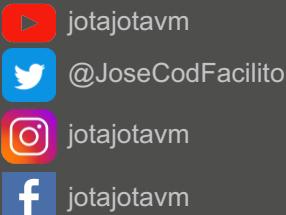




Curso de Fundamentos de Programacion

 jotajotavm.com/programacion



EJERCICIOS SECCION 6



JAVA

Aquí tienes los enunciados de los ejercicios recomendados para esta sección dependiendo del lenguaje que quieras aprender. Recuerda que puede haber varias maneras de desarrollar el código para un mismo ejercicio. Aquí tienes una posible solución

Crea una clase de tipo atleta con propiedades donde puedas guardar datos genéricos y con funciones para entrenar y descansar. También una función que sea para celebrar una victoria.

```
public class Atleta {  
    private String nombre;  
    private String deporte;  
    private int energia;  
  
    public Atleta(String nombre, String deporte) {  
        this.nombre = nombre;  
        this.deporte = deporte;  
        this.energia = 100;  
    }  
  
    public void entrenar(int horas) {  
        System.out.println(nombre + " ha entrenado por " + horas + " horas.");  
    }  
  
    public void descansar() {  
        System.out.println(nombre + " está descansando.");  
    }  
  
    public void celebrarVictoria() {  
        System.out.println(nombre + " ha ganado y está celebrando la victoria.");  
    }  
}
```



Curso de Fundamentos de Programacion

 jotajotavm.com/programacion



-  jotajotavm
-  @JoseCodFacilito
-  jotajotavm
-  jotajotavm

EJERCICIOS SECCION 6



JAVA

Aquí tienes los enunciados de los ejercicios recomendados para esta sección dependiendo del lenguaje que quieras aprender. Recuerda que puede haber varias maneras de desarrollar el código para un mismo ejercicio. Aquí tienes una posible solución

Crea un objeto futbolista que herede de atleta todas las propias y metodos, ademas debe tener propiedades exclusivas como "equipo de futbol" y "posicion" y un metodo para celebrar la victoria pero de una forma propia diferente a la de atleta.

```
public class Futbolista extends Atleta {  
    private String equipo;  
    private String posicion;  
  
    public Futbolista(String nombre, String deporte, String equipo, String posicion) {  
        super(nombre, deporte);  
        this.equipo = equipo;  
        this.posicion = posicion;  
    }  
  
    public void celebrarVictoria() {  
        System.out.println("¡Gooooool! " + getNombre() + " ha marcado y está celebrando con su equipo " + equipo + ".");  
    }  
}
```



Curso de Fundamentos de Programacion

 jotajotavm.com/programacion



-  jotajotavm
-  @JoseCodFacilito
-  jotajotavm
-  jotajotavm

EJERCICIOS SECCION 6



JAVA

Aquí tienes los enunciados de los ejercicios recomendados para esta sección dependiendo del lenguaje que quieras aprender. Recuerda que puede haber varias maneras de desarrollar el código para un mismo ejercicio. Aquí tienes una posible solución

Crea otra clase tenista que herede de atleta todas propiedades propias y metodos, ademas debe tener propiedades exclusivas como "grand slams" y debe tener un metodo para celebrar la victoria pero de una forma propia diferente a la de atleta y a la de futbolista.

```
public class Tenista extends Atleta {  
    private int grandSlams;  
  
    public Tenista(String nombre, String deporte, int grandSlams) {  
        super(nombre, deporte);  
        this.grandSlams = grandSlams;  
    }  
  
    public void celebrarVictoria() {  
        System.out.println(getNombre() + " ha ganado el torneo de tenis y está celebrando su victoria con su familia y su equipo.");  
    }  
  
    public int getGrandSlams() {  
        return grandSlams;  
    }  
  
    public void setGrandSlams(int grandSlams) {  
        this.grandSlams = grandSlams;  
    }  
}
```

NOTA: En este caso se pueden ver los métodos get y set. Estos métodos también deberían crearse en la clase atleta y futbolista con las propiedades correspondientes de esas clases



Curso de Fundamentos de Programacion



jotajotavm.com/programacion



EJERCICIOS SECCION 6



JAVA

Aquí tienes los enunciados de los ejercicios recomendados para esta sección dependiendo del lenguaje que quieras aprender. Recuerda que puede haber varias maneras de desarrollar el código para un mismo ejercicio. Aquí tienes una posible solución

Crea objetos de tipo atleta, futbolista y tenista y llama a sus métodos para entrenar, descansar y celebrar la victoria.

Comprueba que todos ejecutan del mismo modo los métodos de entrenar y descansar pero de una forma diferente el método de celebrar la victoria gracias al polimorfismo

```
public class Main {
    public static void main(String[] args) {
        // Crear un objeto Atleta
        Atleta atleta = new Atleta("Juan", "Atletismo");
        atleta.entrenar();
        atleta.descansar();
        atleta.celebrarVictoria();

        // Crear un objeto Futbolista
        Futbolista futbolista = new Futbolista("Lionel Messi", "Fútbol", "Paris Saint-Germain", "Delantero");
        futbolista.entrenar();
        futbolista.descansar();
        futbolista.celebrarVictoria();

        // Crear un objeto Tenista
        Tenista tenista = new Tenista("Rafael Nadal", "Tenis", 20);
        tenista.entrenar();
        tenista.descansar();
        tenista.celebrarVictoria();
    }
}
```