

Universidad Nacional de San Agustín
Escuela Profesional de Ingeniería de Sistemas
Fundamentos de Programación II
Practica de Laboratorio 3:
Arreglos de Objetos

Nombre: Jhonatan Benjamin Mamani Céspedes

CUI: 20232188

1. Analice, complete y pruebe el Código de la clase DemoBatalla

Código del programa

```
1 // LABORATORIO N° 3 - EJERCICIO 1
2 // AUTOR: JHONATAN BENJAMIN MAMANI CÉSPEDES
3 // TIEMPO: 57 MINUTOS
4 import java.util.*;
5 public class DemoBatalla {
6     Run | Debug
7     public static void main(String [] args){
8         Nave [] misNaves = new Nave[10];
9         Scanner sc = new Scanner(System.in);
10        String nomb, col;
11        int fil, punt;
12        boolean est;
13        for (int i = 0; i < misNaves.length; i++) {
14            System.out.println("Nave " + (i + 1));
15            System.out.print(s:"Nombre: ");
16            nomb = sc.next();
17            System.out.print(s:"Fila (Entre 1 y 10): ");
18            do{
19                fil = sc.nextInt();
20            } while (fil > 10 || fil < 1);
21            System.out.print(s:"Columna (De la A a la J): ");
22            String abc = "ABCDEFGHIIJ";
23            do {
24                col = sc.next().toUpperCase();
25            } while (abc.indexOf(col) == -1);
26            System.out.print(s:"Estado: ");
27            est = sc.nextBoolean();
28            System.out.print(s:"Puntos: ");
29            punt = sc.nextInt();
30            misNaves[i] = new Nave();
```

```

31         misNaves[i].setNombre(nomb);
32         misNaves[i].setFila(fil);
33         misNaves[i].setColumna(col);
34         misNaves[i].setEstado(est);
35         misNaves[i].setPuntos(punt);
36     }
37 }
38 System.out.println(x: "\nGenerando naves...");
39
40 System.out.println(x: "\nNaves creadas:");
41 mostrarNaves(misNaves);
42 System.out.print(s: "\nIntroduzca el nombre de la nave que busca: ");
43 mostrarPorNombre(misNaves);
44 System.out.print(s: "\nIntroduzca los puntos maximos que puede tener una nave: ");
45 mostrarPorPuntos(misNaves);
46 System.out.println(x: "\nEn su flota, la nave con el mayor numero de puntos es:");
47 mostrarMayorPuntos(misNaves);
48 }
49 public static void mostrarNaves(Nave [] flota){
50     for (int i = 0; i < flota.length; i++){
51         System.out.println("\nNave " + (i + 1) + ": " + flota[i].getNombre());
52         System.out.println("Ubicacion: " + flota[i].getColumna() + flota[i].getFila());
53         if (flota[i].getEstado() == true)
54             System.out.println(x: "Estado: Activo");
55         else
56             System.out.println(x: "Estado: Inactivo");
57         System.out.println("Puntos: " + flota[i].getPuntos());
58     }
59 }

```

```

60 public static void mostrarPorNombre(Nave [] flota){
61     Scanner sc = new Scanner(System.in);
62     String nombre = sc.next();
63     for (int i = 0; i < flota.length; i++){
64         if (nombre.equals(flota[i].getNombre())){
65             System.out.println("Nave " + (i + 1) + ": " + nombre);
66             System.out.println("Ubicacion: " + flota[i].getColumna() + flota[i].getFila());
67             if (flota[i].getEstado() == true)
68                 System.out.println(x: "Estado: Activo");
69             else
70                 System.out.println(x: "Estado: Inactivo");
71             System.out.println("Puntos: " + flota[i].getPuntos());
72         }
73     }
74 }
75 public static void mostrarPorPuntos(Nave [] flota){
76     Scanner sc = new Scanner(System.in);
77     int puntos = sc.nextInt();
78     for (int i = 0; i < flota.length; i++){
79         if (puntos >= flota[i].getPuntos()){
80             System.out.println("La nave " + (i + 1) + " (" + flota[i].getNombre() + ") "
81                 + " tiene: " + flota[i].getPuntos() + " puntos");
82         }
83     }
84 }

