Try with resource y multicatch

Try with resources

El bloque de código try with resources se usa para declarar uno o más recursos que se deben cerrar al final del bloque de código, de forma que no sea necesario escribir código adicional para cerrar los recursos manualmente como lo podría ser código insertado en un finally. La sintaxis sería algo como esto:

```
try (ClaseAutocoseable autocoseables) {
    // código que usa el recurso
} catch (ExceptionType e) {
    // manejo de excepciones
}
```

En este bloque, después del catch entre paréntesis se encuntran las variables de tipo autocoseable. El recurso se cierra automáticamente al final del bloque, ya sea que se produzca una excepción o no.

El uso de "try with resources" puede reducir el código necesario para manejar recursos externos y garantizar que los recursos se cierren de manera segura y eficiente. Sin embargo, es importante tener en cuenta que esta característica solo está disponible en Java 7 y versiones posteriores.

Multicatch

El multicatch es una característica de Java introducida en Java 7 que permite capturar múltiples excepciones en un solo bloque catch. Antes de Java 7, cada excepción tenía que ser capturada en un bloque catch separado, lo que podía dar lugar a código redundante y difícil de mantener.

Con el multicatch, es posible especificar varias excepciones que deben ser capturadas en un solo bloque catch. La sintaxis es la siguiente:

```
try {
    // código que puede lanzar excepciones
} catch (ExceptionTypo1 | ExceptionTypo2 | ... | ExceptionTypoN e) {
    // manejo de excepciones
}
```

En este bloque, "ExceptionTypo1", "ExceptionTypo2", "...", "ExceptionTypoN" son las excepciones que se desean capturar en el mismo bloque catch. Cuando se lanza una excepción de cualquiera de estos tipos dentro del bloque try, se captura en el mismo bloque catch y se maneja de acuerdo con el código dentro del bloque catch.

El uso del multicatch puede reducir el código necesario para manejar múltiples excepciones y hacer que el código sea más legible y fácil de mantener. Sin embargo, es importante tener en cuenta que el multicatch solo debe utilizarse cuando se desea manejar excepciones de manera similar. Si se requiere un manejo de excepciones más específico para cada tipo de excepción, es necesario usar bloques catch separados.