**T08 – Conectando sem as classes do tutorial JDBC**

Igor Augusto Reis Gomes – 12011BSI290 – [igor.augusto@ufu.br](mailto:igor.augusto@ufu.br)

Heitor Guimarães Da Fonseca Filho – 12011BSI203 – [heitor.filho@ufu.br](mailto:heitor.filho@ufu.br)

Obs.: o resumo mencionado no item 7 está como comentários ao longo do próprio código.

1. **Inicialmente, a quantidade de bibliotecas a ser importada será menor quando comparada com as bibliotecas da classe MyQueries:**
2. **Para implementação do código você precisa de, pelo menos, uma classe e quatro métodos:**

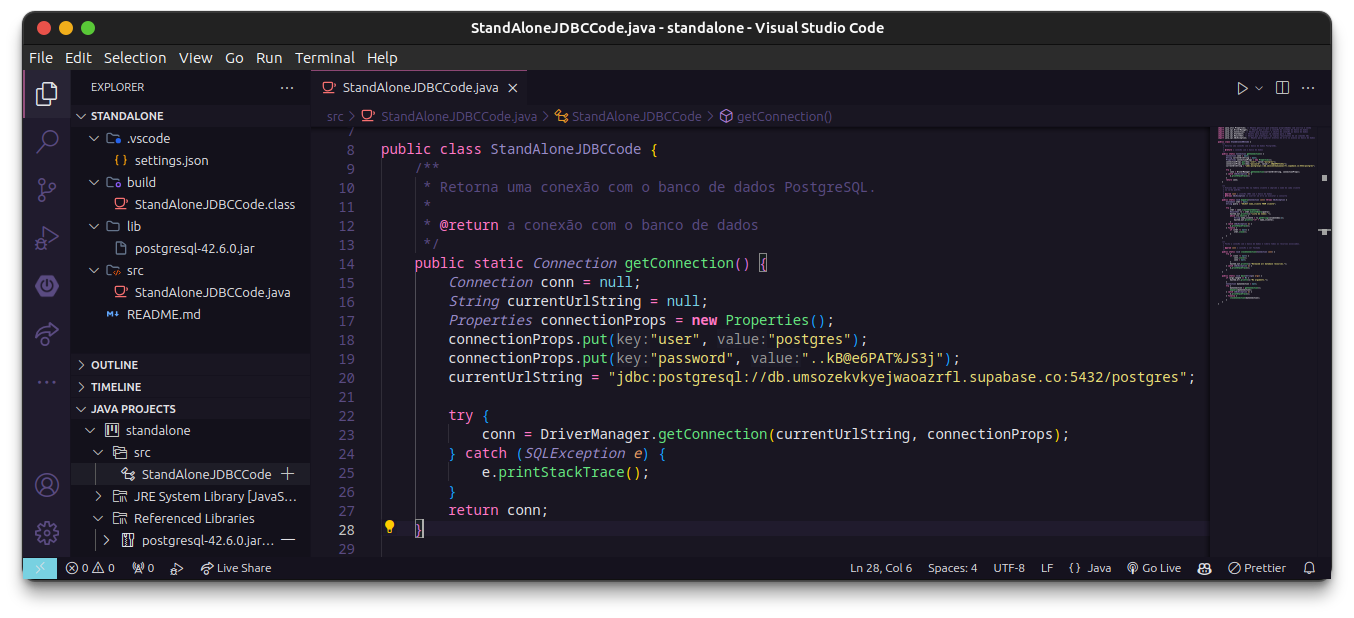
Tanto as bibliotecas foram importadas, quanto a estrutura dos quatro métodos foram inseridos:



1. **O código abaixo mostra um esboço do método criador do objeto armazenador da conexão ao banco de dados. Perceba que este método retorna um objeto de conexão do tipo Connection, cuja biblioteca da classe foi incorporada no início do código.**

