

Unidad 03 Utilización de objetos

Ejemplos



Módulo Programación
modalidad @vanza

Unidad 03 Utilización de objetos

Vamos a trabajar los contenidos de la unidad 3 con tres ejemplos sencillos:

- Clase Pajaro
- Clase Perro
- Clase Libro

Los pasos serán similares:

- Definir la clase: Atributos y métodos.
 - Dentro de los métodos: siempre interesa tener los set, get y el toString.
- Crear el programa principal:
 - crear una instancia de la clase y trabajar con ella



Ejemplo: Clase Pájaro

Ejercicio resuelto : Apartado 5 de la unidad 3

Clase principal llamada **Pajaro**, se definen los atributos y métodos:

- `pajaro()`. **Constructor por defecto**. En este caso, el constructor por defecto no contiene ninguna instrucción, ya que Java inicializa de forma automática las variables miembro, si no le damos ningún valor.
- `pajaro(String nombre, int posX, int posY)`. **Constructor** que recibe como argumentos una cadena de texto y dos enteros para inicializar el valor de los atributos.
- `volar(int posX, int posY)`. **Método** que recibe como argumentos dos enteros: `posX` y `posY`, y devuelve un valor de tipo `double` como resultado, usando la palabra clave `return`. El valor devuelto es el resultado de aplicar un desplazamiento de acuerdo con una fórmula

Diseña un programa que utilice la clase Pajaro, cree una instancia de dicha clase y ejecute sus métodos.



Ejemplo: Clase Pájaro

Lo primero, crear la **clase Pajaro**, con sus métodos y atributos:

```
public class Pajaro {  
  
    String nombre;  
    int posX, posY;  
  
    public Pajaro() {  
  
    }  
  
    public Pajaro(String nombre, int posX, int posY) {  
        this.nombre=nombre;  
        this.posX=posX;  
        this.posY=posY;  
    }  
  
    double volar (int posX, int posY) {  
  
        double desplazamiento = Math.sqrt( posX*posX + posY*posY );  
        this.posX = posX;  
        this.posY = posY;  
  
        return desplazamiento;  
    }  
}
```



Ejemplo: Clase Pájaro

Ahora la clase principal con su método main().

- **Crear** un objeto de la clase e inicializarlo, para ello utilizaremos el constructor con parámetros.
- **Invocar o ejecutar** el método volar del objeto creado.
- **Imprimir** por pantalla la distancia recorrida.

```
public static void main(String[] args) {  
    Pajaro loro = new Pajaro("Lucy",50,50) ;  
    double d = loro.volar(50,50);  
    System.out.println("El desplazamiento ha sido " + d);  
}
```



Ejemplo: Clase Pájaro

Si ejecutamos nuestro programa el resultado sería el siguiente:

```
Output - UtilizarObjetos (run)
run:
El desplazamiento ha sido 70.71067811865476
BUILD SUCCESSFUL (total time: 0 seconds)
```



Ejemplo: Clase Perro

Ejemplo Perro

Dada una clase principal llamada **Perro**, se definen los atributos y métodos.

- **Atributos:** nombre, raza , color y edad.
- perro(). **Constructor por defecto..**
- perro(String nombre, String raza, String color, int edad). **Constructor** que recibe como argumentos cadenas de texto .
- Ladrar(). **Método** que muestra “Guau Guau”
- Comer() **Método** que muestra “Ñan Ñan”
- Dormir() **Método** que muestra “Zzzzzz”

Diseña un programa que cree una instancia de la clase y ejecute sus métodos.



Ejemplo: Clase Perro

- Atributos y constructores:

```
public class Perro {  
    //Atributos  
    private String nombre;  
    private String raza;  
    private String color;  
    private int edad;  
  
    //Constructor por defecto  
    public Perro() {  
    }  
  
    //Constructor con parámetros  
    public Perro(String nombre, String raza, String color, int edad) {  
        this.nombre = nombre;  
        this.raza = raza;  
        this.color = color;  
        this.edad = edad;  
    }  
}
```




Ejemplo: Clase Perro

- Set y get:

```
//getters y setters
public String getNombre() {
    return nombre;
}
public void setNombre(String nombre) {
    this.nombre = nombre;
}
public String getRaza() {
    return raza;
}
public void setRaza(String raza) {
    this.raza = raza;
}
public String getColor() {
    return color;
}
public void setColor(String color) {
    this.color = color;
}
public int getEdad() {
    return edad;
}
public void setEdad(int edad) {
    this.edad = edad;
}
```



