

Programación Orientada a Objetos

Objetivo

Familiarizarse con los conceptos básicos de la programación orientada a objetos (POO) mediante la definición de clases, objetos, métodos y tipos de métodos, así como la instanciación y uso de objetos en programas funcionales.

Instrucciones

1. **Herramientas requeridas:** Un editor de código o IDE que soporte Java.
 2. **Formato del archivo:** Guarda los programas con extensión `.java`.
 3. **Lenguaje sugerido:** Java.
-

Actividad 1: Definición de Clases y Atributos

Objetivo: Comprender la estructura básica de una clase y sus atributos.

Instrucciones:

1. Crea una clase llamada `Coche` con los siguientes atributos:
 - `marca` (tipo `String`)
 - `modelo` (tipo `String`)
 - `anio` (tipo `int`)
 - `color` (tipo `String`)
2. Agrega un método llamado `mostrarDetalles` que imprima los valores de los atributos en un formato claro.

Desarrollo:

```
public class Coche {  
    String marca;  
    String modelo;  
    int anio;  
    String color;  
  
    public void mostrarDetalles() {  
        System.out.println("Marca: " + marca);  
        System.out.println("Modelo: " + modelo);  
        System.out.println("Año: " + anio);  
        System.out.println("Color: " + color);  
    }  
}
```

Actividad 2: Tipos de Métodos

Objetivo: Diferenciar entre métodos con y sin parámetros, así como métodos que retornan valores.

Instrucciones:

1. En la clase `Coche`, agrega los siguientes métodos:
 - `acelerar()`: Imprime un mensaje indicando que el coche está acelerando.
 - `cambiarColor(String nuevoColor)`: Cambia el valor del atributo `color` y muestra un mensaje indicando el cambio.
 - `calcularAntigüedad(int anioActual)`: Devuelve la antigüedad del coche calculada a partir de su año de fabricación.

Desarrollo:

```
public class Coche {
    String marca;
    String modelo;
    int anio;
    String color;

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

    public void acelerar() {
        System.out.println("El coche está acelerando...");
    }

    public void cambiarColor(String nuevoColor) {
        color = nuevoColor;
        System.out.println("El color del coche ha sido cambiado a: " + color);
    }

    public int calcularAntigüedad(int anioActual) {
        return anioActual - anio;
    }
}
```

Actividad 3: Instanciación y Uso de Objetos

Objetivo: Practicar la creación y manipulación de objetos.

Instrucciones:

1. Crea una clase llamada `PruebaCoche` con el método `main`.
2. En el método `main`:
 - Instancia un objeto de la clase `Coche`.

- Asigna valores a sus atributos.
- Llama a los métodos definidos en la clase para probar su funcionalidad.

Desarrollo:

```
public class PruebaCoche {
    public static void main(String[] args) {
        // Crear un objeto de la clase Coche
        Coche miCoche = new Coche();

        // Asignar valores a los atributos
        miCoche.marca = "Toyota";
        miCoche.modelo = "Corolla";
        miCoche.anio = 2015;
        miCoche.color = "Rojo";

        // Mostrar detalles del coche
        miCoche.mostrarDetalles();

        // Llamar a los métodos
        miCoche.acelerar();
        miCoche.cambiarColor("Azul");
        int antigüedad = miCoche.calcularAntigüedad(2025);
        System.out.println("La antigüedad del coche es: " + antigüedad + "
años.");
    }
}
```

Resultado Esperado: El programa imprime los detalles del coche, muestra mensajes al acelerar y cambiar el color, y calcula su antigüedad.

Actividad 4: Programas con Atributos y Métodos

Objetivo: Crear una clase más compleja con atributos adicionales y métodos personalizados.

Instrucciones:

1. Crea una clase llamada **Libro** con los siguientes atributos:
 - **titulo** (tipo **String**)
 - **autor** (tipo **String**)
 - **paginas** (tipo **int**)
 - **leido** (tipo **boolean**)
2. Agrega los siguientes métodos:
 - **mostrarInfo()**: Imprime los detalles del libro.
 - **marcarComoLeido()**: Cambia el atributo **leido** a **true** y muestra un mensaje confirmándolo.
 - **calcularTiempoLectura(int paginasPorHora)**: Calcula y devuelve las horas necesarias para leer el libro.

