



Professor: Edwar Saliba Júnior

- 1) Crie uma pequena agenda de pessoas e endereços utilizando os componentes já vistos até então e um objeto *Collection* (Exemplo: *ArrayList*) para armazenar as pessoas cadastradas na memória principal do computador. Exigências: para o campo sexo, utilize o componente `JRadioButton` e dê somente duas opções; para os campos data de nascimento e CEP utilize o componente `JTextField` formatado; para o campo Unidade da Federação (estado) deverá ser usado um `JComboBox`; a opção de gosto (gosta de música, gosta de filmes, gosta de baladas e gosta de beber) utilize o componente `JCheckBox`. E para um resumo sobre a pessoa, utilize o componente `JTextArea`, que não foi passado em sala de aula, mas fica como atividade de pesquisa individual para que você aprenda como utilizá-lo. Crie botões para possibilitar a navegação entre os objetos cadastrados. Seu *software* deverá possuir obrigatoriamente: uma classe Pessoa (que será os objetos pessoas que serão armazenados), uma classe Gerencia Pessoa (que deverá possuir o objeto *Collection* para armazenar os objetos do tipo pessoa) e a classe da interface para facilitar a entrada de dados pelo usuário. Ah! Deverá haver também uma tela para visualização das pessoas cadastradas utilizando o componente `JTable`.

Observação: Com exceção do método *main*, você não deverá utilizar outros métodos e nem classes estáticas.

- 2) Escreva um aplicativo que imita um antigo *software* conhecido como "Lúcifer". Este *software* tratava-se de uma brincadeira que funcionava assim:
- quando executado mostrava uma interface arrepiante com os dizeres: "Faça uma pergunta a Lúcifer e ele responderá". E havia um `JTextField` onde o usuário digitaria a pergunta;
 - o usuário contava uma pequena estória arrepiante a um terceiro e dizia que o *software* tinha poderes mágicos. Mas que para funcionar seria necessário "adular" o computador antes de entrar na pergunta propriamente dita. Caso não o fizesse, o computador não responderia. Então pedia para o terceiro formular uma pergunta ao *software*. Obviamente que quem digitava a pergunta era o usuário do *software* e não o terceiro que elaborou a pergunta;
 - Funcionamento do Lúcifer:
 1. Supondo que o terceiro formulou a seguinte pergunta: *qual a cor da minha camisa?* (resposta: branca)
 2. O usuário do *software* vendo a cor da camisa do terceiro digita, sem que o terceiro perceba, ";" (ponto e vírgula) e começa a **digitação da resposta (branca)**, no entanto as letras que vão aparecendo no `JTextField` não correspondem ao que está sendo digitado. Ao invés disto, começa a ser escrito na tela uma frase adulando o computador, tipo: "Meu querido computador",

“Computadorzinho lindo, esperto, sabichão, simpático, maravilhoso...” e etc. Esta frase adulando o computador será escrita automaticamente a cada tecla digitada pelo usuário. Até que ele digite novamente a tecla “;” (ponto e vírgula).

3. No momento que a tecla “;” é apertada pela segunda vez, então tudo o que o usuário do *software* digitar aparecerá no `JTextField`. Assim, o usuário poderá terminar a “adulação” iniciada pela frase automática e digitar a pergunta feita pelo terceiro;
4. Ao terminar a pergunta feita pelo terceiro o usuário diz ao terceiro que “o grande momento chegou” e aperta a tecla `<Enter>`. Então o programa mostra na tela a resposta digitada “por baixo dos panos” ou seja “Branca”.
5. Exemplo:

O usuário começa a digitar a resposta que ficará armazenada numa variável qualquer:

Valor digitado pelo usuário: ;branca;
Valor que aparecerá na tela: Meu quer

Veja que a frase pré-formulada e armazenada em alguma variável do *software* terminará assim que o valor digitado for “;” e neste momento tudo que for digitado deverá aparecer no `JTextField` sem prejudicar a resposta já armazenada.

No exemplo acima o usuário teria que acabar de “adular” o computador, pois, como a resposta é pequena, não apareceu toda a “adulação” pré-formulada.

Observação 1: A resposta digitada pelo usuário pode ser grande, sendo assim, devem existir frases com adulação suficiente para tais respostas. Caso contrário o truque é descoberto facilmente.

Observação 2: Deve haver mais de um tipo de frase de adulação, pois, senão, vendo que a mesma frase se repete toda hora, o terceiro poderá sacar o funcionamento do *software*.

Observação 3: O Lúifer só obedece a você! Rsrtrs...

- 3) Faça uma cópia do exercício número 1, porém, reescreva-o utilizando para armazenamento de dados, um SGBD PostgreSQL. Utilize os padrões já aprendidos.
- 4) Desafio: desenvolva um *software* com características semelhantes ao “Paint” da Microsoft. Escolha três funcionalidades do Paint e implemente-as em seu *software*.