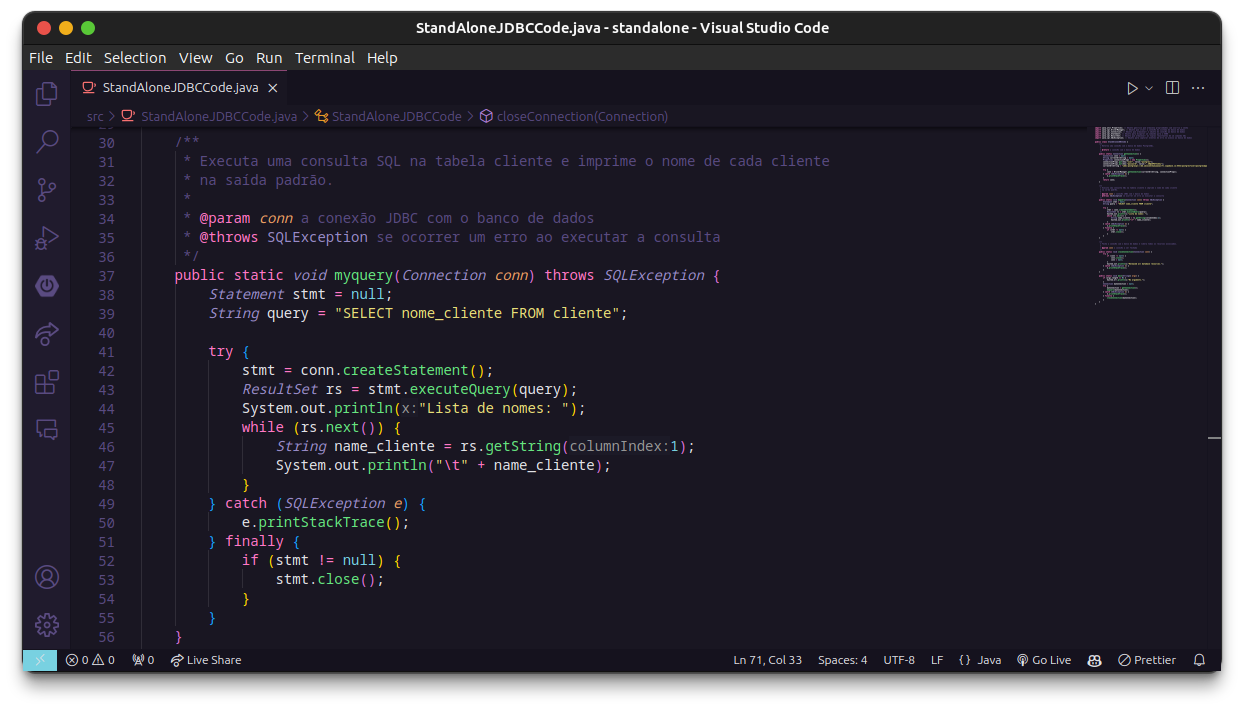
**Termine de implementar este código exemplo na sua classe de consulta StandAloneJDBCCode, dentro do arquivo java de mesmo nome.**

Adicionei o corpo do método e as propriedades do banco:



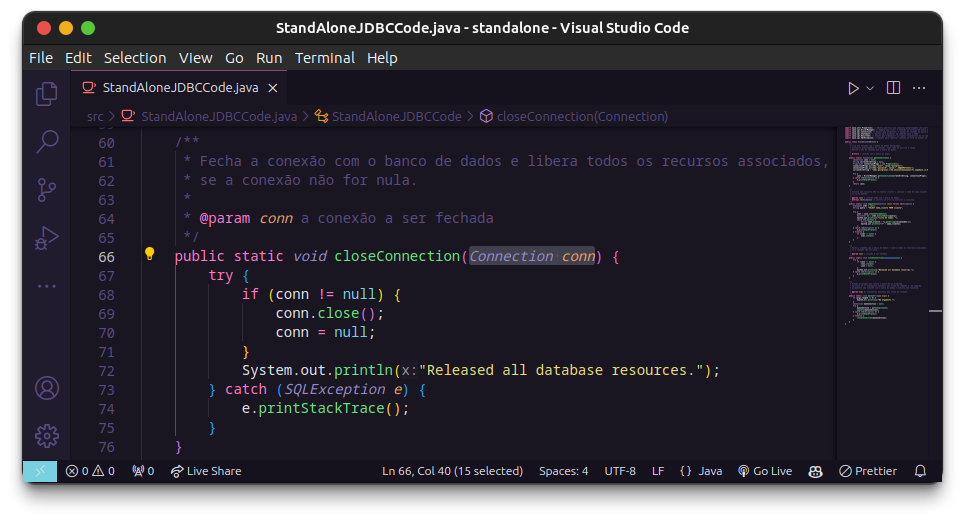
1. **Termine de implementar este código exemplo na sua classe de consulta StandAloneJDBCCode, dentro do arquivo java de mesmo nome, provendo um nome correto de uma tabela do seu banco de dados, bem como tratando os dados retornados no objeto do tipo ResultSet.**