Desarrollo:

```

public class Libro {
    String titulo;
    String autor;
    int paginas;
    boolean leido;

    public void mostrarInfo() {
        System.out.println("Título: " + titulo);
        System.out.println("Autor: " + autor);
        System.out.println("Páginas: " + paginas);
        System.out.println("Leído: " + (leido ? "Sí" : "No"));
    }

    public void marcarComoLeido() {
        leido = true;
        System.out.println("El libro ha sido marcado como leído.");
    }

    public int calcularTiempoLectura(int paginasPorHora) {
        return paginas / paginasPorHora;
    }
}

```

Prueba:

```

public class PruebaLibro {
    public static void main(String[] args) {
        Libro libro = new Libro();
        libro.titulo = "El Quijote";
        libro.autor = "Miguel de Cervantes";
        libro.paginas = 900;
        libro.leido = false;

        libro.mostrarInfo();
        libro.marcarComoLeido();
        int tiempoLectura = libro.calcularTiempoLectura(100);
        System.out.println("Tiempo estimado de lectura: " + tiempoLectura + "
horas.");
    }
}

```

Resultado Esperado: El programa imprime los detalles del libro, actualiza su estado a "leído" y calcula el tiempo de lectura según las páginas por hora proporcionadas.

Mas Ejercicios

Ejercicio 1: Clase "Persona"

Instrucciones:

1. Crea una clase llamada **Persona** con los siguientes atributos:
 - **nombre** (tipo **String**)
 - **edad** (tipo **int**)
 - **genero** (tipo **char**)
2. Agrega un método llamado **mostrarInformacion()** que imprima los valores de los atributos.
3. En el método **main** de otra clase llamada **PruebaPersona**, crea un objeto de la clase **Persona**, asigna valores a sus atributos e invoca el método **mostrarInformacion()**.

Desarrollo:

```
public class Persona {
    String nombre;
    int edad;
    char genero;

    public void mostrarInformacion() {
        System.out.println("Nombre: " + nombre);
        System.out.println("Edad: " + edad);
        System.out.println("Género: " + genero);
    }
}

public class PruebaPersona {
    public static void main(String[] args) {
        Persona persona = new Persona();
        persona.nombre = "María";
        persona.edad = 30;
        persona.genero = 'F';

        persona.mostrarInformacion();
    }
}
```

Ejercicio 2: Clase "Rectángulo"

Instrucciones:

1. Crea una clase llamada **Rectangulo** con los siguientes atributos:
 - **base** (tipo **double**)
 - **altura** (tipo **double**)
2. Agrega dos métodos:
 - **calcularArea()**: Devuelve el área del rectángulo.

- `calcularPerimetro()`: Devuelve el perímetro del rectángulo.
3. En el método `main` de una clase llamada `PruebaRectangulo`, crea un objeto de la clase `Rectangulo`, asigna valores a sus atributos y llama a los métodos para calcular e imprimir el área y el perímetro.

Desarrollo:

```
public class Rectangulo {
    double base;
    double altura;

    public double calcularArea() {
        return base * altura;
    }

    public double calcularPerimetro() {
        return 2 * (base + altura);
    }
}

public class PruebaRectangulo {
    public static void main(String[] args) {
        Rectangulo rectangulo = new Rectangulo();
        rectangulo.base = 5;
        rectangulo.altura = 3;

        System.out.println("Área: " + rectangulo.calcularArea());
        System.out.println("Perímetro: " + rectangulo.calcularPerimetro());
    }
}
```

Ejercicio 3: Clase "Producto"

Instrucciones:

1. Crea una clase llamada `Producto` con los siguientes atributos:
 - `nombre` (tipo `String`)
 - `precio` (tipo `double`)
 - `cantidad` (tipo `int`)
2. Agrega un método llamado `calcularTotal()` que devuelva el costo total del producto (`precio * cantidad`).
3. En el método `main` de una clase llamada `PruebaProducto`, crea un objeto de la clase `Producto`, asigna valores a sus atributos y llama al método para calcular el total, imprimiéndolo.

