UAST
Unidade Acadêmica de Serra Talhada - PE
UFRPE
Desde 2006

BSi
Bacharelado em Sistemas de Informação

MPOO

Site: https://sites.google.com/site/profricodemery/mpoo

http://ava.ufrpe.br/

https://sigs.ufrpe.br/sigaa/ava/index.jsf

Disciplina: Modelagem e Programação Orientada a Objetos (MPOO)

Profº: Richarlyson D'Emery

LISTA DE EXERCÍCIOS VIII

Leia atentamente as instruções gerais:

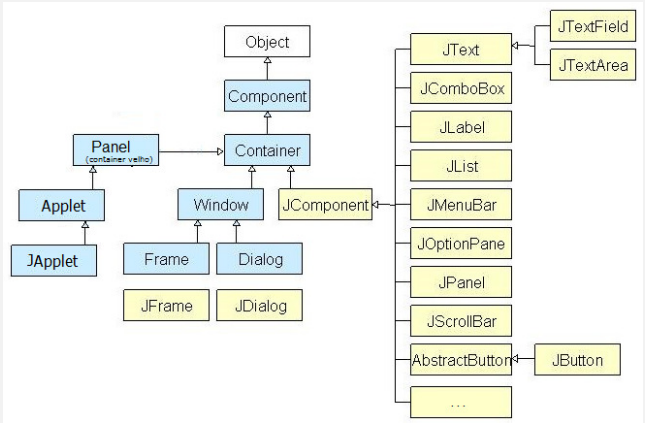
- No Eclipse crie um novo **projeto chamado br.edu.mpoo.listaVIII.SeuNomeSobrenome**, o qual deverá ter **pastas de pacotes** para cada questão: questao1, questao2, e assim sucessivamente, contendo todas as respostas da lista.
- A lista envolve questões práticas.

Fique atento!

Prezado aluno, esta é a lista de exercícios é relativa ao assunto de "Componentes Gráficos".

Saiba Mais!

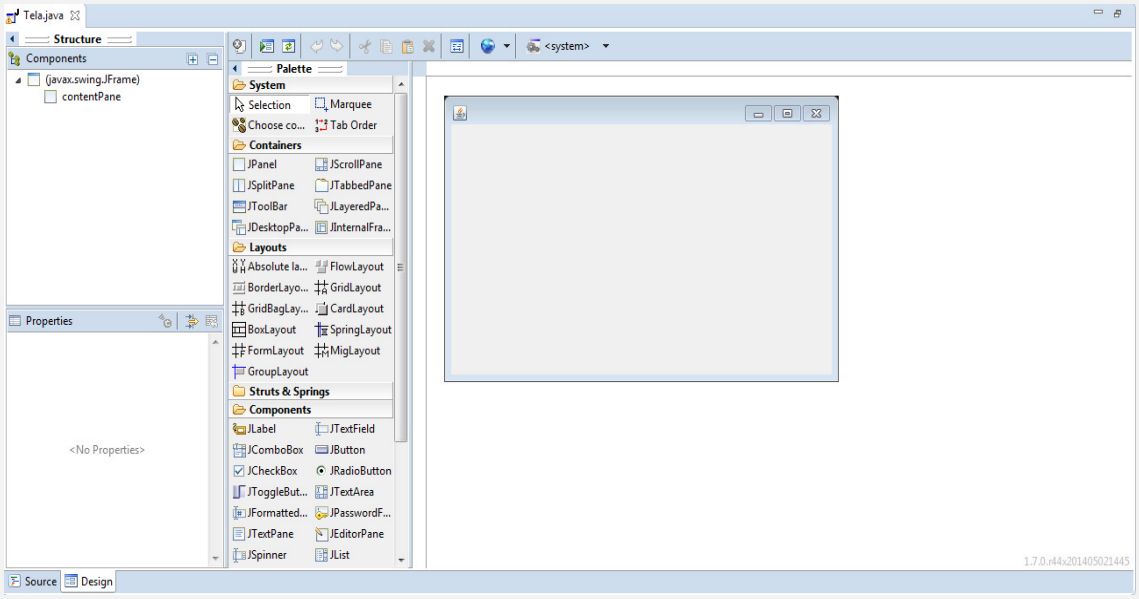
Os elementos básicos necessários para criar uma interface gráfica (GUI, do inglês *graphical user interface*) para um usuário utilizar um sistema estão em dois pacotes java.awt e javax.swing, em que awt foi o primeiro conjunto de classes Java para construir GUI's, enquanto swing é uma extensão de awt que mantem e amplia os conceitos de awt, em especial para tratar as aparências multiplataformas.



