

Creating Custom Exceptions



Jim Wilson

Mobile Solutions Developer & Architect

@hedgehogjim | jwhh.com



Overview



Throwing an exception

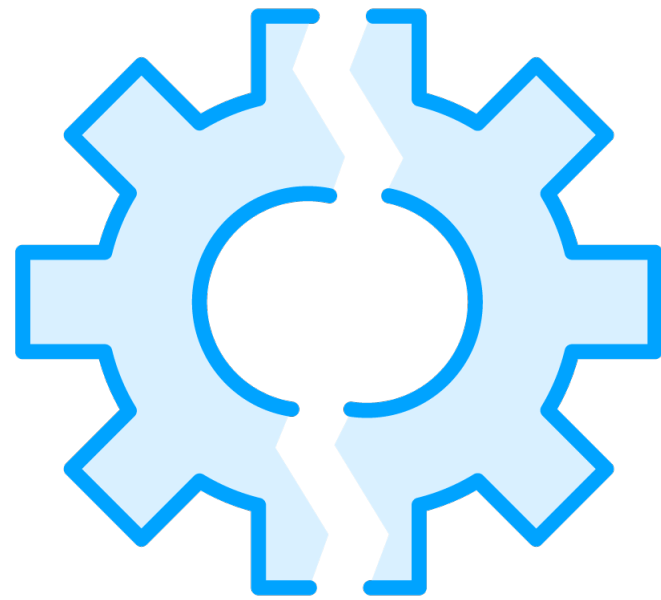
Creating a new exception instance

Defining a custom exception

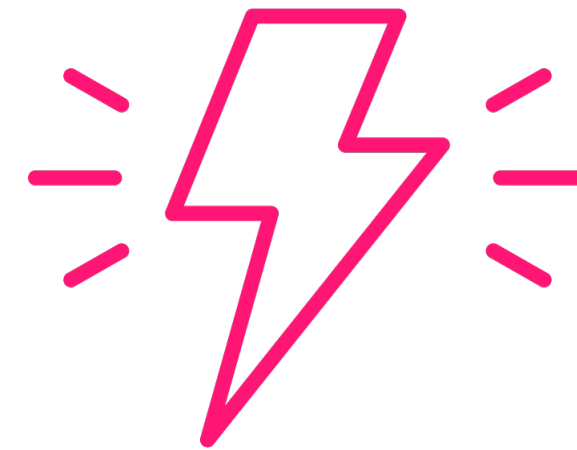
Chaining exceptions



Dealing with Errors



Responding to Issues That Occur
Catch exceptions



Indicating an Issue Has Occurred
Throw exceptions



Creating an Exception

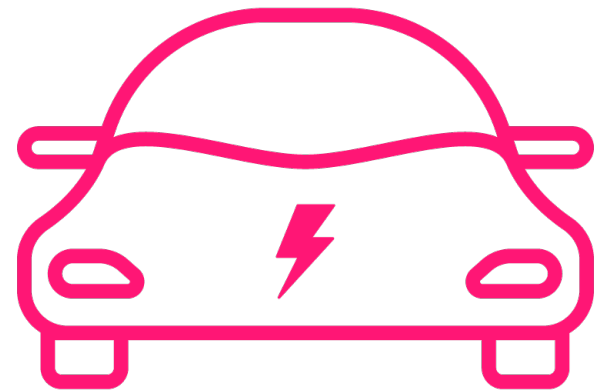


Create with new Operator
Remember exceptions are classes

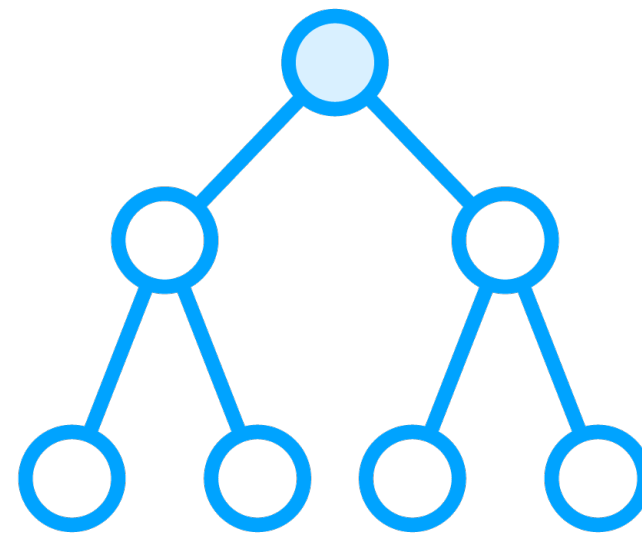


Include Information with Constructor
Normally include a string description
Some allow including additional detail

Custom Exceptions



Can Create Custom Exceptions
Define custom exception class



Inherit from Exception
Normally directly inherit
Provides most required features



Class Members to Add
Appropriate constructors
Other members if needed



