



1. Escreva um programa que pergunte vários números ao usuário. O usuário inserirá os números um após o outro. Quando o usuário não quiser inserir mais números, ele digitará 0 (zero). Após isso, o programa deverá calcular a média dos números digitados. Há a possibilidade do usuário não digitar nenhum número, ou seja, digitar um zero logo de cara.
2. Escreva um programa que pergunte dois números inteiros ao usuário. Em seguida, o programa deverá calcular a soma de todos os números entre os dois digitados pelo usuário (incluindo os números digitados). O programa também deve ser capaz de ir de trás pra frente (exemplo: somar de 9 até 3).
3. Escreva um programa para ajudar a responder a equação $x * y = z$. O programa funcionará da seguinte forma: pergunte ao usuário os valores de x , y e z . O valor que for 0, será o valor que o programa deverá calcular. Ex.: $x = 4$, $y = 0$, $z = 8$. Neste exemplo, o programa deverá calcular o valor de y de forma que a equação fique correta (no caso, $y = 2$). Se o usuário digitar mais de um 0, o programa deverá informar um erro (exemplo de mensagem de erro: “Você só pode digitar um zero”). O programa deverá perguntar os números até o usuário digitar 0 para as três variáveis.
4. Escreva um programa que calcule a soma de todos os números primos entre um intervalo especificado pelo usuário, incluindo os números que o usuário digitou (caso eles também sejam primos).
5. Crie um programa que calcule quantos quadrados pequenos cabem em um retângulo grande. Para isto, o programa deverá receber o tamanho do lado dos quadrados pequenos e a altura e largura do retângulo grande. Em seguida, deverá informar quantos destes quadrados pequenos cabem no retângulo.
6. Escreva um programa que desenhe um quadrado na tela com o tamanho que o usuário informar. Abaixo, exemplo de um quadrado de tamanho 4 (OBS.: Não existem quadrados de tamanho negativo ou nulo).

```
****
*  *
*  *
****
```

Referência para a classe JOptionPane

Para importar:

```
import javax.swing.JOptionPane;
```

Para solicitar dados ao usuário:

```
String nome = JOptionPane.showInputDialog("Qual seu nome?");
```



Lembre-se que o `showInputDialog()` retorna um objeto da classe `String`. Para solicitar um inteiro, devemos fazer um parse na string da seguinte forma:

```
int idade = Integer.parseInt(JOptionPane.showInputDialog("Qual sua idade?"));
```

Para exibir uma mensagem:

```
JOptionPane.showMessageDialog(null, "Olá, mundo!");
```

Para exibir um vetor:

```
// importamos a classe que é usada pra converter um vetor em String  
import java.util.Arrays;
```

Na hora de exibir:

```
JOptionPane.showMessageDialog(null, Arrays.toString(vetor));
```

Outros parâmetros para o `showMessageDialog()` e `showConfirmDialog()`:

- Tipos de mensagens: `ERROR_MESSAGE`, `INFORMATION_MESSAGE`, `WARNING_MESSAGE`, `QUESTION_MESSAGE`, `PLAIN_MESSAGE`.
- Opções da caixa: `DEFAULT_OPTION`, `YES_NO_OPTION`, `YES_NO_CANCEL_OPTION`, `OK_CANCEL_OPTION`.

Para usar estes parâmetros, devemos acrescentá-los ao método. Exemplos:

```
JOptionPane.showConfirmDialog(null, "Deseja sair?", "Saída",  
JOptionPane.YES_NO_OPTION);
```

Isso exibirá uma caixa com a mensagem "Deseja sair?". O título da caixa será "Saída" e mostrará apenas os botões "Sim" e "Não". Nesse caso, o método poderá retornar os valores `YES_OPTION` ou `NO_OPTION`. Outros valores que podem ser retornados usando os parâmetros acima são: `CANCEL_OPTION`, `OK_OPTION` e `CLOSED_OPTION`.

Os parâmetros de tipos de mensagens podem ser usados no `showMessageDialog()` da mesma forma, sempre como quarto parâmetro, após a janela pai (que estamos usando sempre `null`), o texto e o título da caixa. Exemplo:

```
JOptionPane.showMessageDialog(null, "Entrou água!", "Erro",  
JOptionPane.ERROR_MESSAGE);
```