

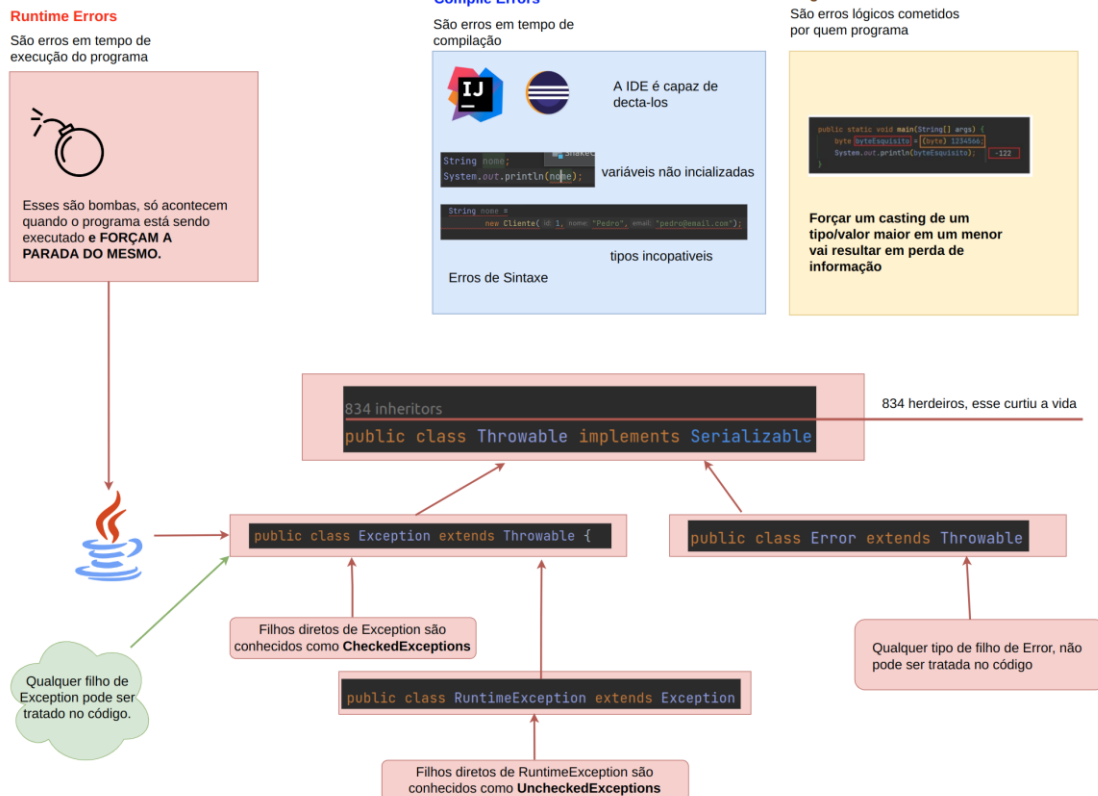


Exceptions – Java

Erros de programação no geral podem ser classificados em três tipos: “**Runtime, compile e logical**”.

Erros de Runtime são representados como uma classe de **Exception** no Java, sempre que uma Exception acontecer a JVM vai instanciar uma classe e **lançar a Exception**.

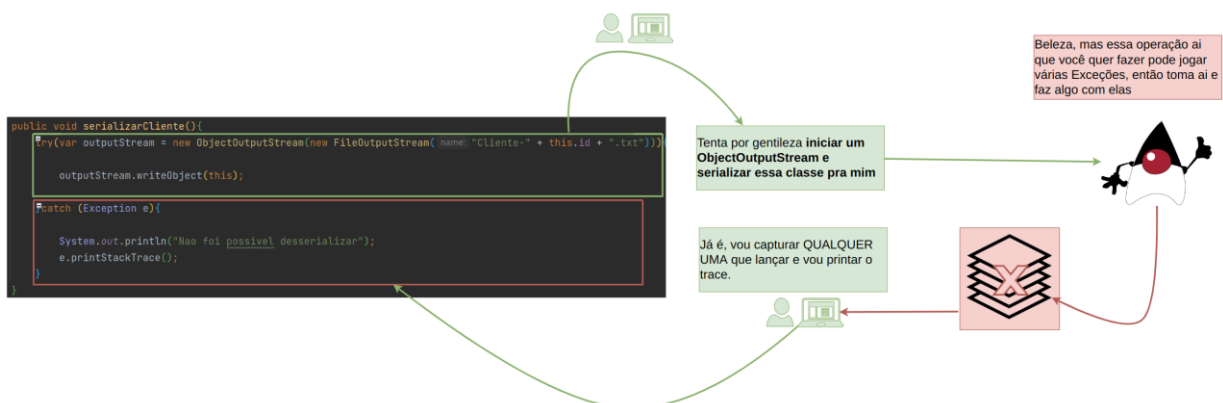
Existem diversas exceções prontas no Java, mas nada impede que criemos as nossas que podem inclusive estender dessas já existentes:



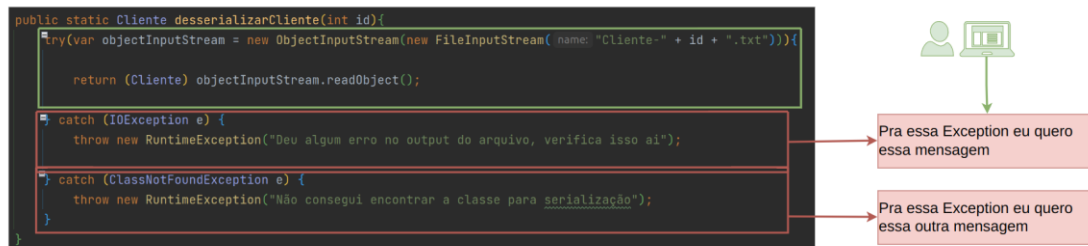
Tratar Exceptions – Java

O Java tem uma **sintaxe para lidar com exceptions** e também **algumas palavras chaves**.

Basicamente você tem um **bloco onde você tenta fazer uma operação** e **se der erro, pode capturá-la e tratá-la** ou **joga-la** para um método acima:



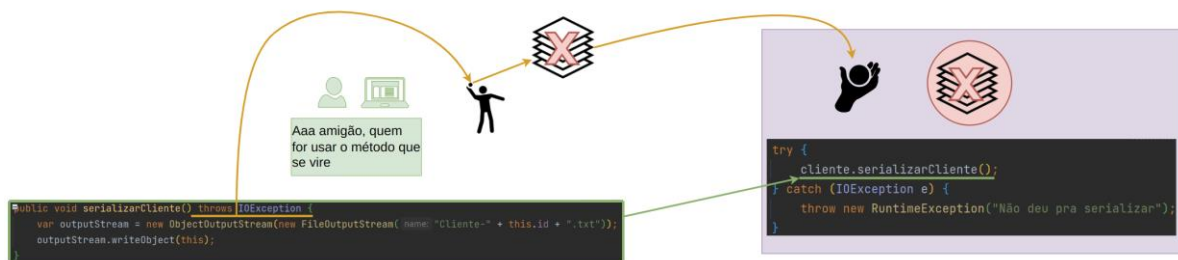
Exceptions são úteis para você entender o que fazer quando **determinado problema específico acontecer**, então você pode usar uma **estratégia diferente para cada tipo de exception** em uma **mesma tentativa**:



Trhow, Throws – Java]

Trhow

Existem dois modos de se lançar uma exceção, o primeiro deles é usando **throws** na **declaração do método**, você está dizendo: **“O que eu estou fazendo aqui dentro pode ter exceções, mas eu vou jogar elas pra quem me chamou, aí ele que se vire”**:



Se você **ficar jogando a exceção até chegar no método main**, então o handler da JVM vai lidar com ela, e vai usar a mensagem default que não é muito amigável.

Throws

O **Throw** é mais usado para **RuntimeExceptions (aquelas que não precisam ser necessariamente checadas pelo compilador)**, a ideia do **Throw** é simplesmente jogar a **exceção** direto pra JVM. Você não obriga quem tiver te chamando a tratar a exceção, embora seja possível.