**Arquitetura de Software – Tarefa 2\_01**

Prof. **Dr. Aparecido V. de Freitas** – Ciência da Computação

Aluno Luis Eduardo Leoni Gisoldi, Turma 03AN

Em uma **arquitetura monolítica**, todos os componentes do software (interface do usuário, lógica de negócios, acesso a dados, etc.) são **integrados** em um único **programa**, ou "**monolito**". Essa abordagem trata a aplicação como uma unidade indivisível e atômica.

Características de sistemas de software com Arquitetura Monolítica:

✓ **Integração Fortemente Acoplada**: Todos os componentes e funcionalidades estão fortemente interligados e operam como uma única unidade.

✓ **Simplicidade de Desenvolvimento e Implantação**: Inicialmente, é mais simples de desenvolver e implantar, pois tudo é centralizado.

✓ **Escalabilidade Horizontal**: Para escalar uma aplicação monolítica, geralmente replicamos o monólito inteiro.

Os sistemas de mainframe da década de 1980 são bons exemplos de arquitetura monolítica, uma vez que possuem as seguintes características:

✓ **Centralização**: Os mainframes eram sistemas centralizados que executavam todas as operações de computação, armazenamento e processamento de dados em uma única máquina.

✓ **Software e Hardware Integrados**: O software desenvolvido para esses sistemas era frequentemente específico para o hardware do mainframe e funcionava como um único sistema integrado.

✓ **Aplicações Empresariais**: Eram comumente usados para aplicações empresariais críticas, como processamento de transações bancárias, sistemas de folha de pagamento e gerenciamento de registros.

Os sistemas de mainframe da década de **1980** são bons exemplos de **arquitetura monolítica**, uma vez que possuem as seguintes características:

✓ **Software Unificado**: Os aplicativos eram muitas vezes grandes programas unificados, onde diferentes funções e módulos estavam todos contidos dentro de uma única base de código e ambiente de execução.

✓ **Manutenção e Atualização**: Atualizar ou modificar uma parte do sistema muitas vezes significava ter que reimplementar ou reiniciar o sistema inteiro.

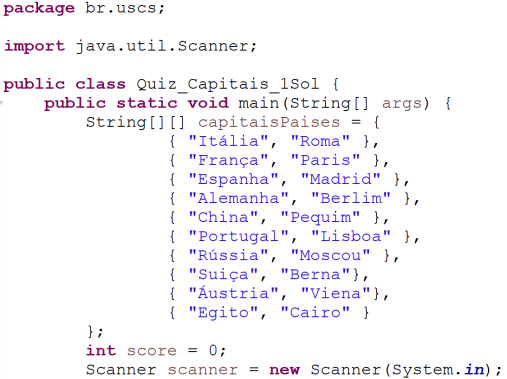
✓ **Escalabilidade e Flexibilidade Limitadas**: Devido à sua natureza integrada, escalar ou adaptar esses sistemas para novas necessidades era um processo complexo e muitas vezes dispendioso.

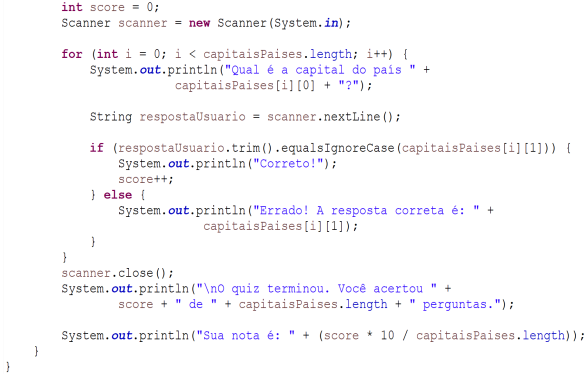
Uma aplicação desktop desenvolvida em **C** que use diversas funções, sendo estruturada em módulos, pode ser um exemplo de **arquitetura monolítica**, pelas seguintes razões:

✓ **Módulos**: A aplicação pode ser organizada em diferentes módulos ou arquivos de código-fonte em C, cada um responsável por uma funcionalidade específica (por exemplo, processamento de dados, interface do usuário, operações de arquivo).

✓ **Compilação**: Apesar da estrutura modular no código-fonte, todos os módulos são compilados juntos para formar um único executável.