```

```

85     public static void mostrarMayorPuntos(Nave [] flota){
86         int mayor = flota[0].getPuntos();
87         int idx = 0;
88         for (int i = 0; i < flota.length; i++){
89             if (mayor < flota[i].getPuntos()){
90                 mayor = flota[i].getPuntos();
91                 idx++;
92             }
93         }
94         System.out.println("Nave " + (idx + 1) + ": " + flota[idx].getNombre());
95         System.out.println("Ubicacion: " + flota[idx].getColumna() + flota[idx].getFila());
96         if (flota[idx].getEstado() == true)
97             System.out.println(x:"Estado: Activo");
98         else
99             System.out.println(x:"Estado: Inactivo");
100        System.out.println("Puntos: " + flota[idx].getPuntos());
101    }
102 }

```

Ejecución del programa

Nave 1
Nombre: Martin
Fila (Entre 1 y 10): 2
Columna (De la A a la J): d
Estado: true
Puntos: 4
Nave 2
Nombre: Carlos
Fila (Entre 1 y 10): 2
Columna (De la A a la J): j
Estado: true
Puntos: 8
Nave 3
Nombre: Jose
Fila (Entre 1 y 10): 3
Columna (De la A a la J): e
Estado: true
Puntos: 6
Nave 4
Nombre: Alonso
Fila (Entre 1 y 10): 4
Columna (De la A a la J): b
Estado: false
Puntos: 1
Nave 5
Nombre: Leandro
Fila (Entre 1 y 10): 7
Columna (De la A a la J): f
Estado: false
Puntos: 7

Nave 6
Nombre: Daniel
Fila (Entre 1 y 10): 8
Columna (De la A a la J): i
Estado: true
Puntos: 5
Nave 7
Nombre: Jhonatan
Fila (Entre 1 y 10): 6
Columna (De la A a la J): d
Estado: true
Puntos: 11
Nave 8
Nombre: Ricardo
Fila (Entre 1 y 10): 9
Columna (De la A a la J): a
Estado: false
Puntos: 7
Nave 9
Nombre: Noel
Fila (Entre 1 y 10): 3
Columna (De la A a la J): b
Estado: false
Puntos: 8
Nave 10
Nombre: Marco
Fila (Entre 1 y 10): 6
Columna (De la A a la J): d
Estado: true
Puntos: 9

Generando naves...

Naves creadas:

Nave 1: Martin
Ubicacion: D2
Estado: Activo
Puntos: 4

Nave 2: Carlos
Ubicacion: J2
Estado: Activo
Puntos: 8

Nave 3: Jose
Ubicacion: E3
Estado: Activo
Puntos: 6

Nave 4: Alonso
Ubicacion: B4
Estado: Inactivo
Puntos: 1

Nave 5: Leandro
Ubicacion: F7
Estado: Inactivo
Puntos: 7

```
Nave 6: Daniel
Ubicacion: I8
Estado: Activo
Puntos: 5

Nave 7: Jhonatan
Ubicacion: D6
Estado: Activo
Puntos: 11

Nave 8: Ricardo
Ubicacion: A9
Estado: Inactivo
Puntos: 7

Nave 9: Noel
Ubicacion: B3
Estado: Inactivo
Puntos: 8

Nave 10: Marco
Ubicacion: D6
Estado: Activo
Puntos: 9
```

```
Introduzca el nombre de la nave que busca: Jhonatan
Nave 7: Jhonatan
Ubicacion: D6
Estado: Activo
Puntos: 11

Introduzca los puntos maximos que puede tener una nave: 6
La nave 1 (Martin) tiene: 4 puntos
La nave 3 (Jose) tiene: 6 puntos
La nave 4 (Alonso) tiene: 1 puntos
La nave 6 (Daniel) tiene: 5 puntos

En su flota, la nave con el mayor numero de puntos es:
Nave 7: Jhonatan
Ubicacion: D6
Estado: Activo
Puntos: 11
PS C:\Users\Lenovo>
```

2. Solucionar la Actividad 4 de la Práctica 1 pero usando arreglo de objetos

Actividad 4: Escribir un programa donde se creen 5 soldados considerando su nombre y nivel de vida. Ingresar sus datos y después mostrarlos.