Decidi por fazer uma consulta simples na tabela cliente, que retorna todas as linhas de nome\_cliente:



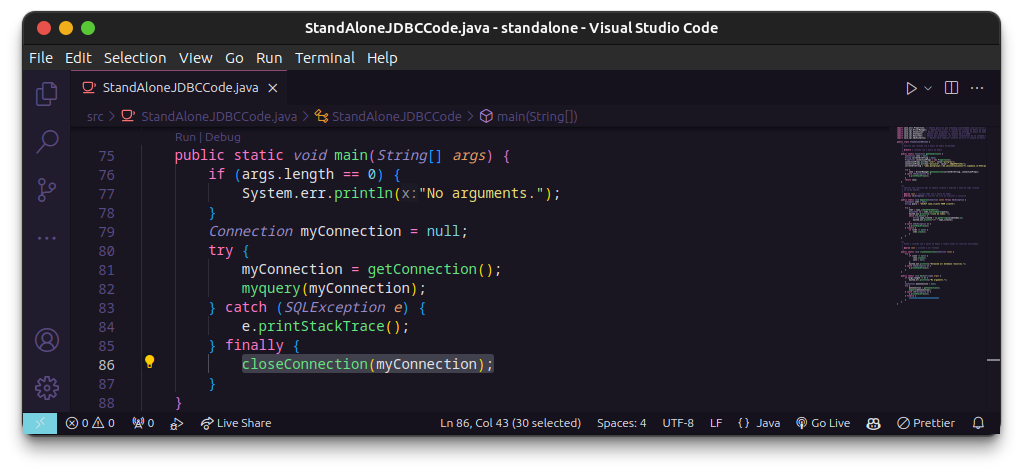
1. **O código acima possui um erro. Transcreva o código para o programa StandAloneJDBCCode.java e conserte este erro. Se tiver dificuldade em encontrar o erro então espere até a compilação do código completo do programa StandAloneJDBCCode.java e o compilador do java pode fornecer alguma informação adicional para você corrigir o erro:**

O erro está na falta do argumento/parâmetro de conexão que deveria ser passado na assinatura do método: **closeConnection(Connection conn)**:

