Uso de concurrencia en Kotlin

[crayon-6a9d68407a978895645906/]

La clase Executors en Java (y en este caso también en Kotlin) proporciona métodos estáticos para crear y administrar ejecutores de subprocesos. Los ejecutores de subprocesos son útiles cuando se necesita realizar tareas de forma asíncrona, en paralelo o en segundo plano.

Aquí hay algunas funcionalidades principales de la clase Executors:

1. Creación de ejecutores: Executors proporciona métodos estáticos para crear diferentes tipos de ejecutores, como ejecutores de subprocesos de un solo hilo (newSingleThreadExecutor()), ejecutores de subprocesos con un pool fijo (newFixedThreadPool(int nThreads)), ejecutores de subprocesos con pool variable (newCachedThreadPool()) y más.
2. Gestión de tareas concurrentes: Los ejecutores de subprocesos administran la ejecución concurrente de tareas. Permiten enviar tareas para su ejecución y se encargan de asignarlas a los subprocesos disponibles en el pool. Esto ayuda a aprovechar eficientemente los recursos del sistema y a ejecutar tareas en paralelo.
3. Control de concurrencia: Los ejecutores proporcionan un mecanismo para controlar y limitar la concurrencia de las tareas. Por ejemplo, un ejecutor de subprocesos con pool fijo permite especificar el número máximo de subprocesos disponibles, lo que puede ser útil para limitar la carga del sistema o administrar los recursos.
4. Programación asíncrona: Los ejecutores de subprocesos son útiles en situaciones donde se necesitan realizar operaciones largas o bloqueantes en segundo plano, evitando que el hilo principal del programa se bloquee.

Permiten ejecutar tareas en paralelo y asíncronamente, lo que puede mejorar la capacidad de respuesta y la eficiencia de las aplicaciones.