

Aula 20 - Exercício 1

- b. Executar, somente, a classe ***SimpleClientTest.java***, verificar e registrar o que ocorre. Explique, em detalhes, o que ocorre;

Como o SimpleClientTest foi executado sozinho sem ter o servidor aberto, o código joga uma exceção por não ter servidor para se conectar e exibe a mensagem de Falha na conexão.

```
PS C:\Users\Leonardo\OneDrive\Documents\Linguagens de Programação\Java\Orientado a Objeto> & 'C:\Program Files\Java\jdk-17\bin\java.exe' '@C:\Users\Leonardo\AppData\Local\Temp\cp_5emzy2g3w16ggthzrvrzl7pbt.argfile' 'Aula_20.SimpleClientTest'  
Falha na comunicação
```

- c. Executar a classe ***SimpleServerTest.java***, verificar e registrar o que ocorre. Explique, em detalhes, o que ocorre;

Executar somente o SimpleServerTest vai abrir o servidor e exibir a mensagem de servidor iniciado, mas ele ficará preso nessa mensagem para sempre pois o servidor foi programado para esperar um cliente para definir se a conexão foi bem sucedida ou mal sucedida. Como o cliente nunca vai se conectar, nenhuma exceção é feita e o código continuará rodando.

```
PS C:\Users\Leonardo\OneDrive\Documents\Linguagens de Programação\Java\Orientado a Objeto> & 'C:\Program Files\Java\jdk-17\bin\java.exe' '@C:\Users\Leonardo\AppData\Local\Temp\cp_5emzy2g3w16ggthzrvrzl7pbt.argfile' 'Aula_20.SimpleServerTest'  
Servidor iniciado na porta 3334
```

- d. Sem encerrar a execução da classe ***SimpleServerTest.java***, executar, a seguir, a classe ***SimpleClientTest.java***, digitando algumas mensagens na interface com o usuário, uma após a outra e, por fim, optar por sair. Verificar e registrar o que ocorre. Explique, em detalhes, o que ocorre;

Com o servidor ligado antes do cliente, a conexão é bem sucedida dos dois lados. As mensagens são exibidas nos terminais do cliente e servidor, e quando é digitada a mensagem “sair” o cliente é interrompido, e o servidor fecha a conexão com sucesso e depois termina sua execução.

Cliente:

```
Cliente: 127.0.0.1:3334 conectado ao servidor!  
Mensagens para o servidor:  
Mensagem 1  
Mensagem 2
```

Servidor:

```
Servidor iniciado na porta 3334  
Cliente no IP 127.0.0.1 conectado ao servidor  
Mensagem 1  
Mensagem 2  
Servidor finalizado!!
```

- e. Sem encerrar a execução da classe ***SimpleServerTest.java***, executar, a seguir, a classe ***SimpleClientTest.java***, digitando uma mensagem, apenas, na interface com o usuário e não optar por sair. Sem encerrar a execução da classe ***SimpleServerTest.java***, nem da primeira instância de execução da classe ***SimpleClientTest.java***, executar outra instância da classe ***SimpleClientTest.java***. Digitar algumas mensagens na segunda instância da classe ***SimpleClientTest.java***, sem sair dela, verificar e registrar o que ocorre. Repetir a digitação para a primeira instância da classe ***SimpleClientTest.java***, sem sair dela também. Explique, em detalhes, o que ocorre;

Durante o experimento, o servidor exibe normalmente as mensagens do primeiro cliente, mas as mensagens do segundo cliente são ignoradas pelo servidor e as últimas mensagens do primeiro cliente após abrir o segundo cliente são exibidas novamente. Isso ocorre porque, mesmo que a conexão seja aceita, o servidor só tem suporte para um Socket, então o segundo Socket nunca é aceito.

Cliente 1:

```
Cliente: 127.0.0.1:3334 conectado ao servidor!  
Mensagens para o servidor:  
Mensagem 1  
Mensagem 3  
Mensagem 4
```

Cliente 2:

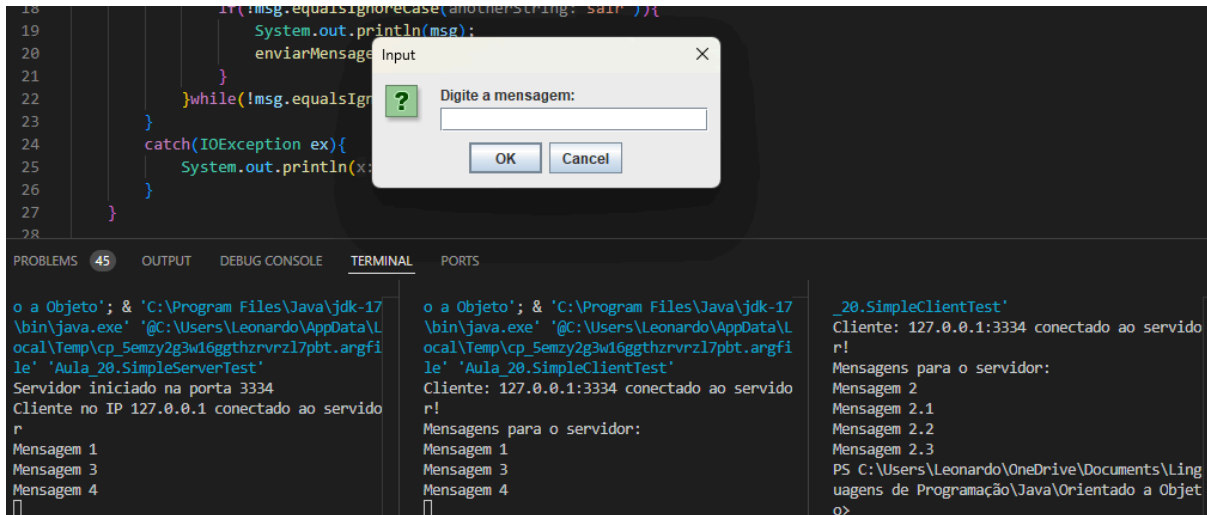
```
Cliente: 127.0.0.1:3334 conectado ao servidor!  
Mensagens para o servidor:  
Mensagem 2  
Mensagem 2.1  
Mensagem 2.2  
Mensagem 2.3
```

