Informe de Laboratorio 02
Tema: Arreglos de Objetos

		Nota

Estudiante	Escuela	Asignatura
Dayana Katherine Cayo Lahuana	Escuela Profesional de Ingeniería de Sistemas	Fundamentos de Programación II Semestre: II Código:20192188
Laboratorio	**Tema**	**Duración**
02	Arreglos de Objetos	02 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - A	08-05-2024	12-05-2024

1. Tarea

- Implementar el algoritmo ……
- Utilizar Git para evidenciar su trabajo.
- Enviar trabajo al profesor en un repositorio GitHub Privado, dándole permisos como colaborador.

2. Equipos, materiales y temas utilizados

- Sistema Operativo ….
- OpenJDK 64-Bits 17.0.7.
- Git 2.39.2.
- Cuenta en GitHub con el correo institucional.

3. URL de Repositorio Github

- URL para el laboratorio 01 en el Repositorio GitHub.
- https://github.com/……

4. Actividades

4.1. Actividad 1

Cree un Proyecto llamado Laboratorio2 Usted deberá agregar las clases Nave.java y DemoBatalla.java.

Clase Nave

```
1  //Laboratorio Nro 2 - Ejercicio1
 2 //Autor: Dayana Katherine Cayo Lahuana
 3 package actividad01;
 4
 5 public class Nave {
 6       private String nombre;
 7       private int fila;
 8       private String columna;
 9       private boolean estado;
10       private int puntos;
11
12       // Metodos mutadores
13       public void setNombre( String n){
14             nombre = n;
15       }
16
17       public void setFila(int f){
18             fila = f;
19       }
20
21       public void setColumna(String c){
22             columna = c;
23       }
24
25       public void setEstado(boolean e){
26             estado = e;
27       }
28
29       public void setPuntos(int p){
30             puntos = p;
31       }
32
33       // Metodos accesores
34       public String getNombre(){
35             return nombre;
36       }
37
38       public int getFila(){
39             return fila;
40       }
41
42       public String getColumna(){
43             return columna;
```

```
44        }
45        public boolean getEstado(){
46              return estado;
47        }
48
49        public int getPuntos(){
50              return puntos;
51        }
52 }
```

