

Informe de Laboratorio 09

Tema: Definición de Clases de Usuario Clase Soldado

Nota

Estudiante	Escuela	Asignatura
Sarayasi Huanaco,Jeferson Jesus	Escuela Profesional de Ingeniería de Sistemas	Fundamentos de la Programación 2

Laboratorio	Tema	Duración
09	Definición de Clases de Usuario Clase Soldado	48 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - B		

Docente
Aedo Lopez, Marco Wilfredo

1. Tarea

- Que el alumno demuestre poder crear “clases definidas por el programador”
- Implementar métodos para las clases definidas por el programador
- Crear Métodos Sobrecargados (Overloaded)

2. Equipos, Materiales y temas utilizados

- Sistema Operativo Windows 11.
- VIM 9.0.
- OpenJDK 64-Bits 17.0.7.
- Git 2.39.2.
- Cuenta en GitHub con el correo institucional
- Arreglos Estandar

3. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar
- <https://github.com/JefersonSH/FP2-2024B.git>
- URL para el laboratorio 07 en el Repositorio GitHub.
- https://github.com/JefersonSH/FP2-2024B/tree/eb261ca92998cb186b738e515a242259f00efc83/Laboratorio_09

4. Ejercicios Resuletos

4.1. Commits

Ejercicio 01

- **Commit 1:** VideoJuego numero 6 con las modificaciones pedidas

Listing 1: Clase Direccion

```
1 import java.util.*;
2
3 public class VideoJuego6 {
4
5     public static void main(String[] args){
6         Scanner sc = new Scanner(System.in);
7
8         int tamao1 = (int)(Math.random()*10+1); //Cantidad de soldados del ejercito 1
9         int tamao2 = (int)(Math.random()*10+1); //Cantidad de soldados del ejercito 2
10        Soldado[][] tablero = new Soldado[10][10]; //Creando un arreglo de Soldados
11                                   bidimensional
12
13        AsignarAtributos(tablero, tamao1, 1); //Creando Soldados en el ejercito 0
14        AsignarAtributos(tablero, tamao2, 2); //Creando Soldados en el ejercito 1
15
16        //Metodos pedidos en la practica
17        ImprimirTablero(tablero);
18        ImprimirDatos(tablero);
19        System.out.println("Soldado con Mas Vida del Ejercito 1: ");
20        EncontrarMayorVida(tablero, 10, 1);
21        System.out.println("Soldado con Mas Vida del Ejercito 2: ");
22        EncontrarMayorVida(tablero, 10, 2);
23        System.out.println("Promedio de Vida: " + PromedioVida(tablero, tamao1 + tamao2));
24        System.out.println("\n-----\n");
25        ImprimirVida(tablero);
26        System.out.println("Ranking de Vida por Ordenamiento Burbuja");
27        OrdenamientoBurbuja(tablero, tamao1 + tamao2);
28        System.out.println("Ranking de Vida por Ordenamiento por Insercion");
29        OrdenamientoInsercion(tablero, tamao1 + tamao2);
30
31        EjercitoGanador(tablero);
32    }
33
34    public static void AsignarAtributos(Soldado[][] tablero, int n, int ejercito){
35        for(int i = 0; i < n; i++){
```

```
35     int x = (int)(Math.random()*10); //x representa a las columnas, siendo del 1 al 9
36     int y = (int)(Math.random()*10); //y representa a las filas, siendo del 1 al 9
37     if(tablero[y][x] == null){ //Si la posicion y x esta vacia, crea un Soldado en esa
        posicion
38         String nombre = "Soldado" + (ejercito) + "x" + (i + 1);
39         int vida = (int)(Math.random()*5+1);
40         int ataque = (int)(Math.random()*5+1);
41         int defensa = (int)(Math.random()*5+1);
42         tablero[y][x] = new Soldado(nombre, vida, ataque, defensa, x + 1, y + 1, ejercito);
43     } else { //Si la posicion y x no esta vacia, el ciclo continua por el i--
44         i--;
45     }
46 }
47 }
48
49 public static void ImprimirTablero(Soldado[][] tablero){
50     for(int i = 0; i < 10; i++){ //filas
51         for(int j = 0; j < 10; j++){ //columnas
52             if(tablero[j][i] != null) //Si el indice contiene un Soldado continua
53                 if(tablero[j][i].getVive()){ //Si el soldado esta vivo continua
54                     if(tablero[j][i].getEjercito() == 1) System.out.print("1 ");
55                     if(tablero[j][i].getEjercito() == 2) System.out.print("2 ");
56                 } else System.out.print("=" );
57                 else System.out.print("=" );
58
59             }
60             System.out.print("\n"); //Una vez acabada una fila, se pasa a la siguiente linea
61         }
62         System.out.println("\n-----\n");
63     }
64
65     public static void ImprimirDatos(Soldado[][] tablero){
66         for(int i = 0; i < 10; i++)
67             for(int j = 0; j < 10; j++)
68                 if(tablero[j][i] != null)
69                     if(tablero[j][i].getVive()){
70                         System.out.println("Nombre: " + tablero[j][i].getNombre());
71                         System.out.println("Salud: " + tablero[j][i].getVidaActual());
72                         System.out.println("Fila: " + tablero[j][i].getFila());
73                         System.out.println("Columna: " + tablero[j][i].getColumna());
74                         System.out.println("Ejercito: " + tablero[j][i].getEjercito());
75                     }
76         System.out.println("\n-----\n");
77     }
78
79     public static void EncontrarMayorVida(Soldado[][] tablero, int n, int ejercito){
80         int mayor = Integer.MIN_VALUE;
81
82         Soldado masVida = new Soldado();
83         for(int i = 0; i < n; i++) //Filas
84             for(int j = 0; j < n; j++) //Columnas
