Informe Técnico: Sistema de Gestión de Cursos

Luis Carlos Cortez Guzmán

Introducción

El presente informe detalla la creación de un sistema de gestión de cursos para un instituto educativo, utilizando la programación orientada a objetos en Java. El sistema permite manejar información de diferentes tipos de cursos: `CursoOnline`, `CursoPresencial` y `CursoHibrido`.

Estructura del Programa

Clase Padre: Curso

La clase `Curso` es la clase base que contiene los atributos comunes a todos los tipos de cursos. Estos atributos son `nombre`, `profesor` y `capacidad`.

Atributos

- `nombre` (String): Nombre del curso.

- `profesor` (String): Nombre del profesor que imparte el curso.

- `capacidad` (int): Número máximo de estudiantes permitidos en el curso.

Métodos

- **Constructor**: Inicializa los atributos `nombre`, `profesor` y `capacidad`.

- **Getters y Setters**: Métodos para acceder y modificar cada uno de los atributos.

- **mostrarDetalles()**: Método que devuelve una cadena con la información del curso.

Clases Hijas

Cada clase hija extiende la clase `Curso` y agrega atributos y funcionalidades específicas.

CursoOnline

- **Atributo adicional**: `url` (String) - Dirección web donde se imparte el curso.

- **Constructor**: Inicializa los atributos de la clase padre y el atributo `url`.

- **mostrarDetalles()**: Método sobrescrito para incluir la URL del curso en la información mostrada.

CursoPresencial

- **Atributo adicional**: `sala` (String) - Sala donde se imparte el curso presencial.

- **Constructor**: Inicializa los atributos de la clase padre y el atributo `sala`.

- **mostrarDetalles()**: Método sobrescrito para incluir la sala del curso en la información mostrada.

CursoHibrido

- **Atributos adicionales**: `url` (String) y `sala` (String) - Dirección web y sala donde se imparte el curso híbrido.

- **Constructor**: Inicializa los atributos de la clase padre y los atributos `url` y `sala`.

- **mostrarDetalles()**: Método sobrescrito para incluir tanto la URL como la sala del curso en la información mostrada.

Código Fuente

A continuación, se presenta el código completo de las clases `Curso`, `CursoOnline`, `CursoPresencial` y `CursoHibrido`, así como una clase `Main` para probar su funcionamiento.

Cursos > Curso.java > ...

```java
package Cursos;

public class Curso {
    private String nombre;
    private String profesor;
    private int capacidad;

    // Constructor
    public Curso(String nombre, String profesor, int capacidad) {...

    // Getters y Setters
    public String getNombre() {...

    public void setNombre(String nombre) {...

    public String getProfesor() {...

    public void setProfesor(String profesor) {...

    public int getCapacidad() {...

    public void setCapacidad(int capacidad) {...

    // Método para mostrar detalles
    public String mostrarDetalles() {
        return "Curso: " + nombre + "\nProfesor: " + profesor + "\nCapacidad: " + capacidad;
    }
}
```

Cursos > CursoOnline.java > ...

```java
package Cursos;

public class CursoOnline extends Curso{
    private String url;

    // Constructor
    public CursoOnline(String nombre, String profesor, int capacidad, String url) {
        super(nombre, profesor, capacidad);
        this.url = url;
    }

    // Getter y Setter para URL
    public String getUrl() {
        return url;
    }

    public void setUrl(String url) {
        this.url = url;
    }

    // Override del método mostrarDetalles
    @Override
    public String mostrarDetalles() {
        return super.mostrarDetalles() + "\nURL: " + url;
    }
}
```

Cursos > CursoPresencial.java > ...

```java
package Cursos;

public class CursoPresencial extends Curso{
    private String sala;

    // Constructor
    public CursoPresencial(String nombre, String profesor, int capacidad, String sala) {
        super(nombre, profesor, capacidad);
        this.sala = sala;
    }

    // Getter y Setter para Sala
    public String getSala() {
        return sala;
    }

    public void setSala(String sala) {
        this.sala = sala;
    }

    // Override del método mostrarDetalles
    @Override
    public String mostrarDetalles() {
        return super.mostrarDetalles() + "\nSala: " + sala;
    }
}
```

Cursos > CursoHibrido.java > ...

```java
package Cursos;

public class CursoHibrido extends Curso{
    private String url;
    private String sala;

    // Constructor
    public CursoHibrido(String nombre, String profesor, int capacidad, String url, String sala) {...

    // Getters y Setters para URL y Sala
    public String getUrl() {...

    public void setUrl(String url) {...

    public String getSala() {...

    public void setSala(String sala) {...

    // Override del método mostrarDetalles
    @Override
    public String mostrarDetalles() {
        return super.mostrarDetalles() + "\nURL: " + url + "\nSala: " + sala;
    }
}
```

Explicación de la Ejecución del Programa

1. Instanciación de Objetos:

- Se crean instancias de `CursoOnline`, `CursoPresencial` y `CursoHibrido` utilizando los constructores definidos en cada clase.

- Ejemplo: `CursoOnline cursoOnline = new CursoOnline("Programación en Java", "Juan Pérez", 100, "http://cursojava.com");`

2. Llamada a Métodos:

- Se llama al método `mostrarDetalles()` para cada instancia creada.

- Ejemplo: `System.out.println(cursoOnline.mostrarDetalles());`

3. Salida del Programa:

- El método `mostrarDetalles()` de cada clase hija sobrescribe el método de la clase padre `Curso` para incluir información específica (URL y sala).

- Se imprime la información detallada de cada curso, mostrando el nombre, profesor, capacidad, y los atributos adicionales según el tipo de curso.

Ejemplo de Salida

```
Curso: Programación
Profesor: Walter mata
Capacidad: 40
URL: http://cursojava.com

Curso: Algebra
Profesor: Elizabeth santiago
Capacidad: 30
Sala: Sala 2B

Curso: Ingles
Profesor: Luis Benavides
Capacidad: 50
URL: http://cursoingles.com
Sala: Sala 2B
```

Conclusión

Este sistema de gestión de cursos permite manejar eficientemente la información de diferentes tipos de cursos, aprovechando la herencia y el polimorfismo en Java. La estructura del programa es extensible, lo que facilita la adición de nuevos tipos de cursos en el futuro, si es necesario.