Clase DemoBatalla

```
1 //Laboratorio Nro 2 - Ejercicio1
2 //Autor: Dayana Katherine Cayo Lahuana
3 package actividad01;
4 import java.util.*;
5 public class DemoBatalla {
6
7         public static void main(String[] args) {
8                 Nave [] misNaves = new Nave[3];
9                 Scanner sc = new Scanner(System.in);
10
11                String nomb, col;
12                int fil, punt;
13                boolean est;
14
15                for (int i = 0; i < misNaves.length; i++) {
16                        System.out.println("Nave " + (i+1));
17                        System.out.print("Nombre: ");
18                        nomb = sc.next();
19                        System.out.println("Fila ");
20                        fil = sc.nextInt();
21                        System.out.print("Columna: ");
22                        col = sc.next();
23                        System.out.print("Estado: ");
24                        est = sc.nextBoolean();
25                        System.out.print("Puntos: ");
26                        punt = sc.nextInt();
27
28                        misNaves[i] = new Nave();
29
30                        misNaves[i].setNombre(nomb);
31                        misNaves[i].setFila(fil);
32                        misNaves[i].setColumna(col);
33                        misNaves[i].setEstado(est);
34                        misNaves[i].setPuntos(punt);   }
35
36                System.out.println("\nNaves creadas:");
37                mostrarNaves(misNaves);
38                mostrarPorNombre(misNaves);
39                mostrarPorPuntos(misNaves);
```

```
40                System.out.println("\nNave con mayor número de puntos: " +
41 mostrarMayorPuntos(misNaves));
42         }
43         //Método para mostrar todas las naves public
44         static void mostrarNaves(Nave [] flota){
45                 for (int i = 0; i < flota.length; i++) {
46                  System.out.println("Nave " + (i + 1) + ":");
47                  System.out.println("Nombre: " + flota[i].getNombre());
48                  System.out.println("Fila: " + flota[i].getFila());
49                  System.out.println("Columna: " + flota[i].getColumna());
50                  System.out.println("Estado: " + flota[i].getEstado());
51                  System.out.println("Puntos: " + flota[i].getPuntos());
52                  System.out.println("--------------------------");
53             }
54         }
55
56         //Método para mostrar todas  las naves de un nombre que se pide por
57 teclado
58         public static void mostrarPorNombre(Nave [] flota){
59                 Scanner sc = new Scanner(System.in);
60             System.out.print("Ingrese el nombre de la nave que desea buscar: ");
61             String nombreBuscado = sc.next();
62
63             System.out.println("Naves con nombre '" + nombreBuscado + "':");
64             for (Nave nave : flota) {
65                 if (nave.getNombre().equalsIgnoreCase(nombreBuscado)) {
66                     System.out.println("Nombre: " + nave.getNombre());
67                     System.out.println("Fila: " + nave.getFila());
68                     System.out.println("Columna: " + nave.getColumna());
69                     System.out.println("Estado: " + nave.getEstado());
70                     System.out.println("Puntos: " + nave.getPuntos());
71                     System.out.println("--------------------------");
72                 }
73             }
74         }
75
76         //Método para mostrar todas las naves con un número de puntos inferior o
77 igual
78         //al número de puntos que se pide por teclado public
79         static void mostrarPorPuntos(Nave [] flota){
80                  Scanner sc = new Scanner(System.in);
81                     System.out.print("Ingrese el número de puntos máximo: ");
82                     int puntosMaximos = sc.nextInt();
83
84                     System.out.println("Naves con puntos igual o inferiores a "
85 + puntosMaximos + ":");
86                     for (Nave nave : flota) {
87                         if (nave.getPuntos() <= puntosMaximos) {
88                             System.out.println("Nombre: " + nave.getNombre());
89                             System.out.println("Fila: " + nave.getFila());
90                             System.out.println("Columna: " + nave.getColumna());
91                             System.out.println("Estado: " + nave.getEstado());
```

```
92                              System.out.println("Puntos: " + nave.getPuntos());
93                              System.out.println("--------------------------");
94                          }
95                      }
96          }
97
98          //Método que devuelve la Nave con mayor número de Puntos
99          public static Nave mostrarMayorPuntos(Nave [] flota){
100                 Nave naveMayorPuntos = flota[0];
101
102                     for (int i = 1; i < flota.length; i++) {
103                         if (flota[i].getPuntos() > naveMayorPuntos.getPuntos())
104 {
105                             naveMayorPuntos = flota[i];
106                         }
107                     }
108
109                     return naveMayorPuntos;
110         }
111         //Crear un método que devuelva un nuevo arreglo de objetos con todos los
112 //objetos previamente ingresados
113         //pero aleatoriamente desordenados  }
114         public static Nave[] navesDesordenadas(Nave[] flota) {
115         Nave[] flotaDesordenada = new Nave[flota.length];
116         System.arraycopy(flota, 0, flotaDesordenada, 0, flota.length);
117
118         Random rand = new Random();
119
120         for (int i = flotaDesordenada.length - 1; i > 0; i--) {
121             int j = rand.nextInt(i + 1);
122             Nave temp = flotaDesordenada[i];
123             flotaDesordenada[i] = flotaDesordenada[j];
124             flotaDesordenada[j] = temp;
125         }
126
127         return flotaDesordenada;
128     }
129
130 }
131
```

4.2. Actividad 2

Escribir un programa donde se creen 5 jugadores considerando su nombre y nivel de juego. Ingresar sus datos y después mostrarlos. Restricción: Aplicar arreglo de objetos

Clase Jugador

```
1  //Laboratorio Nro 2 - Ejercicio2
2 //Autor: Dayana Katherine Cayo Lahuana
3 package actividad02;
4
5 public class Jugador {
6       private String nombre;
7       private int nivel;
8
9
10      public String getNombre() {
11            return nombre;
12      }
13      public void setNombre(String nombre) {
14            this.nombre = nombre;
15      }
16      public int getNivel() {
17            return nivel;
18      }
19      public void setNivel(int nivel) {
20            this.nivel = nivel;
21      }
22 }
```

Clase Principal02

```
1  //Laboratorio Nro 2 - Ejercicio2
2 //Autor: Dayana Katherine Cayo Lahuana
3 package actividad02;
4 import java.util.*;
5
6 public class Principal02 {
7
8       public static void main (String []args) {
9             final int N=5;
10            Jugador equipo [] = new Jugador [N];
11            Scanner input=new Scanner(System.in);
12
13            for(int i=0; i<equipo.length;i++) {
14                  equipo[i]= new Jugador();
15                  System.out.println("Ingresar Nombre");
16                  equipo[i].setNombre(input.next());
17                  System.out.println("Ingresar Nivel");
18                  equipo[i].setNivel(input.nextInt());
19            }
20
```

```
21              for(int i=0; i<equipo.length;i++) {
22                      System.out.println("Jugador " + (i+1) );
23                      System.out.println("Nombre: " + equipo[i].getNombre());
24                      System.out.println("Nivel: " + equipo[i].getNivel());
25                      System.out.println("--------------------------------");
26              }
27
28              input.close();
29      }
30
31 }
```