Código del programa

Para esta actividad se solicita el manejo de objetos, por eso, se crea la clase Jugador:

```
1  String nombre;
2  String nivel;
3
4  // Constructor para inicializar los atributos del jugador
5  Jugador(String nombre, String nivel) {
6      this.nombre = nombre;
7      this.nivel = nivel;
8  }
9  }
```

Y finalmente, se sigue la misma lógica pero ahora agregando los objetos a los arreglos:

```
1  public class Act4 {
2      public static void main(String[] args) {
3          Scanner sc = new Scanner(System.in);
4
5          // Arreglo de objetos Jugador para almacenar la información de los jugadores
6          Jugador[] jugadores = new Jugador[5];
7
8          // Ingreso de los nombres y niveles de juego de los jugadores
9          System.out.println("Ingrese el nombre y nivel de juego de 5 jugadores:");
10
11         for (int i = 0; i < 5; i++) {
12             System.out.print("Nombre del jugador " + (i + 1) + ": ");
13             String nombre = sc.nextLine();
14
15             System.out.print("Nivel de juego del jugador " + (i + 1) + " (novato, profesional, veterano, estrella): ");
16             String nivel = sc.nextLine();
17
18             // Crear un nuevo objeto Jugador y almacenarlo en el arreglo
19             jugadores[i] = new Jugador(nombre, nivel);
20         }
21
22         // Mostrar los datos de los jugadores
23         System.out.println("\nDatos de los jugadores:");
24         System.out.println("Nombres\t\tNivel de juego");
25         for (Jugador jugador : jugadores) {
26             System.out.println(jugador.nombre + "\t\t" + jugador.nivel);
27         }
28     }
29 }
```

Ejecución del programa:

```
PS C:\Users\jhona> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetails'
Ingrese el nombre y nivel de juego de 5 jugadores:
Nombre del jugador 1: Pedro
Nivel de juego del jugador 1 (novato, profesional, veterano, estrella): novato
Nombre del jugador 2: Jose
Nivel de juego del jugador 2 (novato, profesional, veterano, estrella): veterano
Nombre del jugador 3: Carlos
Nivel de juego del jugador 3 (novato, profesional, veterano, estrella): estrella
Nombre del jugador 4: Pepe
Nivel de juego del jugador 4 (novato, profesional, veterano, estrella): profesional
Nombre del jugador 5: Mario
Nivel de juego del jugador 5 (novato, profesional, veterano, estrella): novato

Datos de los jugadores:
Nombres      Nivel de juego
Pedro        novato
Jose         veterano
Carlos       estrella
Pepe         profesional
Mario        novato
PS C:\Users\jhona>
```

3. Solucionar la Actividad 5 de la Práctica 1 pero usando arreglos de objetos

Actividad 5: Escribir un programa donde se creen 2 ejércitos, cada uno con un número aleatorio de soldados entre 1 y 5, considerando sólo su nombre. Sus datos se inicializan automáticamente con nombres tales como "Soldado0", "Soldado1", etc. Luego de crear los 2 ejércitos se deben mostrar los datos de todos los soldados de ambos ejércitos e indicar qué ejército fue el ganador.

Código del programa

De manera similar al anterior ejercicio se crea la clase Jugador:

```
1 class Jugador {
2     String nombre;
3
4     // Constructor para inicializar los atributos del jugador
5     Jugador(String nombre) {
6         this.nombre = nombre;
7     }
8 }
```

