

07 - Error Handling

Error Handling não deve confundir ou obscurecer a lógica do código.

Recomendações

- Usar Exceptions em vez de Return Codes:
 - Separa melhor funcionalidade do código.
 - Mais fácil do que lidar com o erro dentro do próprio corpo da função.
- Escrever o try-catch-finally antes de tudo:
 - Mais controle de o que pode acontecer com o código.
 - Facilita o desenvolvimento e a compreensão do fluxo do código.
 - Iterativamente: Forçar Exceções --> Lidar com elas.
- Usar Unchecked Exceptions:
 - Característica do Java (ausente no Python):
 - Checked excePtions: exceções checadas na compilação. Devem ser tratadas.
 - Unchecked exceptions: exceções checadas na execução. Podem ou não ser tratadas.
 - Checked exception violam princípio open/close.
 - Requerem declaração na assinatura -> ruim.
 - Pode gerar necessidade de criar uma cascata imensa de alterações.
- Dar contexto com as exceções.
 - Exceção deve prover informação sobre origem do erro.
 - Mensagens de erro devem ser informativas.
- Definir Classes de exceção a partir das necessidades do caller
 - Definição da exceção deve depender de como ela é capturada.
 - Não da fonte ou um tipo arbitrário.
 - Evitar vários catch em sequência para erros parecidos com o mesmo tratamento.
 - Melhor envelopar tudo numa única classe de exceção
 - É comum uma única classe ser o suficiente para uma região do código.
- Definir o fluxo normal
 - Evitar colocar parte da lógica dentro do tratamento de erro
 - Tratamento de casos especiais pode ser gerido por uma classe (Special Case Object)
 - Assim o código original não precisa lidar com o caso especial dentro da exceção.
- Não retornar nulo
 - None é uma fonte comum de erros
 - É ruim ter vários checks para None e é fácil esquecer um.
 - Em vez de retornar nulo é melhor virar uma exception ou retornar um Special Case Object.
- Não passar nulo
 - Passar None como argumento é pior ainda
 - Existem várias formas de lidar com argumentos nulos e nenhuma é boa.
 - Melhor é prevenir para nunca ter uma situação em que None é passado.