Ejemplo: Clase Perro

- Métodos y toString:

```
//Métodos
public void ladrar() {
    System.out.println("Guau guau");
}

public void comer() {
    System.out.println("Ñan ñan");
}

public void dormir() {
    System.out.println("Zzzzz");
}

@Override
public String toString() {
    return "Perro{" + "nombre=" + nombre + ", raza=" + raza +
        ", color=" + color + ", edad=" + edad + '}';
}
```



Ejemplo: Clase Perro

- Main():

```
public static void main(String[] args) {  
  
    Perro can=new Perro("Scooby","Gran Danes","Marron",5);  
  
    System.out.println(can.toString());  
  
    can.ladRAR();  
    can.comer();  
    can.dormir();  
}
```



Ejemplo: Clase Perro

- Resultado:

A screenshot of a Java IDE's output window. The window has a title bar with a trash icon and the text 'Salida'. Below the title bar are two tabs: 'Ejemplos (run)' (active) and 'Ejemplos (run) #3'. The active tab shows the following output:

```
run:
Perro{nombre=Scooby, raza=Gran Danes, color=Marron, edad=5}
Guau guau
Ñan ñan
Zzzzz
BUILD SUCCESSFUL (total time: 0 seconds)
```



Ejemplo: Clase Libro

Ejemplo Libro

Dada una clase principal llamada **Libro**, se definen los atributos y métodos.

- **Atributos:** Titulo, autor, número de ejemplares y número de prestamos.
- libro(). **Constructor por defecto..**
- Libro(String titulo, String autor, int ejemplares, int prestamos). **Constructor** que recibe como argumentos cadenas de texto y números enteros.
- Prestar(). **Método** que incrementa en uno el atributo numero de préstamo siempre que no supere el numero de ejemplares.
- Devolver() **Método** que decrementa en uno el atributo numero de préstamo siempre que exista algún préstamo.

Diseña un programa que cree una instancia de la clase y ejecute sus métodos.



Ejemplo: Clase Libro

Atributos y constructores:

```
public class Libro {  
    //Atributos  
    private String titulo;  
    private String autor;  
    private int ejemplares;  
    private int prestamos;  
  
    //Constructor por defecto  
    public Libro() {  
    }  
  
    //Constructor con parámetros  
    public Libro(String titulo, String autor, int ejemplares, int prestados) {  
        this.titulo = titulo;  
        this.autor = autor;  
        this.ejemplares = ejemplares;  
        this.prestamos = prestados;  
    }  
}
```



Ejemplo: Clase Libro

Set y get:

```
//getters y setters
public String getAutor() {
    return autor;
}
public void setAutor(String autor) {
    this.autor = autor;
}
public int getEjemplares() {
    return ejemplares;
}
public void setEjemplares(int ejemplares) {
    this.ejemplares = ejemplares;
}
public int getPrestados() {
    return prestamos;
}
public void setPrestados(int prestados) {
    this.prestamos = prestados;
}
public String getTitulo() {
    return titulo;
}
public void setTitulo(String titulo) {
    this.titulo = titulo;
}
```



Ejemplo: Clase Libro

Métodos:

```
//Método para realizar el prestamo de un libro
public boolean prestar() {
    boolean prestado = true;
    if (prestamos < ejemplares) {
        prestamos++;
    } else {
        prestado = false;
    }
    return prestado;
}
```

```
//Método para devolver un libro
public boolean devolver() {
    boolean devuelto = true;
    if (prestamos == 0) {
        devuelto = false;
    } else {
        prestamos--;
    }
    return devuelto;
}
```




Ejemplo: Clase Libro

toString:

```
//método toString sobrescrito para mostrar los datos de la clase Libro
@Override
public String toString() {
    return "titulo: " + titulo + "\nautor: " + autor +
        "\nejemplares: " + ejemplares + "\nprestados: " + prestamos;
}
```



Ejemplo: Clase Libro

Main():

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    String titulo, autor;
    int ejemplares;

    System.out.print("Introduce titulo: ");
    titulo = sc.nextLine();
    System.out.print("Introduce autor: ");
    autor = sc.nextLine();
    System.out.print("Numero de ejemplares: ");
    ejemplares = sc.nextInt();

