

Exercícios

1. Criar um programa que tente abrir um arquivo de texto e exibir seu conteúdo.
 - Capturar `FileNotFoundException` caso o arquivo não exista.
 - Capturar `UnauthorizedAccessException` caso o programa não tenha permissão para acessar o arquivo.
 - Exibir mensagens informativas para cada erro.
 - Utilizar um bloco `finally` para exibir uma mensagem de encerramento.
 - Observação: entrar com o caminho para o arquivo e utilizar `File.ReadAllText(caminhoArquivo)` para obter o conteúdo do arquivo.
2. Crie um programa que solicite ao usuário que insira um número inteiro. Caso o usuário insira um valor inválido (não numérico), capture a exceção e peça para ele inserir novamente até que forneça um valor válido.
3. Crie um programa que peça ao usuário para inserir um número e uma posição em um vetor de 10 posições. Caso o usuário tente acessar uma posição inválida (fora do intervalo 0 a 9), capture a exceção e exiba uma mensagem de erro.

Utilize `try-catch` para capturar `IndexOutOfRangeException`.

4. Crie um programa que solicite ao usuário que insira uma data no formato "dd/MM/yyyy". Caso o usuário insira uma data inválida, o programa deve capturar a exceção e continuar pedindo até que uma data válida seja inserida.

- Utilize `DateTime.TryParseExact()` para validar a data.
- Capture `FormatException` caso o formato esteja errado.
- Permita que o usuário tente novamente até fornecer uma entrada correta.

5. Crie um sistema para uma escola de idiomas que gerencie professores, alunos e os cursos que os alunos podem se matricular. O sistema deve incluir:

1. Uma classe `Aluno`, que contém:
 - a. Nome
 - b. Idade
2. Uma classe `Professor`, que contém:
 - a. Nome
 - b. Especialização
3. Uma classe `Curso`, que contém:
 - a. Nome do curso
 - b. Duração (em meses)
 - c. Professor responsável
4. Uma classe **associativa** `Matricula`, que vincula os alunos aos cursos. Essa classe deve conter:
 - a. Aluno
 - b. Curso
 - c. Data de matrícula

6. O programa deve instanciar alguns **alunos**, **professores** e **cursos** e demonstrar a criação de matrículas.

Você deve criar um sistema para um **hospital**, onde existam **pacientes**, **médicos** e **consultas**. O sistema deve conter:

1. **Uma classe base Pessoa** contendo:
 - a. Nome
 - b. Idade
2. **Duas classes derivadas de Pessoa:**
 - a. **Medico**, com especialidade e CRM.
 - b. **Paciente**, com um histórico de doenças.
3. **Uma classe Consulta (classe associativa), que liga médicos e pacientes.**
 - a. Médico responsável.
 - b. Paciente atendido.
 - c. Data da consulta.
 - d. Diagnóstico.

O programa deve criar objetos para representar **médicos**, **pacientes** e **consultas**, exibindo os detalhes no final.

7. Você deve criar um sistema para gerenciar **bibliotecas**, que envolva **livros** e **empréstimos**. A ideia é que um **usuário** possa **emprestar** e **devolver** livros, com validação para:

- **Tentativa de empréstimo de livro não disponível.**
- **Tentativa de devolução de livro que não foi emprestado.**
- **Tentativa de empréstimo de livro por usuário que tem livro emprestado com data ultrapassada.**

1. Classe Livro

- a. Título
- b. Autor
- c. ID (único para cada livro)

d. Status (Disponível / Emprestado)

2. Classe Usuario

- a. Nome
- b. ID
- c. Lista de livros emprestados

3. Classe Emprestimo (associativa)

- a. Livro emprestado
- b. Usuario que fez o empréstimo
- c. Data do empréstimo
- d. Data prevista para devolução

4. Tratamento de exceções:

- a. `ArgumentException` para IDs inválidos.
- b. `InvalidOperationException` para tentar emprestar ou devolver um livro não disponível ou que não foi emprestado.

8. Crie um programa que simula o download de um arquivo usando `async/await`. O programa deve exibir mensagens indicando o início e o término do download e, enquanto isso, continuar executando outras tarefas.

- Criar um método assíncrono `BaixarArquivoAsync()` que simula um download usando `Task.Delay()`.
- Enquanto o download ocorre, o programa deve executar outra tarefa.
- Após o término do download, exibir uma mensagem confirmando a conclusão.