Chaining Exceptions



Exceptions Can Be Chained

Allows one exception to wrap another

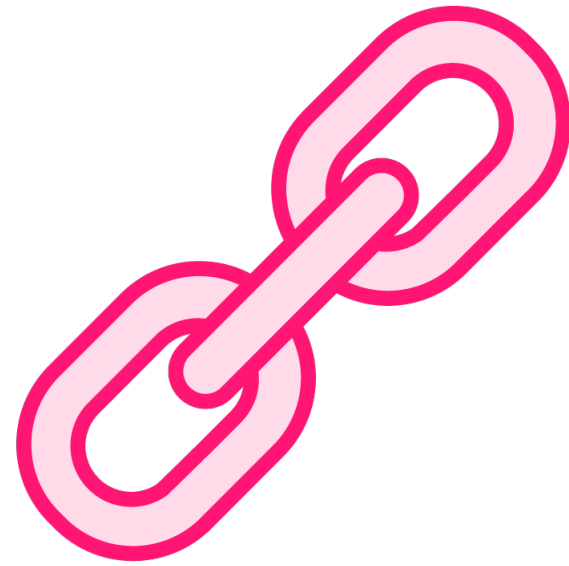


Why Chain Exceptions

Throw more meaningful exception
While preserving underlying exception

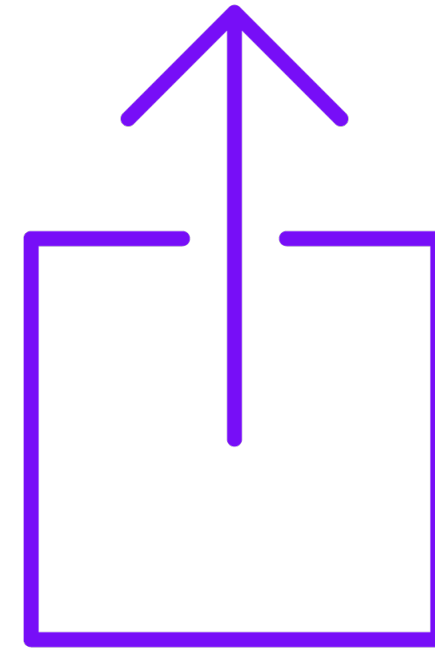


Chaining Exceptions



How to Chain Exceptions

Can use inherited `initCause` method
More commonly include constructor that
accepts original exception



Accessing Chained Exception

Use inherited `getCause` method

Summary



Throw exceptions to indicate error

- Use throw statement
- Must create exception instance first



Summary



Can define custom exception types

- Must inherit from Exception class

Add appropriate class members

- Normally add one or more constructors
- Can add other members if needed



Summary



Exceptions can be chained

- Throw more meaningful exception
- Allows preserving original exception

Chaining exceptions

- Normally passed to constructor
- Access with `getCause` method