85                 if(tablero[j][i] != null)
86                     if(tablero[j][i].getVive()){
87                         if(tablero[j][i].getEjercito() == ejercito) //Para solamente comparar
88                             soldados del mismo ejercito
89                             if(tablero[j][i].getVidaActual() > mayor){
```

```
89         mayor = tablero[j][i].getVidaActual(); //Si es mayor, actualiza la
90             variable
91         masVida = tablero[j][i];
92     }
93     System.out.println("Nombre: " + masVida.getNombre());
94     System.out.println("Vida: " + masVida.getVidaActual());
95     System.out.println("Fila: " + masVida.getFila());
96     System.out.println("Fila: " + masVida.getFila());
97     System.out.println("Columna: " + masVida.getColumna());
98     System.out.println("Ejercito" + masVida.getEjercito());
99     System.out.println("\n-----\n");
100 }
101
102 public static int PromedioVida(Soldado[] [] tablero, int n){
103     int promedio = 0;
104     for(int i = 0; i < 10; i++) //Filas
105         for(int j = 0; j < 10; j++) //Columnas
106             if(tablero[j][i] != null)
107                 if(tablero[j][i].getVive())
108                     promedio += tablero[j][i].getVidaActual();
109     return promedio/n;
110 }
111
112 public static void ImprimirVida(Soldado[] [] tablero){
113     for(int i = 0; i < 10; i++) //Filas
114         for(int j = 0; j < 10; j++) //Columnas
115             if(tablero[j][i] != null)
116                 if(tablero[j][i].getVive())
117                     System.out.println("Vida de " + tablero[j][i].getNombre() + ": " +
118                         tablero[j][i].getVidaActual());
119     System.out.println("\n-----\n");
120 }
121
122 public static void OrdenamientoBurbuja(Soldado[] [] tablero, int n){
123     Soldado temp = new Soldado();
124     int contador = 0;
125     Soldado[] lista = new Soldado[n];
126     for(int i = 0; i < 10; i++) //Filas
127         for(int j = 0; j < 10; j++) //Columnas
128             if(tablero[j][i] != null){
129                 if(tablero[j][i].getVive()){
130                     lista[contador] = tablero[j][i];
131                     contador++;
132                 }
133             }
134
135     for(int i = 0; i < n - 1; i++)
136         for(int j = 0; j < n - 1; j++)
137             if(lista[j].getVidaActual() > lista[j+1].getVidaActual()){
138                 temp = lista[j+1];
139                 lista[j+1] = lista[j];
140                 lista[j] = temp;
141             }
142     for(int i = n - 1; i >= 0; i--){
143         System.out.println("Nombre: " + lista[i].getNombre());
144     }
```

```
143     System.out.println("Vida: " + lista[i].getVidaActual());
144     System.out.println("Fila: " + lista[i].getFila());
145     System.out.println("Columna" + lista[i].getColumna());
146     System.out.println("Ejercito" + lista[i].getEjercito());
147 }
148 System.out.println("\n-----\n");
149 }
150
151 public static void OrdenamientoInsercion(Soldado[] [] tablero, int n){
152     Soldado temp = new Soldado();
153     int contador = 0;
154     Soldado[] lista = new Soldado[n];
155     for(int i = 0; i < 10; i++) //Filas
156         for(int j = 0; j < 10; j++) //Columnas
157             if(tablero[j][i] != null){
158                 if(tablero[j][i].getVive()){
159                     lista[contador] = tablero[j][i];
160                     contador++;
161                 }
162             }
163
164     for(int i = 1; i < n; i++){
165         Soldado key = lista[i];
166         int j = i - 1;
167         while(j >= 0 && lista[j].getVidaActual() > key.getVidaActual()){
168             lista[j + 1] = lista[j];
169             j = j - 1;
170         }
171         lista[j + 1] = key;
172     }
173
174     for(int i = n - 1; i >= 0; i--){
175         System.out.println("Nombre: " + lista[i].getNombre());
176         System.out.println("Vida: " + lista[i].getVidaActual());
177         System.out.println("Fila: " + lista[i].getFila());
178         System.out.println("Columna" + lista[i].getColumna());
179         System.out.println("Ejercito" + lista[i].getEjercito());
180     }
181     System.out.println("\n-----\n");
182 }
183
184 public static void EjercitoGanador(Soldado[] [] tablero){
185     int ejercito1 = 0;
186     int ejercito2 = 0;
187     for(int i = 0; i < 10; i++) //Filas
188         for(int j = 0; j < 10; j++) //Columnas
189             if(tablero[j][i] != null)
190                 if(tablero[j][i].getVive()){
191                     if(tablero[j][i].getEjercito() == 1) ejercito1 +=
192                         tablero[j][i].getVidaActual();
193                     //Si es del ejercito 1, la salud se le suma al total de ese ejercito
194                     if(tablero[j][i].getEjercito() == 2) ejercito2 +=
195                         tablero[j][i].getVidaActual();
196                     //Si es del ejercito 2, la salud se le suma al total de ese ejercito
197                 }
198     //Imprimir quien es el ganador, teniendo en cuenta la vida total de cada uno de los
```

```
ejercitos
197     if(ejercito1>ejercito2) System.out.println("Ganador: Ejercito 1");
198     else if(ejercito1 < ejercito2) System.out.println("Ganador: Ejercito 2");
199     else System.out.println("Empate, no hay ganador");
200     System.out.println("\n-----\n");
201 }
202 }
```

