

(5) Aprende Programación en Java - Y... (1) Video | Facebook Clases java

miumg.instructure.com/courses/75987/assignments/1213249

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help ejercicio1 - Apache NetBeans IDE 18 Search (Ctrl+I)

Projects

- ejercicio1
 - Source Packages
 - Vehículo
 - Taller.java
 - Vehículo.java
 - Test Packages
 - Dependencies
 - Java Dependencies
 - Project Files
- ejercicio2
 - Source Packages
 - Monedero
 - Monedero.java
 - Dependencies
 - Java Dependencies
 - Project Files
- ejercicio3
 - Source Packages
 - Heroe
 - Heroe.java
 - Test Packages
 - Dependencies
 - Java Dependencies
 - Project Files

Start Page x Taller.java x Vehículo.java x Monedero.java x Heroe.java x

Source History

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5  package Vehiculo;
6
7  /**
8   *
9   * @author Daniel Xocol
10  */
11  public class Taller {
12      public static void main(String[] args) {
13          Vehiculo vehiculo1 = new Vehiculo(marca: "Toyota", modelo: "Corolla", año: 2022, color: "Blanco", precio: 25000.0);
14          Vehiculo vehiculo2 = new Vehiculo(marca: "Honda", modelo: "Civic", año: 2021, color: "Gris", precio: 22000.0);
15          Vehiculo vehiculo3 = new Vehiculo(marca: "Ford", modelo: "Focus", año: 2020, color: "Negro", precio: 20000.0);
16          Vehiculo vehiculo4 = new Vehiculo(marca: "Chevrolet", modelo: "Cruze", año: 2022, color: "Rojo", precio: 23000.0);
17          Vehiculo vehiculo5 = new Vehiculo(marca: "Nissan", modelo: "Sentra", año: 2022, color: "Azul", precio: 24000.0);
18
19          vehiculo1.mostrarDetalles();
20          System.out.println();
21          vehiculo2.mostrarDetalles();
22          System.out.println();
23          vehiculo3.mostrarDetalles();
24          System.out.println();
25          vehiculo4.mostrarDetalles();
26          System.out.println();
27          vehiculo5.mostrarDetalles();
28      }
29  }
30  }
```

Output 1:1 INS Windows (CRLF)

19°C Parc. soleado 03:13 p.m. 04/08/2023

```

/*
package Vehiculo;

/**
 *
 * @author Daniel Xocol
 */
class Vehículo {
    private String marca;
    private String modelo;
    private int año;
    private String color;
    private double precio;

    /**
     *
     * @param marca
     * @param modelo
     * @param año
     * @param color
     * @param precio
     */
    //Constructor//
    public Vehículo(String marca, String modelo, int año, String color, double precio) {
        this.marca = marca;
        this.modelo = modelo;
        this.año = año;
        this.color = color;
        this.precio = precio;
    }

    //Metodos
    public String getMarca() {
        return marca;
    }
}

```

```
//Metodos
```

```
public String getMarca() {  
    return marca;  
}
```

```
public void setMarca(String marca) {  
    this.marca = marca;  
}
```

```
public String getModelo() {  
    return modelo;  
}
```

```
public void setModelo(String modelo) {  
    this.modelo = modelo;  
}
```

```
public int getAño() {  
    return año;  
}
```

```
public void setAño(int año) {  
    this.año = año;  
}
```

```
public String getColor() {  
    return color;  
}
```

```
public void setColor(String color) {  
    this.color = color;  
}
```

```
public void setColor(String color) {  
    this.color = color;  
}
```

```
public double getPrecio() {  
    return precio;  
}
```

```
public void setPrecio(double precio) {  
    this.precio = precio;  
}
```

```
public void mostrarDetalles() {  
    System.out.println("Marca: " + marca);  
    System.out.println("Modelo: " + modelo);  
    System.out.println("Año: " + año);  
    System.out.println("Color: " + color);  
    System.out.println("Precio: " + precio);  
}
```