    Libro libro1=new Libro(titulo,autor,ejemplares,0);

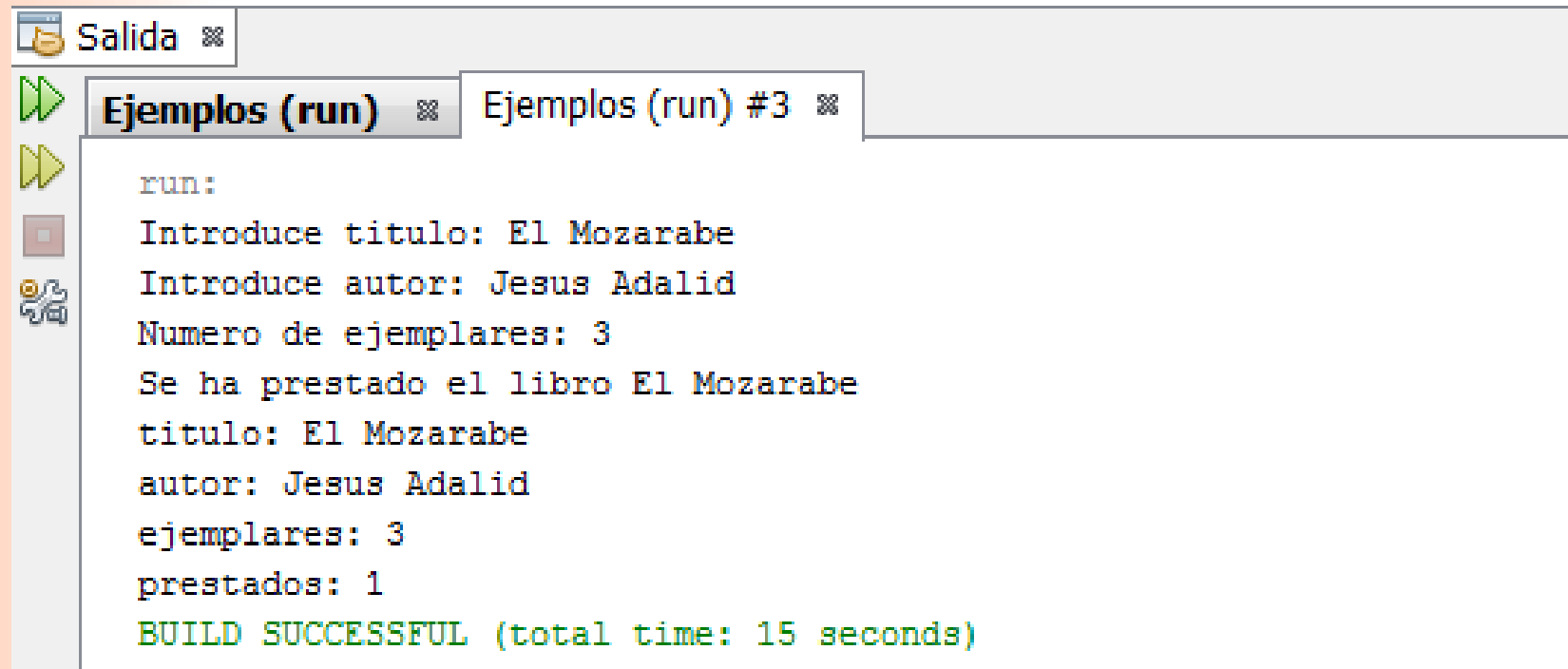
    //Se realiza un préstamo de libro1.
    if (libro1.prestar()) {
        System.out.println("Se ha prestado el libro " + libro1.getTitulo());
    } else {
        System.out.println("No quedan ejemplares de " + libro1.getTitulo() + " para prestar");
    }

    System.out.println(libro1.toString());
}
```



Ejemplo: Clase Libro

Resultado de la ejecución:



```
run:
Introduce titulo: El Mozarabe
Introduce autor: Jesus Adalid
Numero de ejemplares: 3
Se ha prestado el libro El Mozarabe
titulo: El Mozarabe
autor: Jesus Adalid
ejemplares: 3
prestados: 1
BUILD SUCCESSFUL (total time: 15 seconds)
```