Desarrollo:

```
public class Producto {
    String nombre;
```

```

    double precio;
    int cantidad;

    public double calcularTotal() {
        return precio * cantidad;
    }
}

public class PruebaProducto {
    public static void main(String[] args) {
        Producto producto = new Producto();
        producto.nombre = "Laptop";
        producto.precio = 1200.50;
        producto.cantidad = 2;

        System.out.println("Producto: " + producto.nombre);
        System.out.println("Total a pagar: " + producto.calcularTotal());
    }
}

```

Ejercicio 4: Clase "Círculo"

Instrucciones:

1. Crea una clase llamada **Círculo** con los siguientes atributos:
 - **radio** (tipo **double**)
2. Agrega dos métodos:
 - **calcularArea()**: Devuelve el área del círculo ($\pi * \text{radio}^2$).
 - **calcularCircunferencia()**: Devuelve la circunferencia del círculo ($2 * \pi * \text{radio}$).
3. En el método **main** de una clase llamada **PruebaCírculo**, crea un objeto de la clase **Círculo**, asigna valores a su atributo y llama a los métodos para calcular e imprimir el área y la circunferencia.

Desarrollo:

```

public class Circulo {
    double radio;

    public double calcularArea() {
        return Math.PI * Math.pow(radio, 2);
    }

    public double calcularCircunferencia() {
        return 2 * Math.PI * radio;
    }
}

public class PruebaCirculo {
    public static void main(String[] args) {
        Circulo circulo = new Circulo();
    }
}

```

```

        circulo.radio = 4.5;

        System.out.println("Área: " + circulo.calcularArea());
        System.out.println("Circunferencia: " + circulo.calcularCircunferencia());
    }
}

```

Ejercicio 5: Clase "Cuenta"

Instrucciones:

1. Crea una clase llamada **Cuenta** con los siguientes atributos:
 - **numeroCuenta** (tipo **String**)
 - **saldo** (tipo **double**)
2. Agrega dos métodos:
 - **depositar(double monto)**: Incrementa el saldo.
 - **retirar(double monto)**: Disminuye el saldo si hay suficiente dinero.
3. En el método **main** de una clase llamada **PruebaCuenta**, crea un objeto de la clase **Cuenta**, realiza varias operaciones de depósito y retiro, y muestra el saldo final.

Desarrollo:

```

public class Cuenta {
    String numeroCuenta;
    double saldo;

    public void depositar(double monto) {
        saldo += monto;
        System.out.println("Depósito realizado. Saldo actual: " + saldo);
    }

    public void retirar(double monto) {
        if (monto <= saldo) {
            saldo -= monto;
            System.out.println("Retiro realizado. Saldo actual: " + saldo);
        } else {
            System.out.println("Saldo insuficiente.");
        }
    }
}

public class PruebaCuenta {
    public static void main(String[] args) {
        Cuenta cuenta = new Cuenta();
        cuenta.numeroCuenta = "00123456789";
        cuenta.saldo = 1000;

        cuenta.depositar(500);
        cuenta.retirar(200);
    }
}

```



```

        cuenta.retirar(1500);
    }
}

```

Creación de Programas con Clases, Atributos y Métodos

Conceptos Clave

En la programación orientada a objetos, una **clase** es un modelo que define atributos (características) y métodos (acciones) de un objeto. Un **objeto** es una instancia de una clase.