```
}
```

Source

History



```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5   package Monedero;
6
7   /**
8    *
9    * @author Daniel Xocol
10   */
11   public class Monedero {
12       private double cantidad;
13
14       // Constructor para inicializar la cantidad de dinero en el monedero
15       public Monedero(double cantidadInicial) {
16           if (cantidadInicial >= 0) {
17               this.cantidad = cantidadInicial;
18           } else {
19               System.out.println(x: "La cantidad inicial no puede ser negativa. Se establecerá a cero.");
20               this.cantidad = 0;
21           }
22       }
23
24       // Método para agregar dinero al monedero
25       public void meterDinero(double cantidad) {
26           if (cantidad > 0) {
27               this.cantidad += cantidad;
28               System.out.println(cantidad + " unidades de dinero han sido agregadas al monedero.");
29           } else {
30               System.out.println(x: "La cantidad a agregar debe ser positiva.");
31           }
32       }
33
34       // Método para retirar dinero del monedero
35       public void sacarDinero(double cantidad) {
36           if (cantidad > 0 && cantidad <= this.cantidad) {
```

Monedero.Monedero > consultarDisponible >

```
// Método para retirar dinero del monedero
public void sacarDinero(double cantidad) {
    if (cantidad > 0 && cantidad <= this.cantidad) {
        this.cantidad -= cantidad;
        System.out.println(cantidad + " unidades de dinero han sido retiradas del monedero.");
    } else {
        System.out.println(x: "La cantidad a retirar debe ser positiva y no puede exceder el saldo actual.");
    }
}

// Método para consultar el saldo actual en el monedero
public double consultarDisponible() {
    return cantidad;
}
}
```

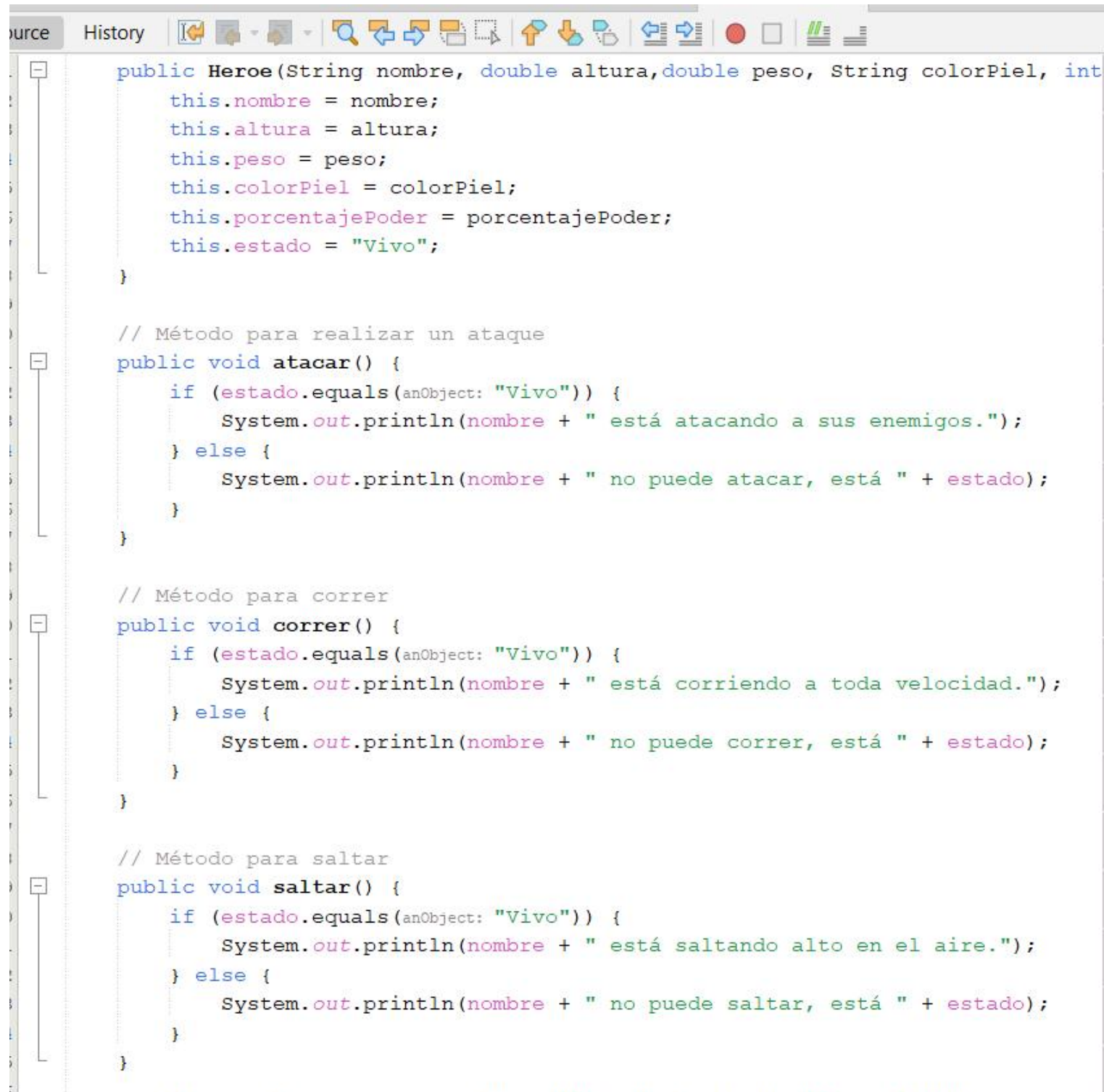


