


Universidade Federal de Viçosa – Departamento de Informática
INF311 – Programação para dispositivos móveis
Prof. Lucas Francisco da Matta Vegi

- Atividade prática 01 -

O objetivo desta atividade prática é desenvolver dois aplicativos que se comportem como uma calculadora. Para isso, essa atividade será dividida em dois níveis de complexidade.

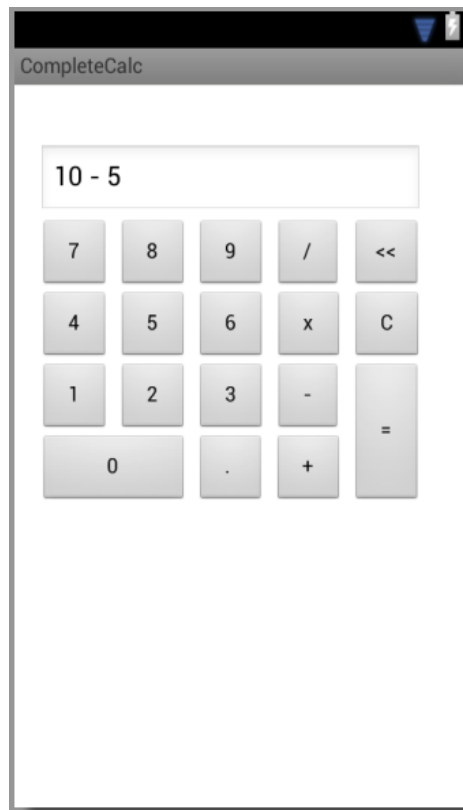
a) Desenvolva uma calculadora simples que siga o seguinte padrão de tela:

The image shows a mobile application interface for a simple calculator. At the top, there is a title bar with the text "SimpleCalc". Below the title bar, the interface is divided into several sections. First, there is a label "Valor 1:" followed by a text input field. Below this, there is a label "Valor 2:" followed by another text input field. Underneath the input fields, there are four buttons arranged horizontally, each containing a mathematical operator: a plus sign (+), a minus sign (-), a multiplication sign (*), and a division sign (/). At the bottom of the screen, there is a label "O Resultado é:" followed by a large, empty text area for displaying the result of the calculation.

Este aplicativo espera que o usuário digite dois valores numéricos e escolha uma das quatro operações matemáticas básicas representadas por botões. Quando o usuário pressionar um desses botões, a respectiva operação matemática será executada e o resultado dela será exibido em uma `TextView` na parte inferior da tela.

Observação: Garanta que o seu app evita que operações inválidas como divisão por 0 (zero) ocorram e provoquem erros de execução.

- b) Desenvolva uma calculadora completa que siga o seguinte padrão de tela:



Este aplicativo possui um visor representado por um `EditText`. Esse visor não deverá ser editável, ou seja, todas as informações exibidas no mesmo serão oriundas do pressionamento dos botões presentes no aplicativo. Para isso, estude os seguintes atributos:

- `android:inputType="none"`
- `android:editable="false"`

Como se pode observar, esse aplicativo terá um comportamento muito similar ao das calculadoras tradicionais. Sendo assim, para que a informação “10 - 5” exibida no exemplo seja apresentada no visor, o usuário deverá pressionar os botões “1”, “0”, “-” e “5” exatamente nessa ordem. Neste momento, se o usuário pressionar o botão “=”, o resultado da operação será exibido no visor, podendo ser reutilizado em novas operações matemáticas.

Em outras palavras, se, imediatamente após a apresentação de um resultado no visor, o usuário clicar em algum operador matemático, o primeiro operando dessa operação será o valor do resultado que já estava presente no visor. Caso contrário, se, imediatamente após a apresentação de um resultado no visor, o usuário começar a digitar um número, a memória da calculadora deverá ser limpa, e o novo número digitado substituirá o valor que era exibido anteriormente.

Observe que a calculadora deverá impedir que operações inválidas, como a divisão por 0 (zero), disparem uma exceção em tempo de execução. Para isso, ela deverá verificar os operandos e operadores envolvidos na

expressão e informar “ERROR” no visor caso o usuário tente realizar uma operação inválida.

Por fim, para simplificar a nossa solução, é esperado que as expressões a serem resolvidas pela calculadora envolvam apenas um tipo de operador matemático por vez.

O botão “<<” representa a operação de *backspace*, excluindo o último dígito informado quando pressionado. O botão “C” deve limpar o visor ao ser pressionado. Por fim, o botão “.” deve ser utilizado para inserir valores decimais.

Ainda com relação ao botão “.”, a calculadora deverá garantir que o separador decimal só seja exibido no visor se, imediatamente antes dele, tiver sido digitado um dígito numérico e se esse dígito fizer parte de um número que ainda não contenha nenhum “.” em sua composição.

Observação:

Para a realização desta atividade, pesquise a respeito dos métodos *split()*, *substring()*, *length()*, *indexOf()*, todos chamados a partir de objetos do tipo *String*. Estude também o método *Double.parseDouble()*, usado na conversão de uma *String* para um valor numérico.

ENTREGA DA ATIVIDADE:

Ao final da atividade, suba os seus dois projetos para um ou dois repositórios públicos do GitHub e informe a URL desses diretórios em um arquivo enviado pelo PVAnet Moodle.