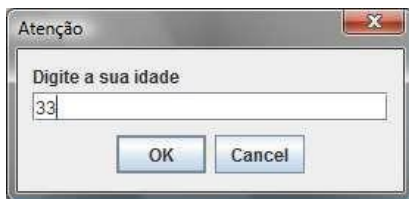


1. Desenvolva um **programa** que solicita ao usuário a digitação de uma frase, depois verifica e exibe as vogais não existentes nesta frase.
2. Desenvolva um **programa** gerador de nomes. O programa deve gerar e apresentar 5 nomes aleatórios utilizando uma sequência de consoante + vogal com tamanho de 4, 6 ou 8 caracteres.

Ex. de possíveis nome gerados: fabi, Julita, tabata, mimosa, batira.

Math.random() retorna um valor aleatório do tipo real entre 0 e 1. Se você quiser gerar um número aleatório entre 1 e 25, faça o seguinte comando: `int n=(int)(Math.random()*25)+1`

3. Faça o **método estático** `LerInt()` em Java que recebe como parâmetro um título e uma mensagem. O método deve abrir uma caixa de entrada no estilo `JOptionPane.showInputDialog(...)` com o título e a mensagem recebidos como parâmetro e retornar na forma de um inteiro, o valor digitado pelo usuário. Caso o usuário digite um valor inválido a função deve retornar o valor zero. Faça também versões dessas funções para a leitura de valores reais.



Ex: `int idade=LerInt("Atenção","Digite a sua idade");`

4. Faça um **programa** que, a partir de um texto digitado pelo usuário, conte o número de caracteres total e o número de palavras (palavra é definida por qualquer sequência de caracteres delimitada por espaços em branco, vírgula ou ponto) e exiba o resultado. Dica: use o método `split` com expressões regulares.

<https://www.devmedia.com.br/conceitos-basicos-sobre-expressoes-regulares-em-java/27539>

5. Desenvolva um **método estático** com a finalidade de testar se uma string recebida como parâmetro pode ser considerada uma placa de carro (AAA9999 ou AAA9A99). Retorne `true` ou `false`. Sugestão: utilize o método **matches** com expressões regulares das Strings.
6. Escreva um **método estático** que ao receber uma string como parâmetro seja capaz de retornar a abreviação da string, ou seja, o resultado da concatenação dos primeiros caracteres das palavras separadas por espaço em branco e que comecem com letra maiúscula.

Exemplo: "Faculdade de Informática de Presidente Prudente" deve resultar em "FIPP"

7. [DESAFIO] Desenvolva um **método estático** que ao receber uma frase como parâmetro e quantidade máxima de colunas, seja capaz de retornar a mesma frase acomodada e justificada em uma área limitada ao número de colunas informada. As palavras não poderão ser divididas.

Exemplo:

O usuário digitou: **O mundo é um lugar perigoso de se viver, não por causa daqueles que fazem o mal, mas sim por causa daqueles que observam e deixam o mal acontecer. Albert Einstein**

Se a quantidade máxima de colunas for o valor 30, A frase deverá ser apresentada assim:

1.....30

**O mundo é um lugar perigoso de
se viver, não por causa
daqueles que fazem o mal, mas
sim por causa daqueles que
observam e deixam o mal
acontecer. Albert Einstein**

Sugestão: Transfira sequencialmente todas as palavras para a String sfraseresult com o cuidado de não dividir uma palavra e adicione o "\n" no final de cada linha, evitando ainda que cada linha não exceda 30 colunas.