Uma GUI é baseda em dois elementos: containers (janelas e painéis) e componentes (menus, botões, caixas de texto e seleção, barras de rolagem, rótulos, tabelas, etc.).

Investigue e pratique a implementação de diversos componentes: https://docs.oracle.com/javase/8/docs/api/javax/swing/JComponent.html

Visando a produtividade de desenvolvimento de software, diversos IDE's dispõem de editores visuais baseados em drag-and-drop (arrastar e soltar) para construção de GUI's.



Você Sabia?

A aparência e a forma em que o usuário interage com a aplicação são chamados de look and feel da aplicação.

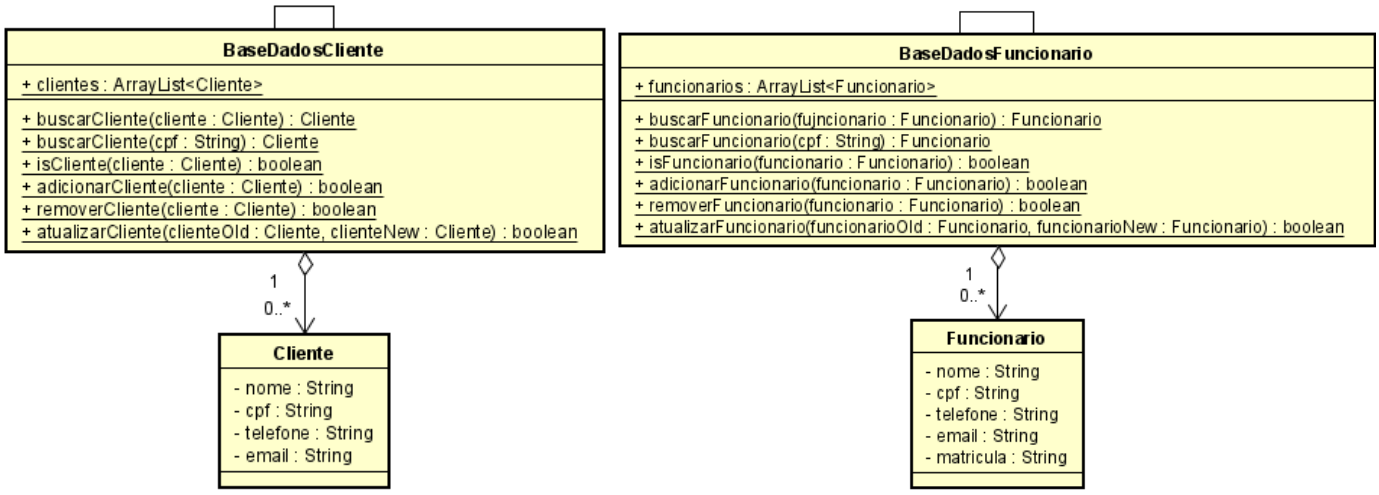
Desde a atualização 10 do Java SE 6, as GUI's passaram a ter uma *cara* nova, elegante e compatível com várias plataformas, conhecida como Nimbus.

```
try {
        UIManager.setLookAndFeel("javax.swing.plaf.nimbus.NimbusLookAndFeel");
} catch (ClassNotFoundException | InstantiationException
          | IllegalAccessException | UnsupportedLookAndFeelException e) {
        e.printStackTrace();
}
```

Mão na Massa!

