

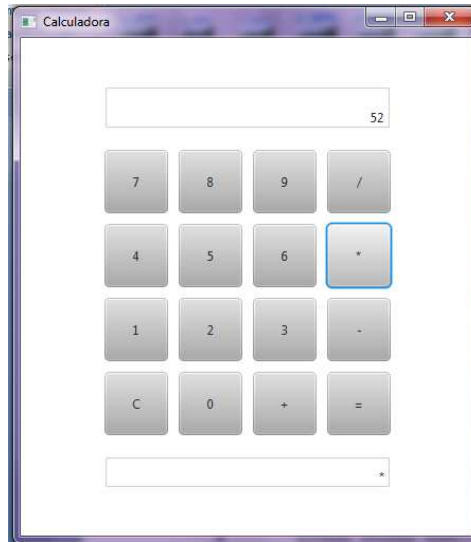
Trabalho de Laboratório – Curso EI

Objectivos:

- Introdução ao JavaFX 2.0 - Painéis e controlos de interface simples

Programas:

- Desenvolvimento de uma calculadora que permite efetuar operações com números inteiros.



Regras de implementação:

- Criar um projeto JavaFX Application.
- Implementar o código necessário e testar no fim de cada nível.
- Use as convenções de codificação adoptadas para a linguagem Java.

Implementação:

Nível 1:

- Crie um novo projeto JavaFx Application.
- Comece por implementar o teclado criando para isso a classe **TeclasPad**. Esta classe deve ter como atributos um painel do tipo **GridPane** e uma lista de botões (os botões são da classe **Button**). No construtor crie e inicialize a lista com os 16 botões do teclado mostrados na figura. Crie ainda a **GridPane** e preenche-a com os botões criados, um por cada célula.
- Acrescente o método "**Pane getTeclasPane()**" que devolve o atributo **GridPane** criado.
- Para testar a classe **TeclasPad**: no método **start** da classe da aplicação comece por alterar o nome e as dimensões da janela de acordo com a figura deste enunciado. De seguida, remova ou comente o código relativo à criação do botão (btn) e altere o tipo do objecto **root** para **FlowPane**. Crie um objecto da classe **TeclasPad** e em vez de adicionar o nó **btn** ao objecto **root** adicione o painel do keypad obtido através do método **getTeclasPane**. Deverá visualizar o painel de botões criado.

Trabalho de Laboratório – Curso EI

Nível 2:

- Para representar a calculadora mostrada na figura crie agora a classe **Calculadora**. Esta classe deve ter como atributos um objecto da classe **TeclasPad**, um objecto da classe **TextField** que irá ser o ecrã da calculadora, um objecto do tipo **TextField** que irá guardar o último operador inserido e um painel do tipo **VBox**. O painel **VBox** irá servir para colocar na primeira célula o campo de texto do ecrã (**TextField**), na segunda o painel de botões do **TeclasPad**, e na terceira célula o campo de texto do operador (**TextField**). No construtor crie os objectos dos atributos e construa o painel **VBox** da consola de acesso colocando os elementos referidos nas respectivas células.
- Acrescente o método “**Pane getCalculadoraPane()**” que retorna o painel **VBox** guardado no atributo.
- Para testar: no método **start** adapte o código para usar um objecto da classe **Calculadora** em vez do objecto da classe **TeclasPad**.

Nível 3:

- Pretende-se agora implementar o funcionamento dos botões da classe **TeclasPad**. Para este fim deve incluir no construtor a captura do evento **ActionEvent** associado a cada um dos botões do keypad. Este procedimento é semelhante ao que existia no código de teste criado pelo Netbeans no método **start**.
- Para testar, no método **handle** de resposta ao evento capturado anteriormente, deve colocar a seguinte linha de código: **System.out.println("Botão premido: " + nomeBotao);** **nomeBotao** neste caso corresponde ao texto do botão que foi premido.

