

Execícios de Técnicas de Programação

Linguagem Java - Listeners e GUI

1. Complete o primeiro exercício da lista no arquivo ExercícioGUI - reforço.pdf fazendo o tratamento dos eventos para realizar o salvamento e carregamento do arquivo. Sempre que necessário use um diálogo JOptionPane para receber o nome do arquivo.

Para abrir e ler o arquivo use:

```
FileInputStream in = new FileInputStream(nomeArq);
Scanner sin = new Scanner(in);
while (sin.hasNextLine()) {
    txt.append(sin.nextLine());
    txt.append("\n");
}
in.close();
```

onde.: nomeArq é uma String com o nome do arquivo
txt é o texto (StringBuffer) a ser colocado no JTextArea

Para gravar o arquivo use:

```
FileOutputStream out = new FileOutputStream(nomeArq);
out.write(txt.getBytes());
out.close();
```

onde.: nomeArq é uma String com o nome do arquivo
txt é o texto (String) retirado do JTextArea

Para fechar o programa use:

```
System.exit(0);
```

As rotinas que faltarem para completar o programa devem ser pesquisadas na documentação do Java (<http://download.oracle.com/javase/8/docs/api/index.html>).

2. Complete o segundo e o terceiro exercícios da lista no arquivo ExercícioGUI - reforço.pdf fazendo o tratamento dos eventos e apresentando o resultado no JLabel que contém o valor inicial "0". Para que o resultado caiba no JLabel inicie-o com " 0" (alguns espaços antes do zero).
3. Faça uma busca na Internet sobre a utilização do ActionListener, KeyListener e MouseListener. Apresente um exemplo para cada um destes três tipos de listeners. Discuta onde cada um pode ser usado.
4. Explique qual a diferença entre Listener e Adapter e quando usá-los.

Dicas para resolver o primeiro exercício:

Para resolver o acionamento do botão "Abrir" - a abertura e leitura do arquivo" - do exercício 1 é necessário declarar a variável txt:

```
StringBuffer txt; /* a classe StringBuffer é uma classe para manter cadeias de caracteres
alteráveis, no exercício, cada linha lida do arquivo pode ser anexada usando o método append().
*/
```

Toda vez que se cria uma instância de StringBuffer, a cadeia de caracteres nesta instância é criada vazia. Para o exercício, uma instância de StringBuffer armazenará cada linha anexada vinda do arquivo. Portanto crie uma instância:

```
txt = new StringBuffer();
```

Após o arquivo lido e todas as linhas do arquivo estarem anexadas no txt (StringBuffer), pode-se carregar o arquivo na instância do JTextFiled, para isso use:

```
instanciaDoJTextArea.setText(txt.toString()); /* o método toString() é usado para retornar uma
instância de String que representa a string armazenada no StringBuffer. */
```

Uma vez que os dados foram colocados no JTextArea, a instância de StringBuffer não será mais necessária.

Não se esqueça de criar previamente o arquivo a ser aberto e lido, mas caso o arquivo não exista, mostre mensagem indicando a falta dele usando JOptionPane.

Para o salvamento do arquivo, botão "Salvar", também é necessário declarar a variável txt, mas agora é uma instância de String:

```
String txt;
```

Para retirar o texto do JTextArea, use o método getText():

```
txt = instanciaDoJTextArea.getText(); /* este método retorna uma instância de String com o texto
que está no JTextArea. Por isso não é necessário criar uma instância de String. */
```