Servidor:

```
Servidor iniciado na porta 3334  
Cliente no IP 127.0.0.1 conectado ao servidor  
Mensagem 1  
Mensagem 3  
Mensagem 4  
Servidor finalizado!!
```

- f. Encerrar a execução da segunda instância da classe ***SimpleClientTest.java***, digitando algumas mensagens na primeira instância da classe ***SimpleClientTest.java***. Verificar e registrar o que ocorre. Explique, em detalhes, o que ocorre;

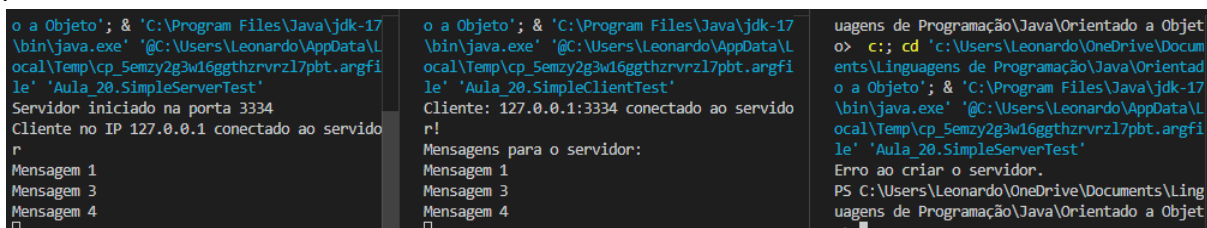
O funcionamento continua normalmente, pois como a Socket para segunda execução não foi criada, a Socket do servidor não é fechada com o fechamento da conexão da segunda execução do cliente.



The screenshot shows an IDE with a Java file open. A dialog box titled "Input" is displayed in the center, asking the user to "Digite a mensagem:" (Enter the message:). The dialog has "OK" and "Cancel" buttons. Below the dialog, the "TERMINAL" tab is active, showing the output of a Java application. The output indicates that a server is running on port 3334 and a client is connected. The client sends four messages: "Mensagem 1", "Mensagem 3", "Mensagem 4", and "Mensagem 2". The server responds with "Mensagem 2.1", "Mensagem 2.2", and "Mensagem 2.3".

- g. Executar uma segunda instância da classe ***SimpleServerTest.java***, ainda com a primeira instância dessa classe em execução. Verificar e registrar o que ocorre. Explique, em detalhes, o que ocorre.

Há um erro na criação do servidor, pois não é possível abrir dois serviços iguais na mesma porta do servidor.



The screenshot shows the same IDE as before, but now the terminal window displays an error message. The error message is "Erro ao criar o servidor." (Error creating the server.) This occurs because the server is already running on port 3334, and attempting to start another instance of the same server on the same port results in a conflict.

Se mudarmos a porta e rodar o código novamente, o segundo servidor abrirá:

11

Run | Debug

12 public static void main(String

13 try{

14 ServerSocket servidor

15 System.out.println("Se

16

17 //Criando canal de comunicação servidor-cliente

18 Socket cliente = servidor.accept();

19 System.out.println("Cliente no IP " + cliente.getInetAddress().getHostAddress() + " conectado ao servidor");

Input

?

Digite a mensagem:

mensagem teste após server 2 ser criado

OK

Cancel

PROBLEMS 45

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

\bin\java.exe' '@C:\Users\Leonardo\AppData\Local\Temp\cp_5emzy2g3w16ggthzrvrzl7pbt.argfile' 'Aula_20.SimpleServerTest'

Servidor iniciado na porta 3334

Cliente no IP 127.0.0.1 conectado ao servidor

r

Mensagem 1

Mensagem 3

Mensagem 4

Mensagem Teste após server 2 ser criado

\bin\java.exe' '@C:\Users\Leonardo\AppData\Local\Temp\cp_5emzy2g3w16ggthzrvrzl7pbt.argfile' 'Aula_20.SimpleClientTest'

Cliente: 127.0.0.1:3334 conectado ao servidor!

Mensagens para o servidor:

Mensagem 1

Mensagem 3

Mensagem 4

Mensagem Teste após server 2 ser criado

o>

PS C:\Users\Leonardo\OneDrive\Documents\Linguagens de Programação\Java\Orientado a Objetos> cd 'c:\Users\Leonardo\OneDrive\Documents\Linguagens de Programação\Java\Orientado a Objeto'; & 'C:\Program Files\Java\jdk-17\bin\java.exe' '@C:\Users\Leonardo\AppData\Local\Temp\cp_5emzy2g3w16ggthzrvrzl7pbt.argfile' 'Aula_20.SimpleServerTest'

Servidor iniciado na porta 3335