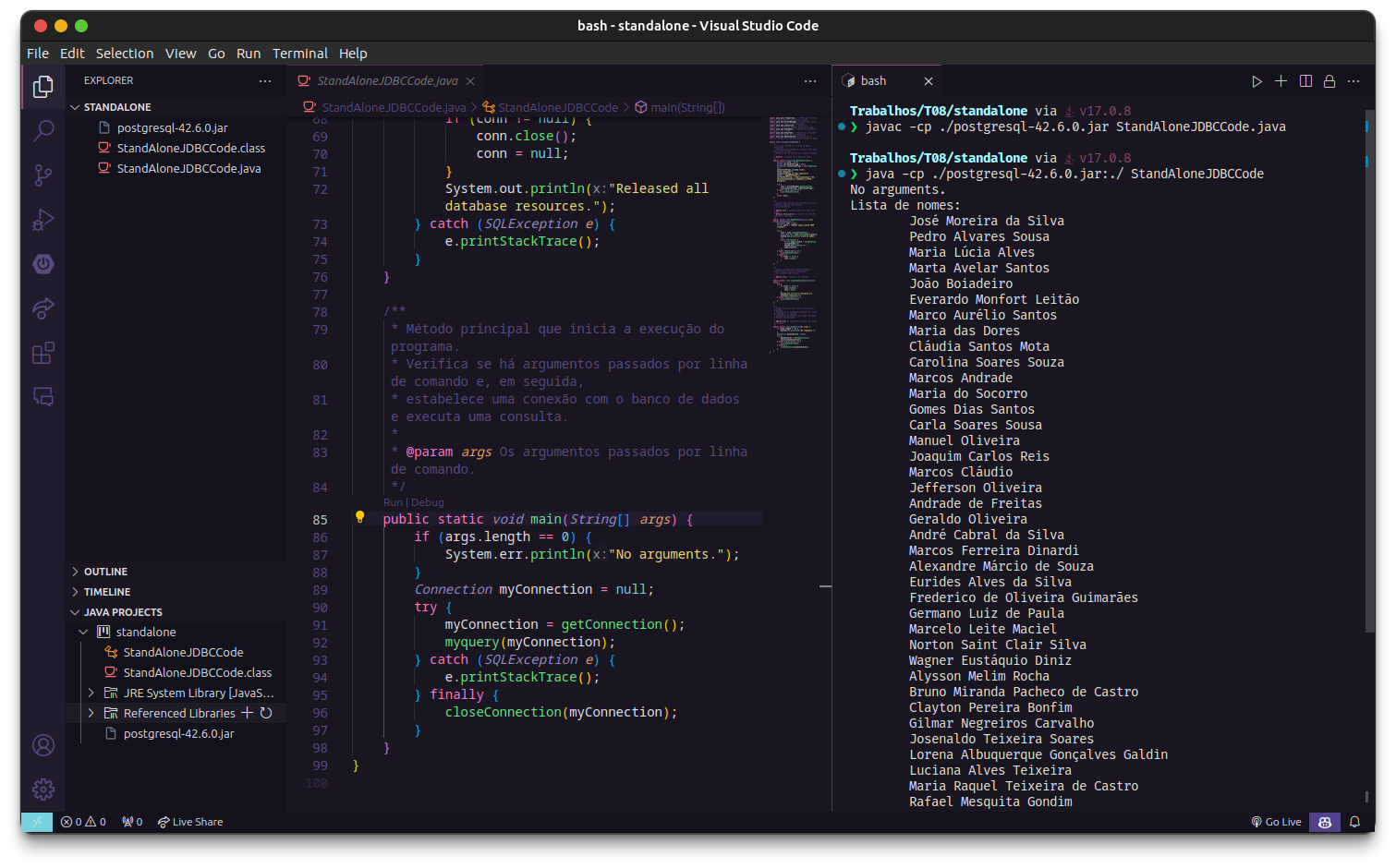


1. **Na sequência, haverá a instanciação de um objeto do tipo Connection que inicialmente está nulo. O método StandAloneJDBCCode.getConnection() fará o trabalho de criar o objeto de conexão ao banco de dados. Por fim, o objeto de conexão é passado para o método deste programa que cumprirá o objetivo maior, realizar a consulta ao banco e tratar os dados como for necessário. Passe este código para nosso programa e execute-o. Ah, tem um erro neste código também, encontre-o.**

O erro está dentro do bloco finally do try, no qual tenta-se encerrar a conexão, porém a método closeConnection espera receber um parâmetro do tipo Connection, no qual não é feito (devido ao erro do item anterior), assim, para que funcione basta inserir a variável como argumento da função:



Compilação e execução do programa como foi solicitado, no lado direito da captura:



Além disso, com intuito de simplificação, descobri que é possível compilar e executar o programa sem a necessidade de digitar manualmente o caminho da classe e do driver do postgres, dado que o VSCode cria um arquivo settings.json no qual é possível colocar diretamente o caminho das bibliotecas referenciadas, neste caso, o driver que coloquei dentro da pasta “lib”. Como é possível observar abaixo, na janela do canto inferior esquerdo com o arquivo .json, e a com a execução direta na direita.