- **Commit 2:** Se modificaron y crearon algunos metodos del constructor

Listing 2: Clase Persona

```
1 public class Soldado{
2     private String nombre;
3     private int nivelAtaque;
4     private int nivelDefensa;
5     private int nivelVida;
6     private int vidaActual;
7     private int velocidad;
8     private String actitud;
9     private boolean vive;
10    private int columna;
11    private int fila;
12    private int ejercito;
13
14    public Soldado(){
15        nivelVida = 0;
16        vidaActual = 0;
17        nivelAtaque = 0;
18        nivelDefensa = 0;
19        vive = true;
20    }
21
22    public Soldado(String nombre){
23        this.nombre = nombre;
24        vive = true;
25    }
26
27    public Soldado(String nombre, int nivelVida, int nivelAtaque, int nivelDefensa, int fila,
28        int columna, int ejercito){
29        this.nombre = nombre;
30        this.nivelVida = nivelVida;
31        this.vidaActual = nivelVida;
32        this.nivelAtaque = nivelAtaque;
33        this.nivelDefensa = nivelDefensa;
34        this.ejercito = ejercito;
35        this.fila = fila;
36        this.columna = columna;
37        vive = true;
38    }
39
40    public void atacar(){
41        avanzar();
42        actitud = "ofensiva";
43    }
```

```
43
44 public void defender(){
45     velocidad = 0;
46     actitud = "defensiva";
47 }
48
49 public void avanzar(){
50     velocidad++;
51 }
52
53 public void retroceder(){
54     if(velocidad > 0){
55         velocidad = 0;
56         actitud = "defensiva";
57     } else velocidad --;
58 }
59
60 public void serAtacado(int num){
61     vidaActual -= num;
62 }
63
64 public void huir(){
65     velocidad += 2;
66     actitud = "fuga";
67 }
68
69 public void morir(){
70     if(vidaActual <= 0){
71         vive = false;
72     }
73 }
74
75 public void setVidaActual(int vidaActual){
76     this.vidaActual = vidaActual;
77 }
78
79 public int getVidaActual(){
80     return vidaActual;
81 }
82
83 public void setColumna(int columna){
84     this.columna = columna;
85 }
86
87 public int getColumna(){
88     return columna;
89 }
90
91 public void setFila(int fila){
92     this.fila = fila;
93 }
94
95 public int getFila(){
96     return fila;
97 }
98
```

```
99     public void setEjercito(int ejercito){
100         this.ejercito = ejercito;
101     }
102
103     public int getEjercito(){
104         return ejercito;
105     }
106
107     public boolean getVive(){
108         return vive;
109     }
110
111     public String getNombre(){
112         return nombre;
113     }
114 }
```

