The Java™ Tutorials

Trail: Essential Classes

Lesson: Exceptions

The Java programming language uses *exceptions* to handle errors and other exceptional events. This lesson describes when and how to use exceptions.

What Is an Exception?

An exception is an event that occurs during the execution of a program that disrupts the normal flow of instructions.

The Catch or Specify Requirement

This section covers how to catch and handle exceptions. The discussion includes the `try`, `catch`, and `finally` blocks, as well as chained exceptions and logging.

How to Throw Exceptions

This section covers the `throw` statement and the `Throwable` class and its subclasses.

The try-with-resources Statement

This section describes the `try`-with-resources statement, which is a `try` statement that declares one or more resources. A resource is as an object that must be closed after the program is finished with it. The `try`-with-resources statement ensures that each resource is closed at the end of the statement.

Unchecked Exceptions — The Controversy

This section explains the correct and incorrect use of the unchecked exceptions indicated by subclasses of `RuntimeException`.

Advantages of Exceptions

The use of exceptions to manage errors has some advantages over traditional error-management techniques. You'll learn more in this section.

Summary

Questions and Exercises

Previous page: Table of Contents
Next page: What Is an Exception?