```
/*
 * Click nbfs:///nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs:///nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package Heroe;
```

```
/**
 *
 * @author Daniel Xocol
 */
public class Heroe {
    // Atributos
    private String nombre;
    private double altura;
    private double peso;
    private String colorPiel;
    private int porcentajePoder;
    private String estado;

    // Constructor para crear el personaje

    /**
     *
     * @param nombre
     * @param altura
     * @param peso
     * @param colorPiel
     * @param porcentajePoder
     * @param estado
     */
    public Heroe(String nombre, double altura, double peso, String colorPiel, int porcentajePoder, String estado) {
        this.nombre = nombre;
        this.altura = altura;
        this.peso = peso;
        this.colorPiel = colorPiel;
        this.porcentajePoder = porcentajePoder;
    }
}
```



The image shows a screenshot of an IDE window with a toolbar at the top and a code editor below. The toolbar contains various icons for file operations, navigation, and development tools. The code editor displays the following Java code:

```
public Heroe(String nombre, double altura, double peso, String colorPiel, int
    this.nombre = nombre;
    this.altura = altura;
    this.peso = peso;
    this.colorPiel = colorPiel;
    this.porcentajePoder = porcentajePoder;
    this.estado = "Vivo";
}

// Método para realizar un ataque
public void atacar() {
    if (estado.equals(anObject: "Vivo")) {
        System.out.println(nombre + " está atacando a sus enemigos.");
    } else {
        System.out.println(nombre + " no puede atacar, está " + estado);
    }
}

// Método para correr
public void correr() {
    if (estado.equals(anObject: "Vivo")) {
        System.out.println(nombre + " está corriendo a toda velocidad.");
    } else {
        System.out.println(nombre + " no puede correr, está " + estado);
    }
}

// Método para saltar
public void saltar() {
    if (estado.equals(anObject: "Vivo")) {
        System.out.println(nombre + " está saltando alto en el aire.");
    } else {
        System.out.println(nombre + " no puede saltar, está " + estado);
    }
}
```



```
// Método para esquivar
public void esquivar() {
    if (estado.equals(anObject: "Vivo")) {
        System.out.println(nombre + " está esquivando los ataques enemigos.");
    } else {
        System.out.println(nombre + " no puede esquivar, está " + estado);
    }
}
```

```
// Método para simular la muerte del personaje
public void morir() {
    estado = "Muerto";
    System.out.println(nombre + " ha muerto en la batalla.");
}
```

```
// Método para obtener el estado actual del personaje
public String obtenerEstado() {
    return estado;
}
```

```
public static void main(String[] args) {
    Heroe miPersonaje = new Heroe("Héroe", 1.75, 75, "Blanca", 90);
    miPersonaje.atacar();
    miPersonaje.correr();
    miPersonaje.saltar();
    miPersonaje.esquivar();
    miPersonaje.morir();
    miPersonaje.atacar();
}
```

```
}
```