Programa compilado


```
= 2 = = = = = = = =  
= = 1 = = = = = = =  
= = = = = = = 1 = =  
= = = = = = = = = =  
= = = = = = = = = =  
= = = = = = = = = 2  
= = = = = = 2 = = =  
= = = = = = = 2 = =  
= = = = = = = = = =  
= = = = = = = = = =
```

Nombre: Soldado2x2

Salud: 2

Fila:1

Columna: 2

Ejercito: 2

Nombre: Soldado1x1

Salud: 4

Fila:2

Columna: 3

Ejercito: 1

Nombre: Soldado1x2

Salud: 5

Fila:3

Columna: 8

Ejercito: 1

Nombre: Soldado2x3

Salud: 1

Fila:6

Columna: 10

Ejercito: 2

Nombre: Soldado2x4

Salud: 1

Fila:7

Columna: 7

Ejercito: 2

Nombre: Soldado2x1

Salud: 3

Fila:8

Columna: 8

Ejercito: 2

Soldado con Mas Vida del Ejercito 1:

Nombre: Soldado1x2

Vida: 5

Fila: 3

Fila: 3

Columna: 8

Ejercito1

Soldado con Mas Vida del Ejercito 2:

Nombre: Soldado2x1

Vida: 3

Fila: 8

Fila: 8

Columna: 8

Ejercito2

Promedio de Vida: 2

Vida de Soldado2x2: 2

Vida de Soldado1x1: 4

Vida de Soldado1x2: 5

Vida de Soldado2x3: 1

Vida de Soldado2x4: 1

Vida de Soldado2x1: 3

Ranking de Vida por Ordenamiento Burbuja

Nombre: Soldado1x2

Vida: 5

Fila: 3

Columna8

Ejercito1

Nombre: Soldado1x1

Vida: 4

Fila: 2

Columna3

Ejercito1

Nombre: Soldado2x1

Vida: 3

Fila: 8

Columna8

Ejercito2

Nombre: Soldado2x2

Vida: 2

Fila: 1

Columna2

Ejercito2

Nombre: Soldado2x4

Vida: 1

Fila: 7

Columna7

Ejercito2

Nombre: Soldado2x3

Vida: 1

Fila: 6

Columna10

Ejercito2

```
Ranking de Vida por Ordenamiento Burbuja
Nombre: Soldado1x2
Vida: 5
Fila: 3
Columna8
Ejercito1
Nombre: Soldado1x1
Vida: 4
Fila: 2
Columna3
Ejercito1
Nombre: Soldado2x1
Vida: 3
Fila: 8
Columna8
Ejercito2
Nombre: Soldado2x2
Vida: 2
Fila: 1
Columna2
Ejercito2
Nombre: Soldado2x4
Vida: 1
Fila: 7
Columna7
Ejercito2
Nombre: Soldado2x3
Vida: 1
Fila: 6
Columna10
Ejercito2
```

5. Referencias

- https://www.w3schools.com/js/js_ajax_intro.asp
- <https://www.youtube.com/watch?v=cAqmF7mtZv0>
- <https://www.youtube.com/watch?v=cAqmF7mtZv0>