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.

1. 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.
2. 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:
   1. Nome
   2. Idade
2. Uma classe Professor, que contém:
   1. Nome
   2. Especialização
3. Uma classe Curso, que contém:
   1. Nome do curso
   2. Duração (em meses)
   3. Professor responsável
4. Uma classe **associativa** Matricula, que vincula os alunos aos cursos. Essa classe deve conter:
   1. Aluno
   2. Curso
   3. 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:
   1. Nome
   2. Idade
2. **Duas classes derivadas de Pessoa:**
   1. Medico, com especialidade e CRM.
   2. Paciente, com um histórico de doenças.
3. **Uma classe Consulta (classe associativa), que liga médicos e pacientes.**
   1. Médico responsável.
   2. Paciente atendido.
   3. Data da consulta.
   4. 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**
   1. Título
   2. Autor
   3. ID (único para cada livro)
   4. Status (Disponível / Emprestado)
2. **Classe Usuario**
   1. Nome
   2. ID
   3. Lista de livros emprestados
3. **Classe Emprestimo (associativa)**
   1. Livro emprestado
   2. Usuario que fez o empréstimo
   3. Data do empréstimo
   4. Data prevista para devolução
4. **Tratamento de exceções:**
   1. ArgumentException para IDs inválidos.
   2. 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.