As telas descritas nas questões abaixo não possuirão funcionalidades, ou seja, se o usuário clicar em um botão ou menu, não haverá interação.

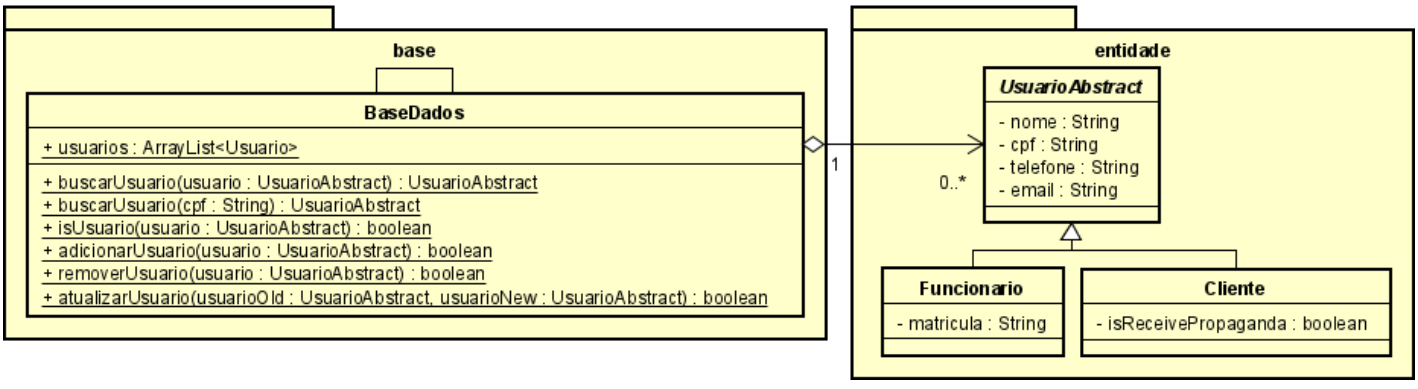
1) Vimos (em outra lista de exercícios) que o programador "O Furão" (codinome para *mustela putórius furo*) precisou resolver a demanda de um contratante da empresa MPOOSoftware LTDA para a atualização de um sistema de cadastro. Inicialmente se deparou com os seguintes diagramas de classes atuais da empresa:



Após analisar os diagramas e as regras de negócios:

- RN01 – um cliente ou funcionário é identificado pelo seu cpf;
- RN02 – um cliente ou funcionário só poderá ser cadastrado uma única vez; e
- RN03 – a empresa só envia propaganda se o cliente permitir recebê-la.

Propôs uma melhoria para o sistema (diagrama abaixo) que foi aceita por seu Scrum Master, fazendo com que passasse a ter uma única base com uso de polimorfismo.

