

PRÁCTICA 05 - PROGRAMACIÓN ORIENTADA A OBJETOS

Alejandro Vaquera López

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 public class Array {  
    Run | Debug  
2     public static void main(String [] args){  
3         int[] integerArray = {1,20,3,44,5,60,7,88,9,100};  
4         System.out.println(integerArray[5]);  
5     }  
6 }  
7
```

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.

```
public class Array {  
    Run | Debug  
    public static void main(String [] args){  
        //ejercicio 1  
        int[] integerArray = {1,20,3,44,5,60,7,88,9,100};  
        System.out.println(integerArray[5]);  
  
        //ejercicio 2  
        int[] ceroArray = {0,0,0,0,0};  
        int[] randomArray = new int[5];  
        for(int cont = 0; cont < 5; cont++){  
            Random rand = new Random();  
            randomArray[cont] = rand.nextInt();  
        }  
    }  
}
```

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

```
//Int[] myClassArray = new int[]{new MyClass(), new MyClass()}
MyClass[] container = new MyClass[3];
MyClass firstInstance = new MyClass();
MyClass secondInstance = new MyClass();
MyClass thirdInstance = new MyClass();
container[0] = firstInstance;
container[1] = secondInstance;
container[2] = thirdInstance;

MyClass[] myClassArray = new MyClass[]{new MyClass(), new MyClass()};
}
```

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.

```
public class AnimalMain{
    Run | Debug
    public static void main(String [] args){
        Aligator aligator = new Aligator(name:"pepe", age:1);
        aligator.eat();//pepe is eating
    }
}
```

Ejercicio 5. Crear otra subclase representando un animal diferente al perro

```
class Animal{
    String name;
    int age;
    public Animal(String name, int age){
        this.name = name;
        this.age = age;
    }

    public void eat(){
        System.out.println(name+ " is eating");
    }
}

class Aligator extends Animal {
    public Aligator (String name, int age){
        super (name, age);
    }

    public void bite(){
        System.out.println(name + "is eating");
    }
}

public class AnimalMain{
    Run | Debug
    public static void main(String [] args){
    }
}
```

Práctica 05

1 superclase Bird y 2 subclases que heredan

```
C:\Users\Vacintosh\Documents\projects\P00-Claudia-tona\Practica 05>javac BirdMain.java
C:\Users\Vacintosh\Documents\projects\P00-Claudia-tona\Practica 05>java BirdMain
attack
PrimerPajaro
is not visible now

attack
SegundoPajaro
exploded

C:\Users\Vacintosh\Documents\projects\P00-Claudia-tona\Practica 05>
```