Y con la misma lógica del primer laboratorio, se actualizan los arreglos ahora para contener los objetos de la clase Jugador:

```
1 public class Act5 {
2     static Jugador[] team1 = new Jugador[21];
3     static Jugador[] team2 = new Jugador[21];
4     static Random random = new Random();
5
6     public static void main(String[] args) {
7         crearEquipos();
8         System.out.println("Datos de los jugadores del equipo 1:");
9         mostrarEquipo(team1);
10        System.out.println("\nDatos de los jugadores del equipo 2:");
11        mostrarEquipo(team2);
12
13        // Selecciona los titulares para los dos equipos
14        Jugador[] titularesE1 = selectTitulares(team1);
15        Jugador[] titularesE2 = selectTitulares(team2);
16
17        // Mostrar los titulares de ambos equipos
18        System.out.println("\nTitulares del Equipo 1:");
19        mostrarEquipo(titularesE1);
20        System.out.println("\nTitulares del Equipo 2:");
21        mostrarEquipo(titularesE2);
22
23        // Generar el número aleatorio de goles para los dos equipos
24        int golesE1 = goles();
25        int golesE2 = goles();
26        System.out.println("\nGoles del Equipo 1: " + golesE1);
27        System.out.println("Goles del Equipo 2: " + golesE2);
28
29        // Definir al ganador o si hubo un empate
30        determinarGanador(golesE1, golesE2);
31    }
32 }
```

```

33 // Método que genera los nombres de la forma JugadorX para los dos equipos de 21 jugadores
34 static void crearEquipos() {
35     for (int i = 0; i < 21; i++) {
36         team1[i] = new Jugador("Jugador" + (i + 1));
37         team2[i] = new Jugador("Jugador" + (i + 22));
38     }
39 }
40
41 // Método para mostrar a los jugadores dentro del arreglo
42 static void mostrarEquipo(Jugador[] team) {
43     for (int i = 0; i < 11; i++) {
44         System.out.println("Jugador " + (i + 1) + ": " + team[i].nombre);
45     }
46 }
47
48 // Método para seleccionar a los titulares de cada equipo
49 static Jugador[] selectTitulares(Jugador[] team) {
50     Jugador[] titulares = new Jugador[11]; // Arreglo para almacenar los titulares seleccionados
51     boolean[] selected = new boolean[21]; // Arreglo para rastrear los jugadores ya seleccionados
52     int count = 0;
53
54     while (count < 11) {
55         // Genera un índice aleatorio entre 0 y 20
56         int index = random.nextInt(21);
57
58         // Verifica que el jugador no haya sido seleccionado previamente y lo agrega al arreglo de titulares
59         if (!selected[index]) {
60             titulares[count] = team[index];
61             selected[index] = true;
62             count++;
63         }
64     }
65     return titulares;
66 }
67
68 // Genera un número aleatorio de goles entre 0 y 5
69 static int goles() {
70     return random.nextInt(6);
71 }
72
73 // Compara los goles y determina el ganador del encuentro (o empate)
74 static void determinarGanador(int golesE1, int golesE2) {
75     if (golesE1 > golesE2) {
76         System.out.println("\nEl Equipo 1 ha ganado el partido!");
77     } else if (golesE1 < golesE2) {
78         System.out.println("\nEl Equipo 2 ha ganado el partido!");
79     } else {
80         System.out.println("\n¡Hubo un empate!");
81     }
82 }
83 }

```

Ejecución del programa

```

PS C:\Users\jhona> & 'C:\Program Files\Java\jdk-21
Datos de los jugadores del equipo 1:
Jugador 1: Jugador1
Jugador 2: Jugador2
Jugador 3: Jugador3
Jugador 4: Jugador4
Jugador 5: Jugador5
Jugador 6: Jugador6
Jugador 7: Jugador7
Jugador 8: Jugador8
Jugador 9: Jugador9
Jugador 10: Jugador10
Jugador 11: Jugador11

```

Datos de los jugadores del equipo 2:

Jugador 1: Jugador22
Jugador 2: Jugador23
Jugador 3: Jugador24
Jugador 4: Jugador25
Jugador 5: Jugador26
Jugador 6: Jugador27
Jugador 7: Jugador28
Jugador 8: Jugador29
Jugador 9: Jugador30
Jugador 10: Jugador31
Jugador 11: Jugador32

Titulares del Equipo 1:

Jugador 1: Jugador2
Jugador 2: Jugador19
Jugador 3: Jugador12
Jugador 4: Jugador17
Jugador 5: Jugador18
Jugador 6: Jugador21
Jugador 7: Jugador3
Jugador 8: Jugador15
Jugador 9: Jugador9
Jugador 10: Jugador6
Jugador 11: Jugador11

Titulares del Equipo 2:

Jugador 1: Jugador23
Jugador 2: Jugador37
Jugador 3: Jugador32
Jugador 4: Jugador28
Jugador 5: Jugador27
Jugador 6: Jugador38
Jugador 7: Jugador29
Jugador 8: Jugador35
Jugador 9: Jugador36
Jugador 10: Jugador41
Jugador 11: Jugador30

Goles del Equipo 1: 2

Goles del Equipo 2: 5

El Equipo 2 ha ganado el partido!

PS C:\Users\jhona>