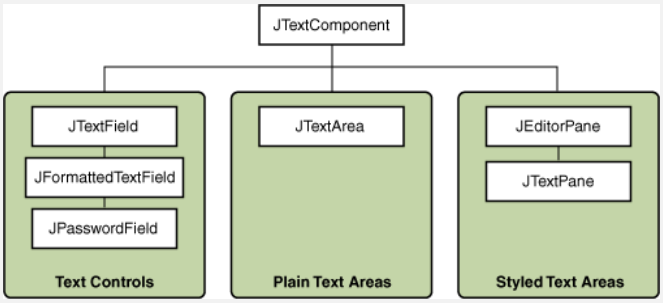


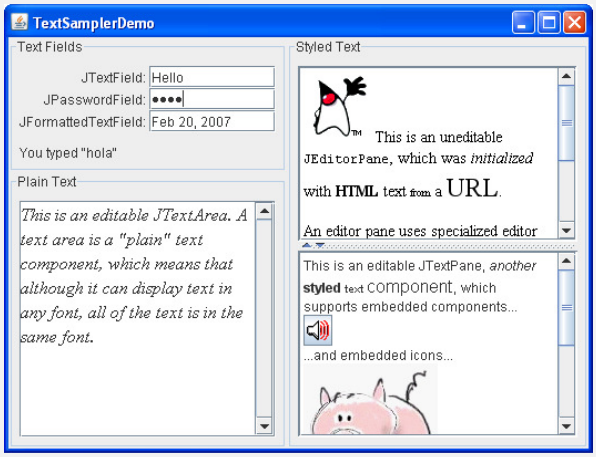
Sendo assim codifique a GUI ao lado **(Fig. 1A)** para que se possa cadastrar um usuário, sabendo que este pode ser um "Funcionario" ou "Cliente". Para o desenvolvimento, o Scrum Master definiu que a solução deve utilizar exclusivamente as bibliotecas disponíveis no JDK, ou seja, em `java` e `javax`, sem o auxílio de recursos de IDE's do tipo *drag-and-drop*, como, por exemplo, a *palette de design* do Eclipse.

Fig. 1A

Saiba Mais!

Os componentes de texto podem ser personalizados. O Swing fornece seis componentes de texto, através de classes e interfaces. Apesar de seus diferentes usos, todos herdam da mesma superclasse, JTextComponent, que fornece uma base altamente configurável e poderosa para manipulação de texto. Vide a hierarquia de JTextComponent e um exemplo de sua utilização.





Para saber mais sobre JTextComponent acesse o tutorial disponível em: https://docs.oracle.com/javase/tutorial/uiswing/components/text.html

Desafio!

2) São diversos os gerenciadores de layout que servem para determinar o tamanho e a posição dos componentes em uma GUI. Entretanto, são diversas as possibilidades, inclusive da utilização de soluções disponibilizadas em classes. Para desafiar "O Furão", seu Scrum Master o apresentou as possibilidades de configuração para JTextComponent, ao SpringLayout através de um exemplo (Fig. 1B) e de um tutorial de JavaSE da Oracle disponível em https://docs.oracle.com/javase/tutorial/uiswing/layout/spring.html. Mostre que você também é capaz de aplicar soluções de layouts, em especial, a tela de cadastro de usuário (Fig. 1A). Para isso, utilize a personalização pas os tipos de JTextComponent e a combinação de JPanels e gerenciadores de layout como BorderLayout e SpringLayout.

