

PRÁCTICA: CLASES ABSTRACTAS

Nombre del Estudiante: Jasciel Adrian Cervantes Couoh

Fecha: 17/Octubre/2024

Curso: Lenguaje de programación II

Introducción

En esta práctica, aprendimos sobre la abstracción en programación y cómo implementar tareas periódicas en Java. Creamos un programa que puede ejecutar comandos y mostrar la hora actual utilizando clases y métodos.

Creación de la Clase **TareaPeriodica**:

- Creamos una clase llamada **TareaPeriodica**, que es abstracta. Esto significa que no se puede crear un objeto directamente de esta clase, sino que sirve como base para otras clases.
- En esta clase, definimos dos métodos abstractos: **necesitaEjecucion()** y **ejecutarTarea()**. Estos métodos no tienen implementación en **TareaPeriodica**, pero deben ser implementados en las clases que heredan de ella.

Implementación de la Clase **Ejecutor**:

- Creamos la clase **Ejecutor**, que extiende de **TareaPeriodica**. Esto significa que **Ejecutor** hereda las características de **TareaPeriodica**.
- En **Ejecutor**, implementamos el método **necesitaEjecucion()**, que verifica si la tarea necesita ser ejecutada. También implementamos **ejecutarTarea()**, que se encarga de ejecutar un comando en el sistema.
- Agregamos un método **run()**, que se utiliza para ejecutar la tarea si es necesario.

Implementación de la Clase **Reloj**:

- Creamos otra clase llamada **Reloj**, que también extiende de **TareaPeriodica**.
- En **Reloj**, implementamos **necesitaEjecucion()** para comprobar si ha pasado el tiempo necesario desde la última ejecución. También

implementamos **ejecutarTarea()**, que muestra la hora actual en la consola.

Creación de la Clase Controladora:

- Finalmente, creamos una clase controladora que contiene el método **main()**. En este método, iniciamos dos hilos: uno para la clase **Reloj** y otro para la clase **Ejecutor**.
- Esto permite que ambas tareas se ejecuten de manera simultánea, mostrando la hora y ejecutando el comando que se le pase a **Ejecutor**.

Conclusiones

La práctica nos ayudó a entender mejor cómo funciona la programación orientada a objetos en Java, especialmente el concepto de abstracción. Aprendimos a crear clases abstractas y a implementar métodos en clases derivadas. También vimos cómo usar hilos para ejecutar tareas al mismo tiempo, lo que es muy útil en programación.