

	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ Curso: ADS Disciplina: Programação Orientada a Objetos Professor: Ely
--	---

Exercício 08 – Pesquisa exploratória

- 1) Identifique pelo menos 3 “erros” que acontecem quando você está programando com TypeScript que levam à parada brusca da execução de código.

Os 3 erros mais comuns que enfrentei foi na declaração de parâmetros nas funções, outro erro também é a função não ser executada corretamente, por fim o erro no input ao pedir para o usuário colocar as entradas.

- 2) O que é uma exceção em programação e como ela se diferencia de um erro comum?

A principal diferença entre uma exceção e um erro comum é a forma como o programa às trata, exceção é uma situação excepcional que pode ser capturada e tratada pelo código, erro comum refere-se a bugs ou falhas no código que podem ser detectados durante a compilação ou execução do programa, mas que não são necessariamente considerados excepcionais.

- 3) Como o TypeScript implementa o tratamento de exceções?

O TypeScript, sendo uma linguagem que adiciona tipagem estática ao JavaScript, segue a mesma estrutura de tratamento de exceções que o JavaScript. Isso significa que o TypeScript se baseia na capacidade de lidar com exceções usando blocos try, catch e finally.

4) Qual é a função do bloco try-catch no tratamento de exceções em TypeScript?

O bloco try-catch no TypeScript (e em JavaScript) permite que você envolva um bloco de código potencialmente problemático dentro do bloco try e capture qualquer exceção que possa ser lançada, permitindo que você a trate de maneira controlada. No bloco try é colocado o trecho de código que pode gerar uma exceção, o bloco catch é onde o código é tratado.

5) Como você pode criar exceções personalizadas em TypeScript?

É possível criar exceções personalizadas estendendo a classe Error ou qualquer uma de suas subclasses para definir seu próprio tipo de exceção.

6) Quais são os tipos comuns de exceções que podem ser encontradas em aplicações TypeScript?

TypeError: Ao tentar usar algo indefinido, nulo ou tipos incompatíveis. ReferenceError: acesso a uma variável não declarada ou fora do escopo. SyntaxError: erros de sintaxe, como palavras-chave incorretas ou parênteses ausentes.

7) Qual é o papel do bloco finally no tratamento de exceções em TypeScript?

O bloco finally em tratamento de exceções, tanto em TypeScript quanto em JavaScript, é usado para definir um conjunto de instruções que são sempre executadas, independentemente de ocorrer ou não uma exceção dentro do bloco try.

8) Como o tratamento de exceções em TypeScript ajuda a melhorar a robustez e a segurança de uma aplicação?

O tratamento de exceções é importante para melhorar o código, principalmente, prevenir falhas catastróficas, manter resiliência nos códigos, feedback controlado para seus usuários, manutenção do fluxo de execução.

9) Existem diferenças significativas no tratamento de exceções entre TypeScript e JavaScript?

O TypeScript oferece tipagem estática, ajudando a prevenir erros de tipos que poderiam causar exceções. Essa verificação em tempo de compilação reduz riscos de exceções por tipos incompatíveis. Além disso, suporta funcionalidades modernas do JavaScript, como async/await para lidar com exceções assíncronas. As ferramentas de desenvolvimento, como o Visual Studio Code, melhoram a detecção e correção de erros, facilitando o tratamento de exceções de forma mais eficiente.

10) Como você pode testar e depurar eficientemente exceções em TypeScript?

Para testar exceções em TypeScript, escreva testes específicos para casos que podem gerar erros. Use blocos try-catch para simular situações excepcionais e depure o código com ferramentas como VS Code para examinar o estado do programa. Registre informações relevantes com console.log para entender o comportamento durante exceções. Considere também ferramentas externas de rastreamento de erros, como Sentry, para capturar e analisar problemas em produção.

Links para consulta:

- https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Control_flow_and_error_handling
- <https://basarat.gitbook.io/typescript/type-system/exceptions>
- <https://www.typescriptlang.org/docs/handbook/basic-types.html>
- <https://stackoverflow.com/questions/31626231/custom-error-class-in-typescript> •
- <https://javascript.info/try-catch>
- <https://www.geeksforgeeks.org/difference-between-typescript-and-javascript/> •
- <https://testingjavascript.com/>