

# Universidad Autónoma de Baja California

## Ingeniero en computación



### Programación Orientada A Objetos

#### Practica 7: Arreglos y Herencia En Java

**Nombre:** López Mercado Brayan

**Matricula:** 1280838

**Grupo:** 542

**Docente:** Mayra Janeth Duran Rodríguez

**Fecha de entrega:** 21 de abril del 2022

## Ejercicio 1

Imprimir en pantalla el elemento que se encuentra en la posición 5 para la inicialización del siguiente arreglo:

```
int[] integerArray1 = {1,20,3,44,5,60,7,88,9,100};
```

```
1  /*
2  Nombre: Lopez Mercado Brayan
3  Matricula: 1280838
4  Materia: Programación Orientada a Objetos
5  Práctica 5: Encapsulamiento, Cohesión y Acoplamiento
6  */
7
8  public class Ejercicio_1 {
9
10     public static void main(String[] args) {
11         int[] integerArray1 = {1,20,3,44,5,60,7,88,9,100};
12         System.out.println(integerArray1[4]);
13     }
14 }
<terminated> Ejercicio_1 [Java Application] C:\Program Files (x86)\Java\jre1.8.0_261\bin\javaw.exe (19/04/2022)
```

## Ejercicio 2

Ejercicio 2. Inicializar un arreglo de tamaño 5 usando 2 métodos diferentes. El primer método inicializará el arreglo con puros ceros y el segundo con números aleatorios.

```
8  public class Ejercicio2 {
9
10     public static void main(String[] args) {
11         double lista1 []= new double [5];
12         double lista2 []= new double [5];
13         System.out.println("Lista 1");
14         fillNormal(lista1);
15         System.out.println("Lista 2");
16         fillRandom(lista2);
17     }
18
19     public static void fillNormal(double list[]){
20         for (int i=0;i<list.length;i++){
21             list[i]=0;
22         }
23         printList(list);
24     }
25 }
```

```

25
26 public static void fillRandom(double list[]){
27     for (int i=0;i<list.length;i++){
28         double a= 1 + Math.random()*100;
29         list[i]=a;
30     }
31     printList(list);
32 }
33
34 public static void printList(double array[]){
35     for (int j=0;j<array.length;j++){
36         System.out.println(array[j]);
37     }
38 }
39 }

```

```

<terminated> Ejercicio2 [Java Application] C:\Program Files (x86)\
Lista 1
0.0
0.0
0.0
0.0
0.0
Lista 2
66.72239574092042
59.827727235797454
35.597366292013746
43.93658039435439
57.1652606811451

```

### Ejercicio 3

Crear 3 objetos usando identificadores y agregarlos a un arreglo, después crear otro arreglo con 3 objetos instanciados en la declaración.

```

7 public class Ejercicio3 {
8
9     public static void main(String[] args) {
10         persona[] array= new persona[]{new persona ("Brayan", "Mercado"), new persona("Elena", "Martinez"),
11             new persona("Jose", "Rivera")};
12         for(int i=0;i<array.length;i++){
13             array[i].display();
14         }
15
16         System.out.println(" ** Array 2 ** " + "\n" );
17         persona a= new persona ("Armando", "Barreras");
18         persona b= new persona("Agustin", "Chacón");
19         persona c= new persona("Maria", "Cortez");
20         persona[] array2= new persona[]{a,b,c};
21         for(int i=0;i<array2.length;i++){
22             array2[i].display();
23         }
24     }
25 }
26 }

```

```

28 class persona{
29     public String name,apellido;
30
31     persona(String name, String apellido){
32         this.name=name;
33         this.apellido=apellido;
34     }
35
36     public void display(){
37         System.out.println("Nombre: " + name + "\n" + "Apellido: " + apellido + "\n" );
38     }
39 }

```

```

<terminated> Ejercicio3 [Java Application] C:\P
Nombre: Brayan
Apellido: Mercado

Nombre: Elena
Apellido: Martinez

Nombre: Jose
Apellido: Rivera

** Array 2 **
Nombre: Armando
Apellido: Barreras

Nombre: Agustin
Apellido: Chacón

Nombre: María
Apellido: Cortez

```

#### Ejercicio 4

Busque la clase Object en la documentación de Java, Observe los métodos disponibles y vea si puede usarlos con un objeto Perro o Animal.

- Algunos métodos como: *boolean equals()* y *Object Clone()* son los métodos que se podrían usarse en los objetos perro o animal, con ellos se puede saber si los objetos son iguales o crear otro objeto del mismo tipo respectivamente.

