SCC0604 – Programação Orientada a Objetos

(Engenharia da Computação)

Prof. Marcelo Manzato

Roteiro da Aula Prática 4 (08/11/2019)

Observações:

I. Deadline para entrega: domingo, dia 10, até às 23:59.

II. A prática deve ser desenvolvida individualmente utilizando a IDE NetBeans.

III. Entregar o trabalho via TIDIA-Ae. Para tanto, acesse o item "Escaninho" do portal da disciplina e crie uma subpasta denominada "Aula Prática 4" dentro da pasta identificada pelo seu nome. Coloque seus arquivos (comprimidos em um arquivo zip, rar, gz, etc) dentro da nova subpasta.

Especificação

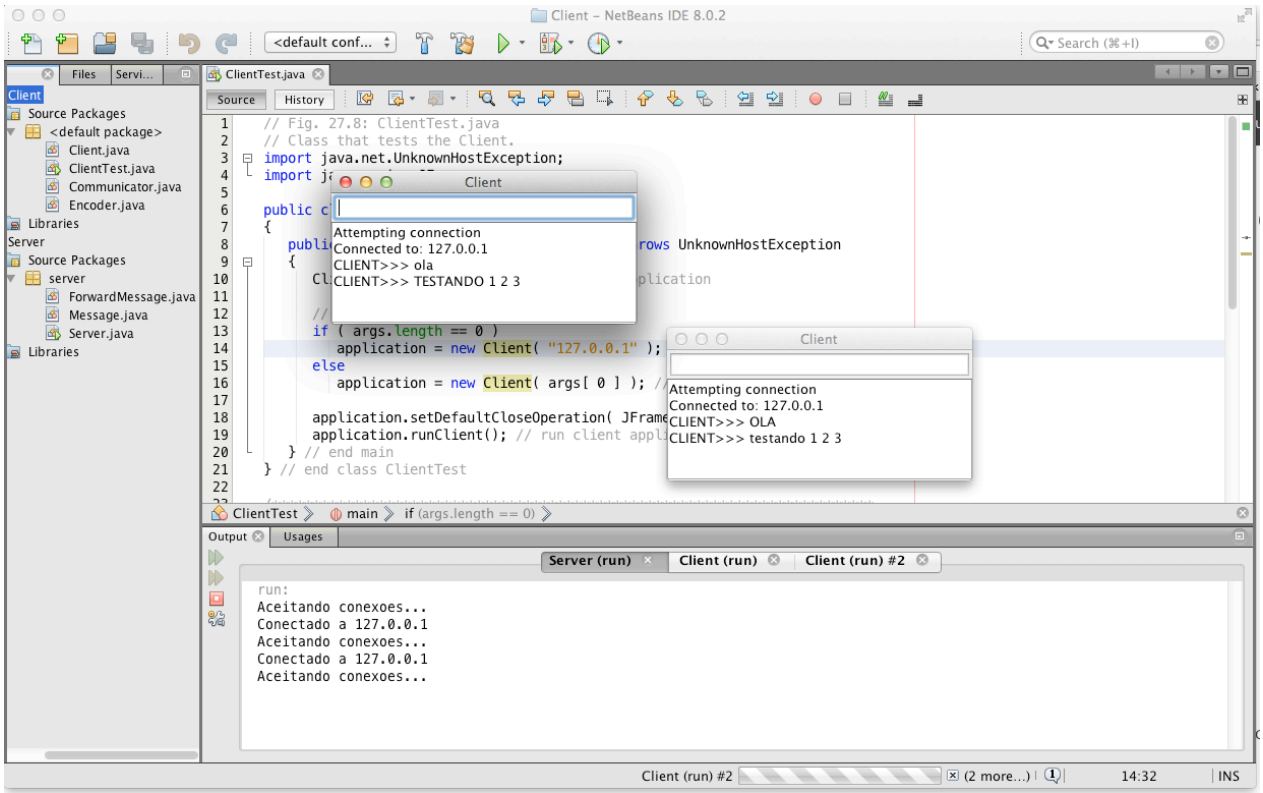
Talvez o mais famoso de todos os esquemas de codificação seja o código Morse, desenvolvido por Samuel Morse em 1832 para utilização com o sistema de telégrafo. O código Morse atribui uma série de pontos e traços para cada letra do alfabeto, para cada dígito e alguns caracteres especiais (como ponto, vírgula, dois-pontos, etc.). Em sistemas orientados para áudio, o ponto representa um som curto e o traço representa um som longo. Outras representações de pontos e traços são utilizadas com sistemas baseados em sinais luminosos e sistemas baseados em sinais de bandeira. A separação entre palavras é indicada por um espaço, ou, simplesmente, a ausência de um ponto ou traço. Em um sistema orientado a som, um espaço é indicado por um tempo curto durante o qual nenhum som é transmitido. A versão internacional do código Morse aparece conforme especificado pela tabela abaixo:

Caractere	*Código*	*Caractere*	*Código*	*Caractere*	*Código*
A	.-	N	-.	*Dígitos*	
B	-...	O	---	1	.----
C	-.-.	P	.--.	2	..---
D	-..	Q	--.-	3	...--
E	.	R	.-.	4	....-
F	..-.	S	...	5	.....
G	--.	T	-	6	-....
H	....	U	..-	7	--...
I	..	V	...-	8	---..
J	.---	W	.--	9	----.
K	-.-	X	-..-	0	-----
L	.-..	Y	-.--		
M	--	Z	--..		

Sua função nesta aula é desenvolver um aplicativo cliente/servidor em que dois ou mais clientes possam enviar mensagens em código Morse entre si por meio de um aplicativo servidor com múltiplas threads. O aplicativo cliente deve permitir que o usuário digite frases em linguagem natural em um *JTextField*. Quando o usuário envia a mensagem, o aplicativo cliente codifica o texto **em código Morse e envia a mensagem codificada**, por meio do servidor, para o outro cliente. Utilize um espaço em branco entre cada letra codificada em Morse, e três espaços em branco entre cada palavra codificada em Morse.

Quando as mensagens são recebidas, elas devem ser decodificadas e exibidas como caracteres normais. O cliente deve ter um *JTextField* para digitar e um *JTextArea* para exibir mensagens dos outros clientes.

A figura abaixo ilustra o servidor rodando em background no NetBeans, e dois clientes se comunicando via código Morse.



Observação 1:

Ao desenvolver seu programa, lembre-se dos conceitos vistos na aula de Projeto de Classes e UML. Por exemplo: ao implementar a comunicação por redes, crie uma classe chamada "Comunicador", que será responsável por manter toda comunicação. Disponibilize funções como conectar(), desconectar(), enviarMensagem(), receberMensagem(), etc., que serão usadas no programa principal. Faça o mesmo para a classe Codificador.

Observação 2:

Você deverá utilizar a tabela acima para realizar a codificação/decodificação da mensagem. Para isso, você poderá escolher uma das duas alternativas abaixo:

- Inserir a tabela acima manualmente no seu programa, usando uma estrutura de dados eficiente (como HashMap's)
- Ler automaticamente a tabela acima que está disponibilizada no arquivo "codigo_morse.txt" (Tidia-Ae), e armazenar na memória usando também uma estrutura de dados eficiente.

Observação 3:

Algumas funções que podem ser úteis na hora de implementar:

```
String[] tokens = str.split(" "); // Separa a string str em um array de strings por meio de 1 espaço entre os tokens
```

```
String[] tokens = str.split("   "); // Separa a string str em um array de strings por meio de 3 espaços entre os tokens
String c = str.substring(1, 2); // Retorna uma substring contendo apenas o 2º caractere da string str
String s = str.toUpperCase(); // Retorna uma string a partir de str com todos os caracteres em letra maiúscula
```

Bom trabalho!