

Conceptos: Daniel Barco

¿Cuáles son las acciones los tres momentos importantes de las excepciones?
¿Cuál es el objetivo de cada una? ¿Cómo se implementa en Java cada acción?

Los principales tipos de excepciones son las de tipo checked y unchecked de las que derivan las que pertenecen a los 3 momentos más importantes de las excepciones (Ya que se derivan más):
RuntimeException, IOException y Excepciones creadas, cada una tiene un objetivo diferente que se explicara a continuación:

RuntimeException, tiene como objetivo prevenir errores del programador básicos como lo son usar un array por fuera de los límites.

IOException, que abarca los errores que no puede evitar el programador, que son los relacionados a errores de inicio y salida en el programa.

Excepciones creadas, estas surgen de la herencia que permite Java, permitiendo crear excepciones con fines distintos a los predeterminados.

Para implementar una excepción se encuentra por medio del “try-catch”, que es un bloque encargado de detectar cuando ocurre una o más excepciones en una parte o más del código, de manera que al encontrarlas podamos usar “throw” para salir del bloque de código al ocurrir una excepción.

¿Qué es sobre-escritura de métodos? ¿Por qué aplicarla? ¿Cómo impedir que se sobre-escriba un método?

La sobre-escritura de métodos es atribuida a la herencia, esta consiste en redefinir un método heredado, haciendo que tenga una función distinta a la que ordena la superclase de la que este hereda.

Esta se aplica para redefinir un método que hereda de una superclase, se suele usar para cambiar la funcionalidad parcial de este mismo, aprovechando aun las características heredadas, pero usándolo para otra función.

La forma más fácil de impedir que se modifique un método es poniendo el atributo “final” a la superclase de la que hereda o al método deseado, de esta forma su contenido será constante y se evita editar sus valores por accidente.