## Ejercicio 5

Crear otra subclase representando un animal diferente al perro.

```
7 public class Pig extends Animal{
8     public Pig(String name, int age){
9         super(name,age);
10    }
11    public void oink(){
12        System.out.println(name + ": oink oink");
13    }
14 }
```

## Practica

### Clase Main

```
8 import java.util.Scanner;
9 public class AngryBirdMain {
10
11     public static void main(String[] args) {
12         String opc;
13         Scanner Entry= new Scanner(System.in);
14
15         do{
16             Red Red1 = new Red();
17             Blue Blue1= new Blue();
18             Bomb Bomb1= new Bomb();
19             AngryBirdMain set []= new AngryBirdMain[]{Red1,Blue1,Bomb1};
20             for (int i=0;i<set.length;i++){
21                 set[i].display();
22             }
23             System.out.println("De Nuevo?");
24             opc=Entry.nextLine();
25             opc=opc.toLowerCase();
26         }while(opc.equals("s"));
27
28     }
29
30     public static double launch(){
31         int angle=(int) (1 + Math.random()*90);
32         System.out.println("Angulo de lanzamiento: " + angle + " Grados" );
33         return angle;
34     }
35
36     public static int speed(){
37         int speed=(int) (15 + Math.random()*20);
38         System.out.println("Velocidad De Lanzamiento: " + speed + " m/s" );
39         return speed;
40     }
}
```

```

41
42 public static void chirp() {
43     System.out.println("Angry Chirping");
44 }
45
46 public static void fly() {
47     System.out.println("Lanzamiento realizado");
48 }
49
50 public static void createBlank() {
51     System.out.print("\n");
52 }
53
54 public static void impact() {
55     System.out.println("Impact");
56 }
57
58 public void display() {
59     //Hola, estoy aqui para mostrar el array.
60 }
61
62 }

```

### Clase Red

```

8 public class Red extends AngryBirdMain{
9
10     public Red() {
11         fly();
12         attackRed();
13         launch();
14         speed();
15         chirp();
16         impact();
17         createBlank();
18     }
19
20
21     public void attackRed() {
22         System.out.println("Red Atacó");
23     }
24 }

```

## Clase Blue

```
8 public class Blue extends AngryBirdMain{
9     public Blue() {
10         fly();
11         attackBlue();
12         launch();
13         speed();
14         chirp();
15         statusBlue();
16         createBlank();
17     }
18
19
20     public void attackBlue() {
21         System.out.println("Blue Atacó");
22     }
23
24     public void division() {
25         System.out.println("Blue se dividió en tres");
26     }
27
28     public void statusBlue() {
29         int number=(int) (Math.random()*2);
30         if(number==1) {
31             division();
32             impact();
33         }
34         else{
35             impact();
36         }
37     }
38
39 }
```

## Clase Bomb

```
8 public class Bomb extends AngryBirdMain {
9     public Bomb() {
10         fly();
11         attackBomb();
12         launch();
13         speed();
14         chirp();
15         statusBomb();
16         createBlank();
17     }
18
19 }
```

```

20 public void attackBomb() {
21     System.out.println("Bomb Atacó");
22 }
23
24 public void kaboom() {
25     System.out.println("Kaboom*");
26 }
27
28 public void statusBomb() {
29     int number=(int) (Math.random()*2);
30     if(number==0){
31         kaboom();
32     }
33     else{
34         impact();
35         kaboom();
36     }
37 }
38 }

```

## Ejecución

AngryBirdMain [Java Application] C:\Program Files (x86)\Java\jdk-8.0.60\bin\java.exe

```

Lanzamiento realizado
Blue Atacó
Angulo de lanzamiento: 59 Grados
Velocidad De Lanzamiento: 26 m/s
Angry Chirping*
Impact*

```

```

Lanzamiento realizado
Bomb Atacó
Angulo de lanzamiento: 74 Grados
Velocidad De Lanzamiento: 29 m/s
Angry Chirping*
Impact*
Kaboom*

```

De Nuevo?

AngryBirdMain [Java Application] C:\Program Files (x86)\Java\jdk-8.0.60\bin\java.exe

```

s
Lanzamiento realizado
Red Atacó
Angulo de lanzamiento: 56 Grados
Velocidad De Lanzamiento: 15 m/s
Angry Chirping*
Impact*

```

```

Lanzamiento realizado
Blue Atacó
Angulo de lanzamiento: 82 Grados
Velocidad De Lanzamiento: 17 m/s
Angry Chirping*
Impact*

```



### **Link de Repositorio En Github**

<https://github.com/BrayanLMercado/Practica7.git>

Nombre: Lopez Mercado Brayan

Matrícula: 1280838

Materia: Programación Orientada a Objetos

Práctica 7: Arreglos y Herencia En Java