O código abaixo representa uma aplicação desktop, desenvolvida em **Java**, estruturada com uma **arquitetura monolítica**:





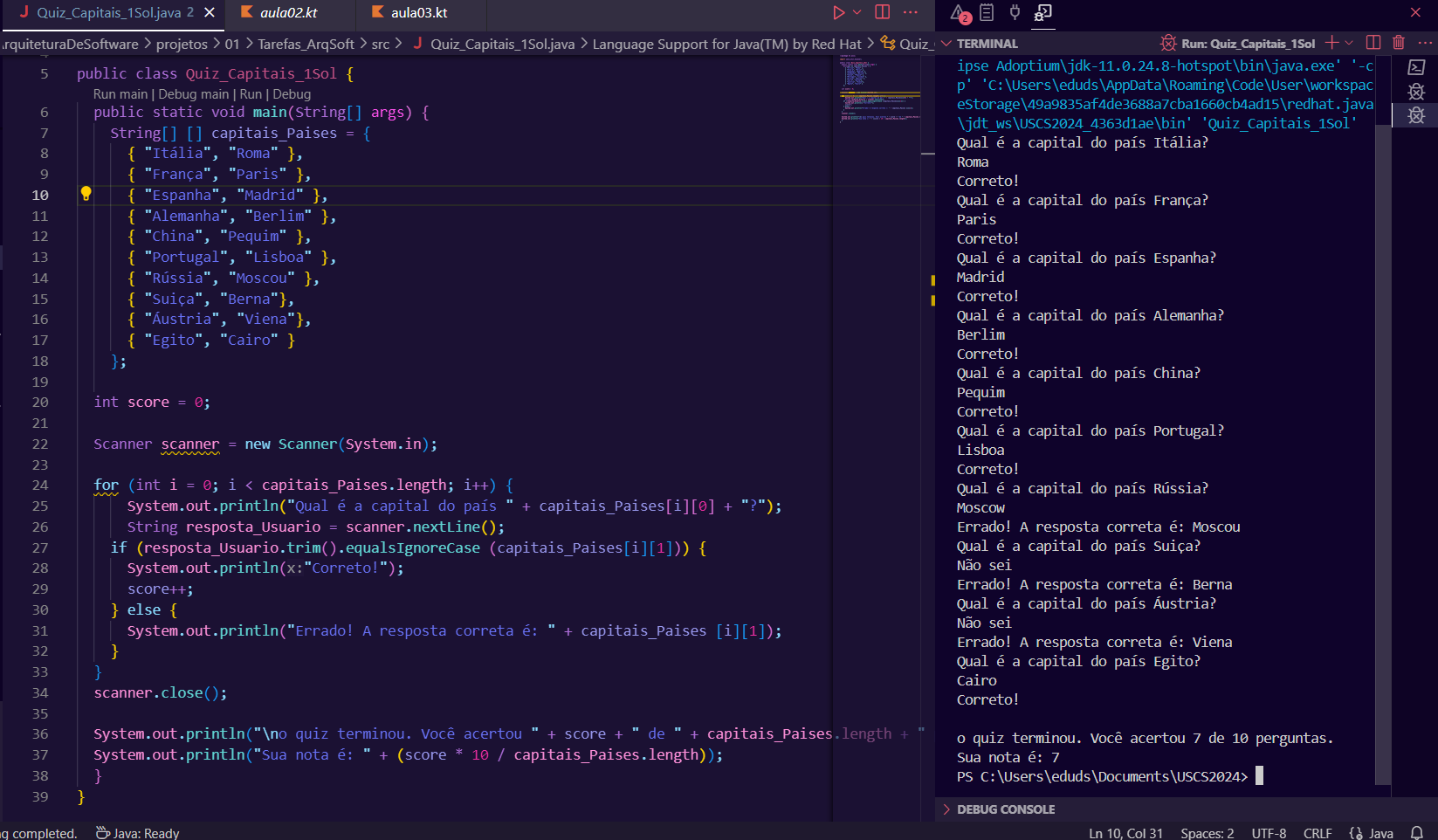
Pede-se:

A) Qual a arquitetura de software da aplicação ?

**R:** A arquitetura usada é a Monolítica.

B) Implemente o código em alguma IDE, por exemplo, **ECLIPSE**, e proceda à execução do código:

**R:** Execução em java na IDE VSCODE:



C) Altere o código para incluir a seguinte funcionalidade: Se o usuário acertar um número menor que 6 questões, o programa deverá enviar a mensagem: “Baixo Desempenho”. Caso o usuário acerte entre seis (inclusive) e oito (inclusive) questões, o programa deverá enviar a mensagem “Bom desempenho”. Caso o usuário acerte mais que oito questões, o programa deverá enviar a mensagem: “Ótimo desempenho”.

**R:**