Nível 4:

- Para concluir a utilização dos botões torna-se necessário que a classe **Calculadora** seja informada que um dos botões do seu **TeclasPane** foi premido. Para que isto aconteça crie dentro da classe **TeclasPane** um atributo para guardar uma referência da classe **Calculadora** e altere o construtor para que passe a receber essa referência como argumento. Na classe **Calculadora** terá de alterar também a criação do objecto **TeclasPane** passando no construtor o próprio objecto da classe **Calculadora** (**this**). Desta forma passa a ser possível aceder ao objecto à classe **Calculadora** a partir do objecto da classe **TeclasPane** incluído como atributo. Note-se que esta solução não é a mais aconselhável sendo utilizada aqui apenas por uma questão de simplicidade.
- Crie agora na classe **Calculadora** o método **void keyPressed (String key)** que irá receber a informação da tecla que foi premida no keypad. Para isso substitua a linha de código associada ao método **handle** criado no nível anterior dentro da classe **TeclasPane** por uma chamada a este método qualquer que seja o botão.
- Dentro do método **KeyPressed** criado deve ir guardando o carácter que vem na String **key** acrescentando-o a uma String **valorEcra** que deve criar como atributo da classe. Apenas deve guardar nesta String algarismos até um máximo de nove. Os seguintes deverão ser ignorados. Sempre que o botão “C” for pressionado o ecrã deve de ser limpo.

Trabalho de Laboratório – Curso EI

Nível 5:

- Nos níveis anteriores criou a base de funcionamento de uma calculadora. Para que a calculadora execute perfeitamente as operações aritméticas elementares (+, -, *, /) deverá fazer o seguinte:
 1. Criar quatro atributos na classe **Calculadora**: uma String para guardar o primeiro operador, uma String para guardar o segundo operador, uma String para guardar o último operador inserido, e uma String para guardar a última tecla inserida.
 2. Na classe **Calculadora** defina o método **isOperador** que recebe um string como parâmetro, devolve true se a String for um operador e false em caso contrário.
 3. Altere o método **keyPressed** na classe **Calculadora** para o código abaixo definido, usando a classe **Calculos** disponibilizada em conjunto com o enunciado. A classe **Calculos** permite efetuar cálculos com números inteiros através do método **calcular**.

De notar que sempre que um operador (+, -, *, /) é pressionado, ele aparece na caixa de texto que está por baixo das teclas.

Trabalho de Laboratório – Curso EI

```
public void keyPressed(String key) {
    switch (key) {
        case "+":
        case "-":
        case "*":
        case "/":
            if (operando1.isEmpty()) {
                operando1 = ecra.getText();
            } else {
                operando2 = ecra.getText();
                ecra.setText(Calculos.calcular(operando1, operando2, ultimoOperador));
                valorEcra = ecra.getText();
                operando1 = valorEcra;
                operando2 = "";
            }
            ultimoOperador = key;
            operador.setText(key);
            break;
        case "=":
            if (!operando1.isEmpty()) {
                operando2 = ecra.getText();
                ecra.setText(Calculos.calcular(operando1, operando2, ultimoOperador));
                valorEcra = ecra.getText();
                operando1 = "";
                operando2 = "";
            }
            ultimoOperador = "";
            operador.setText("");
            break;
        case "C":
            valorEcra = "";
            ecra.setText(valorEcra);
            operador.setText("");
            ultimoOperador = "";
            operando1 = "";
            operando2 = "";
            break;
        default:
            if (isOperador(ultimaTecla)) {
                valorEcra = "";
            }
            if (valorEcra.length() < 9) {
                valorEcra += key;
            }
            ecra.setText(valorEcra);
    }
    ultimaTecla = key;
}
```

Para os identificadores siga as convenções adoptadas normalmente, em particular:

- 1) A notação camelCase para o nome das variáveis locais e identificadores de atributos e métodos.
- 2) A notação PascalCase para os nomes das classes.
- 3) Não utilize o símbolo '_', nem abreviaturas nos identificadores.