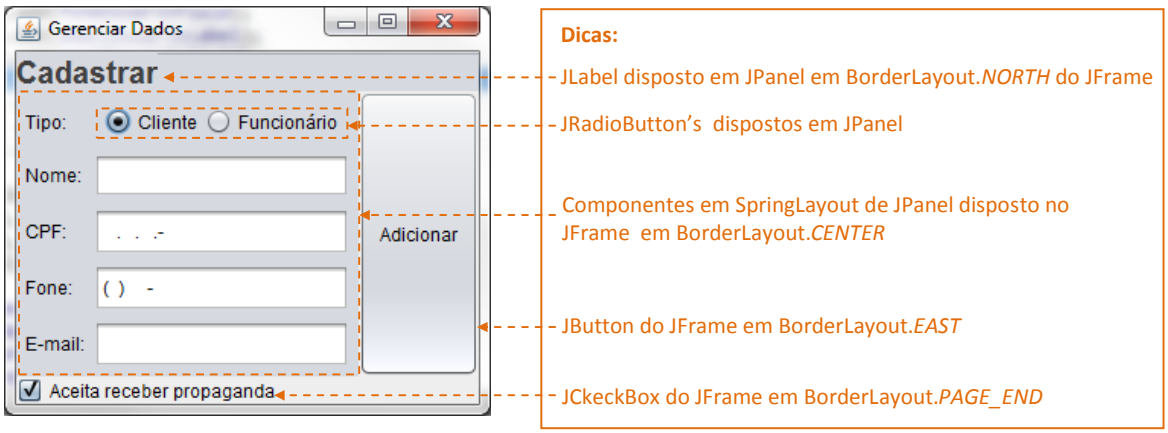


Fig. 1B

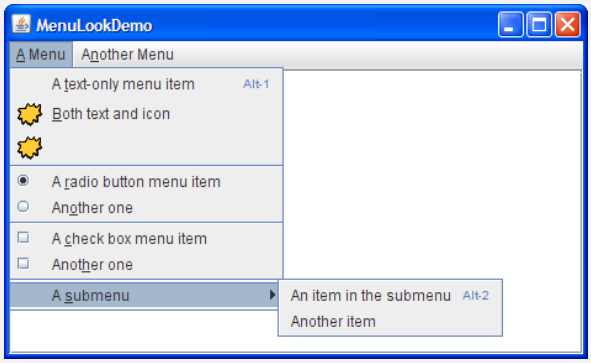
Fique Atento!

Observe a utilização do componente JRadioButton. Para que a seleção entre os tipos de usuários "Cliente" e "Funcionário" é de disjunção exclusiva (*exclusive or* - XOR). Com isso, os dados do cadastro depende se o usuário é um cliente ou um funcionário. Para que essa disjunção seja aplicada é necessário o agrupamento desses componentes em ButtonGroup.

Saiba Mais!

Em um sistema é possível adicionar no menu opções que não são comumente usuais, mas que permitem a personalização amigável de uma interface, através da inclusão de: ImageIcon, JRadioButton, JCheckBox, JSeparator e indicação de tecla de atalho. Por exemplo:

Para saber mais como utilizar esses componentes e para seus respectivos tratamentos e eventos, acesse o tutorial de JavaSE da Oracle disponível em:
https://docs.oracle.com/javase/tutorial/uiswing/components/menu.html

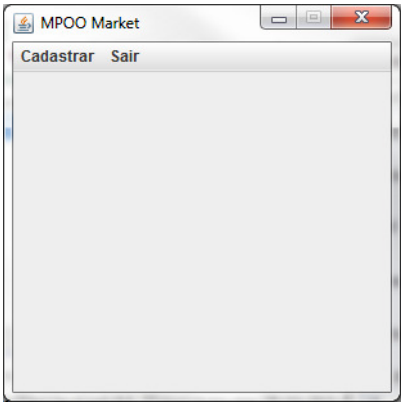


3) Uma Empresa solicitou a um de seus programadores (de codinome *mustela putórius furo* – "O Furão") que propusesse as GUI's para o sistema "MPOO Market" em que são gerenciados clientes e produtos. Para o desenvolvimento, o Scrum Master definiu que a solução deve utilizar exclusivamente as bibliotecas disponíveis no JDK, ou seja, em `java` e `javax`, sem o auxílio de recursos de IDE's do tipo *drag-and-drop*, como, por exemplo, a palette de Design do Eclipse. Para ilustrar o sistema, foram definas duas telas: uma para abertura do sistema (Fig. 3A) e outra para gerenciar dados (Fig. 3B).

(300 x 300)

Fig. 3A*

**Imagem não disponibilizada.*



(300 x 300)

Fig. 3B

Na barra de menu (JMenuBar), Fig. 3C, contém as opções para gerenciamento dos conceitos pretendidos. Mas, é possível observar que as opções estão incompletas, uma vez que se pode cadastrar, buscar, remover e atualizar os dados de um cliente ou um produto.

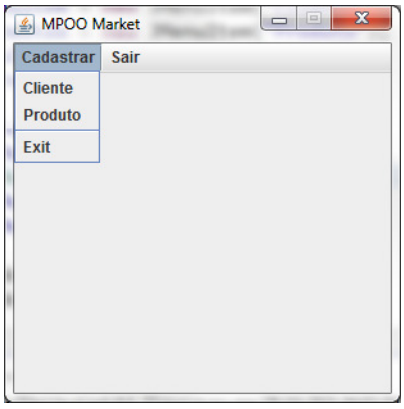


Fig. 3C

Modifique a aparência e adicione as opções de atalho para o menu, conforme a Fig. 4C.