4.3. Actividad 3

Escribir un programa donde se creen 2 equipos de soccer, cada equipo tiene 21 jugadores de los cuales sólo salen al campo 11 (elegir los titulares de manera aleatoria, considerando sólo su nombre). Sus datos se inicializan automáticamente con nombres tales como “Jugador10”, “Jugador8”, etc. Luego de crear los 2 equipos se deben mostrar los datos de todos los jugadores de ambos equipos e indicar qué equipo fue el ganador. Restricción: aplicar arreglos estándar y métodos para inicializar los equipos, mostrar los equipos y mostrar equipo ganador. La métrica a aplicar para indicar el ganador es el mayor número de goles anotados de cada equipo (generar la cantidad de goles de manera aleatoria), puede haber empates. (Todavía no aplicar arreglo de objetos).

```
1  //Laboratorio Nro 2 - Ejercicio3
2  //Autor: Dayana Katherine Cayo Lahuana
3  package actividad03;
4  import java.util.*;
5  public class Actividad03 {
6
7       public static void main(String[] args) {
8               //DECLARACION Y CREACION DE LOS ARREGLOS
9               final int N = 21;
10
11              String equipo1 [] = new String [N];
12              String equipo2 [] = new String [N];
13
14              inicializarEquipo(equipo1);
15              inicializarEquipo(equipo2);
16
17              System.out.println("----------EQUIPO1----------");
18              mostrarEquipo(equipo1);
19              System.out.println("----------EQUIPO2----------");
20              mostrarEquipo(equipo2);
21
22              System.out.println("----------TITULARES EQUIPO1----------");
```

```
23              mostrarEquipo(seleccionarTitulares(equipo1));
24              System.out.println("----------TITULARES EQUIPO2----------");
25              mostrarEquipo(seleccionarTitulares(equipo2));
26
27              int golesEquipo1 = generarGoles();
28              int golesEquipo2 = generarGoles();
29
30              System.out.println("--------------MARCADOR---------------");
31              System.out.println("EQUIPO1 : "+ golesEquipo1 + " VS " + "EQUIPO2 :
32 "+golesEquipo2);
33
34                                  System.out.println("--------------GANADOR------------
35 ---");
36              System.out.println(determinarGanador(golesEquipo1, golesEquipo2));
37      }
38
39      //METODO DE INICIALIZAR JUGADORES
40      public static void inicializarEquipo(String equipo[])
41      {
42              String nombre = "Jugador";
43              for(int i = 0;i<equipo.length; i++)
44              {
45                      equipo[i] = nombre + (i+1);
46              }
47      }
48
49      //METODO DE MOSTRAR JUGADORES
50      public static void mostrarEquipo(String equipo[])
51      {
52              for(int i = 0;i<equipo.length; i++)
53              {
54                      System.out.println(equipo[i]);
55              }
56      }
57
58      // METODO DE 11 TITULARES
59      public static String[] seleccionarTitulares(String[] equipo) {
60              Random random = new Random();
61              String[] titulares = new String[11];
62              boolean[] seleccionados = new boolean[equipo.length];
63
64              for (int i = 0; i < 11; i++)
65              {
66                      int index;
67                      do
68                      {
69                              index = random.nextInt(equipo.length);
70                      }
71                      while (seleccionados[index]);
72
73                      titulares[i] = equipo[index];
74                      seleccionados[index] = true;
```

```
75              }
76              return titulares;
77        }
78        //METODO PARA GENERAR LOS GOLES DEL 1 AL 7
79        public static int generarGoles() {
80              Random random = new Random();
81              return random.nextInt(8);
82        }
83        //METODO PARA CONOCER EL GANADOR
84        public static String determinarGanador(int golesEquipo1, int golesEquipo2) {
85              if (golesEquipo1 > golesEquipo2) {
86                    return "EQUIPO 1";
87              } else if (golesEquipo2 > golesEquipo1) {
88                    return "EQUIPO 2";
89              } else {
90                    return "EMPATE";
91              }
92        }

  }
```

5. Cuestionario:

5.1. Pregunta 1

¿Qué ventajas tienen los arreglos de objetos sobre los arreglos estándar?

Los arreglos de objetos ofrecen flexibilidad al permitir almacenar cualquier tipo de objeto y promueven la reutilización del código al encapsular datos y comportamientos relacionados en objetos individuales.

5.2. Pregunta 2

¿Cómo debe inicializarse un arreglo de objetos?

Un arreglo de objetos se inicializa asignando una nueva instancia de cada objeto al índice correspondiente del arreglo, utilizando el operador new para crear cada objeto y luego asignando cada objeto al arreglo.

5.3. Pregunta 3

¿Cuándo deberíamos usar los arreglos de objetos?
Deberíamos usar arreglos de objetos cuando necesitamos almacenar y manipular múltiples instancias de objetos relacionados, lo que proporciona flexibilidad, reutilización de código y facilita la gestión de datos estructurados en la programación.

7. Referencias

- https://www.w3schools.com/java/default.asp
- https://www.geeksforgeeks.org/java/