



Exercícios

Exercício 1:

- Criar um `ArrayList` de `Strings` e inicializá-lo com uma nova lista vazia.
- Adicionar os seguintes valores: "Pato", "Cachorro", "Gato".
- Imprimir o conteúdo da posição 3.
- Executar o programa. O que acontece?
- Criar uma cláusula `try-catch` para controlar a exceção anterior. O tipo de exceção e onde surgiu deverão ser impressos na tela.

Exercício 2

- Crie uma Classe **CalculoMatematico**
 - Nela, **crie um método divisão, que recebe como parâmetros os valores a serem divididos**. O retorno é o resultado da divisão (todos os números devem ser do tipo inteiro);
- Crie uma **classe de teste** para testar a `CalculoMatematico`
 - **Nela crie um objeto `CalculoMatematico`** e acesse o método `divisao`, tentando dividir 4 por 0.
- Execute a classe e veja o que acontece;
- Crie um bloco **`try...catch`** no método `divisão` para tratar a operação realizada;
- No `catch`:
 - Informar o objeto do tipo **`ArithmeticException`**





Exercícios

- o Imprimir uma mensagem informando que a operação não pode ser realizada
 - o Retorna zero
- Tire o bloco **try...catch** do método divisão;
- Adicione **throw ArithmeticException** na assinatura do método;
- E na primeira linha do bloco do método, faça uma verificação se o divisor é igual a 0:
 - o Se for, lance uma exceção **throw new ArithmeticException("Texto");**
- Na classe de teste, crie um bloco **try...catch**, tentando executar o método divisão;
 - o Catch **para ArithmeticException**
 - o No bloco do Catch, **imprima o atributo message** do objeto criado do tipo **ArithmeticException**