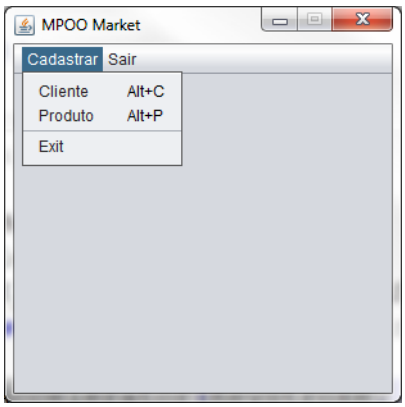
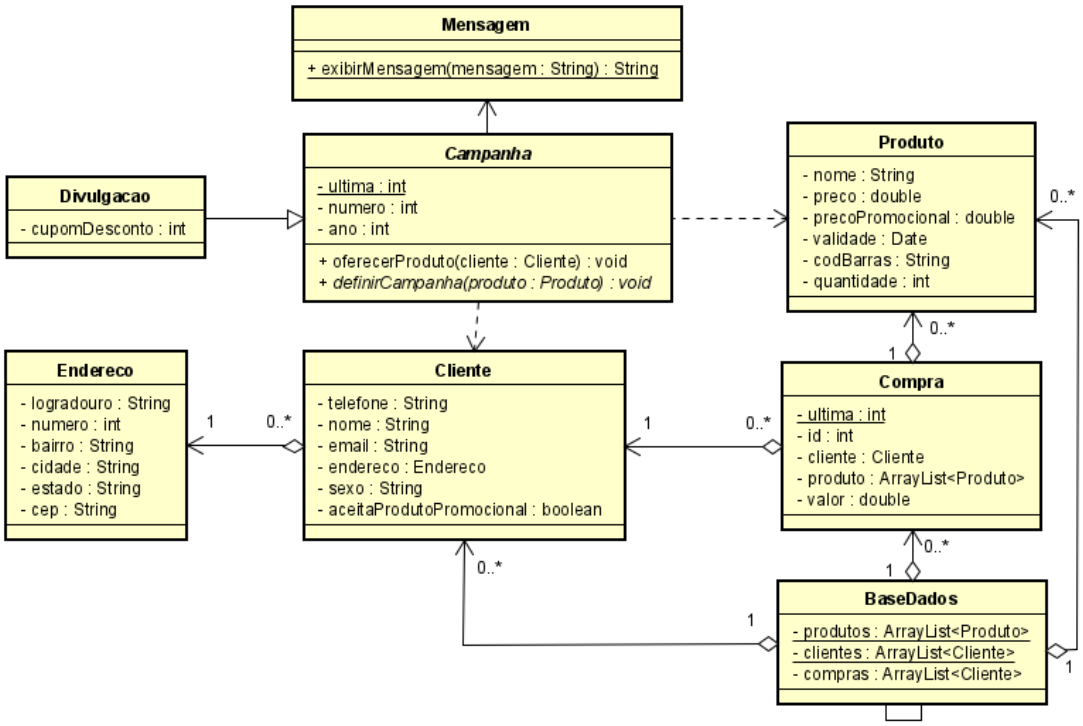


Fig. 4C

4) A partir do diagrama de classe abaixo, proponha uma GUI para gerenciar os dados de Cliente e Produto.



Desafio!

5) A GUI da calculadora ao lado (Fig. 3). Você deve utilizar componentes gráficos de javax.swing. Observe a disposição das opções da calculadora e escolha o devido layout.

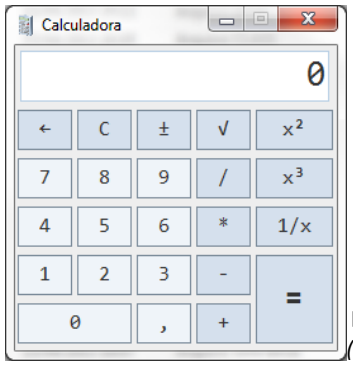


Fig. 1
(260 x 255)