Ejemplo de una Clase en Java

```

// Definición de la clase Producto
class Producto {
    // Atributos
    String nombre;
    double precio;
    int cantidad;

    // Constructor
    public Producto(String nombre, double precio, int cantidad) {
        this.nombre = nombre;
        this.precio = precio;
        this.cantidad = cantidad;
    }

    // Método para calcular el valor total del stock
    public double calcularValorStock() {
        return precio * cantidad;
    }

    // Método para mostrar información del producto
    public void mostrarInformacion() {
        System.out.println("Producto: " + nombre + ", Precio: " + precio + ",
Cantidad: " + cantidad);
    }
}

```

Creación e Instanciación de Objetos

```

public class Main {
    public static void main(String[] args) {
        // Creación de un objeto de la clase Producto
        Producto producto1 = new Producto("Laptop", 1200.50, 5);
    }
}

```

```

        // Llamada a un método del objeto
        producto1.mostrarInformacion();
        System.out.println("Valor del stock: " + producto1.calcularValorStock());
    }
}

```

Ejercicio 6: Clase Coche

```

class Coche {
    String marca;
    String modelo;
    int anio;
    int velocidad;

    public Coche(String marca, String modelo, int anio) {
        this.marca = marca;
        this.modelo = modelo;
        this.anio = anio;
        this.velocidad = 0;
    }

    public void acelerar(int incremento) {
        velocidad += incremento;
    }

    public void frenar(int decremento) {
        velocidad = Math.max(0, velocidad - decremento);
    }

    public void mostrarDetalles() {
        System.out.println("Marca: " + marca + ", Modelo: " + modelo + ", Año: " +
        anio + ", Velocidad: " + velocidad + " km/h");
    }
}

```

Ejercicio 7: Clase CuentaBancaria

```

class CuentaBancaria {
    String titular;
    double saldo;

    public CuentaBancaria(String titular, double saldo) {
        this.titular = titular;
        this.saldo = saldo;
    }

    public void depositar(double cantidad) {
        saldo += cantidad;
    }
}

```

```

    public void retirar(double cantidad) {
        if (cantidad <= saldo) {
            saldo -= cantidad;
        } else {
            System.out.println("Saldo insuficiente");
        }
    }

    public void mostrarSaldo() {
        System.out.println("Titular: " + titular + ", Saldo: " + saldo);
    }
}

```

Ejercicio 8: Clase Estudiante

```

class Estudiante {
    String nombre;
    int edad;
    double[] calificaciones;

    public Estudiante(String nombre, int edad, double[] calificaciones) {
        this.nombre = nombre;
        this.edad = edad;
        this.calificaciones = calificaciones;
    }

    public double calcularPromedio() {
        double suma = 0;
        for (double nota : calificaciones) {
            suma += nota;
        }
        return suma / calificaciones.length;
    }

    public void mostrarInformacion() {
        System.out.println("Nombre: " + nombre + ", Edad: " + edad + ", Promedio: " + calcularPromedio());
    }
}

```

Ejercicio 9: Clase Restaurante

```

class Restaurante {
    String nombre;
    String tipoComida;
    double calificacion;
    int capacidad;
}

```

```
public Restaurante(String nombre, String tipoComida, double calificacion, int
capacidad) {
    this.nombre = nombre;
    this.tipoComida = tipoComida;
    this.calificacion = calificacion;
    this.capacidad = capacidad;
}

public boolean verificarDisponibilidad(int personas) {
    return personas <= capacidad;
}
}
```

Ejercicios

1. **Clase Película:** Implementa una clase `Pelicula` con `titulo`, `director` y `duracion`. Incluye un método para mostrar la duración en horas y minutos.
2. **Clase Tienda:** Crea una clase `Tienda` con `nombre`, `ubicacion` y `inventario`. Implementa un método para agregar productos al inventario.
3. **Clase Vehículo:** Define una clase `Vehiculo` con `tipo`, `color` y `velocidadMaxima`. Agrega métodos para acelerar y frenar.
4. **Clase Empleado:** Implementa una clase `Empleado` con `nombre`, `salario` y `puesto`. Agrega un método para calcular el salario anual con bonificaciones.
5. **Clase Animal:** Crea una clase `Animal` con `especie`, `peso` y `habitat`. Incluye métodos para alimentar y cambiar el hábitat del animal.