```
//al nmero de puntos que se pide por teclado
public static void mostrarPorPuntos(Nave [] flota){
    for (int i = 0; i < flota.length; i++) {
        int puntoComparacion;
        System.out.println("INGRESE LA CANTIDAD DE VIDA QUE DESEA COMPARAR");
        Scanner objeto= new Scanner(System.in);
        puntoComparacion=objeto.nextInt();
        for (int j = 0; j < flota.length; j++) {
            if (flota[i].getPuntos()<=puntoComparacion) {
                System.out.println("NOMBRE: " +flota[i].getNombre());
                System.out.println("FILA: " +flota[i].getFila());
                System.out.println("COLUMNA: " +flota[i].getColumna());
                if (flota[i].getEstado()==true) {
                    System.out.println("vivo");
                } else {
                    System.out.println("muerto");
                }
                System.out.println("PUNTOS DE VIDA: "+flota[i].getPuntos());
            }else{
                System.out.println("---");
            }
        }
    }
}

// Mtodo que devuelve la Nave con mayor nmero de Puntos

public static Nave mostrarMayorPuntos(Nave[] flota) {
    Nave naveMayor = flota[0]; // Supongamos que la primera nave tiene la mayor
    cantidad de puntos inicialmente
    for (int i = 1; i < flota.length; i++) {
        if (flota[i].getPuntos() > naveMayor.getPuntos()) {
            naveMayor = flota[i];
        }
    }
    return naveMayor;
}
}
```