Software Engineering Concepts

SE - Lab 05

Topic: Threads communication and synchronization
Estimation: 15 hours.

Lab Overview

The purpose of this lab is to deepen your understanding of multithreading and concurrency in Java through practical implementation of various concepts.

Task 1: Implement a Thread-Safe Bank Account

Create a `BankAccount` class that supports concurrent deposits and withdrawals without race conditions.

Task 2: Producer-Consumer with Bounded Buffer

Implement a producer-consumer system where producers add items to a bounded buffer and consumers remove them.

Task 3: Reader-Writer Problem

Implement a solution where multiple readers can access a resource simultaneously, but writers need exclusive access.

Task 4: Dining Philosophers Problem

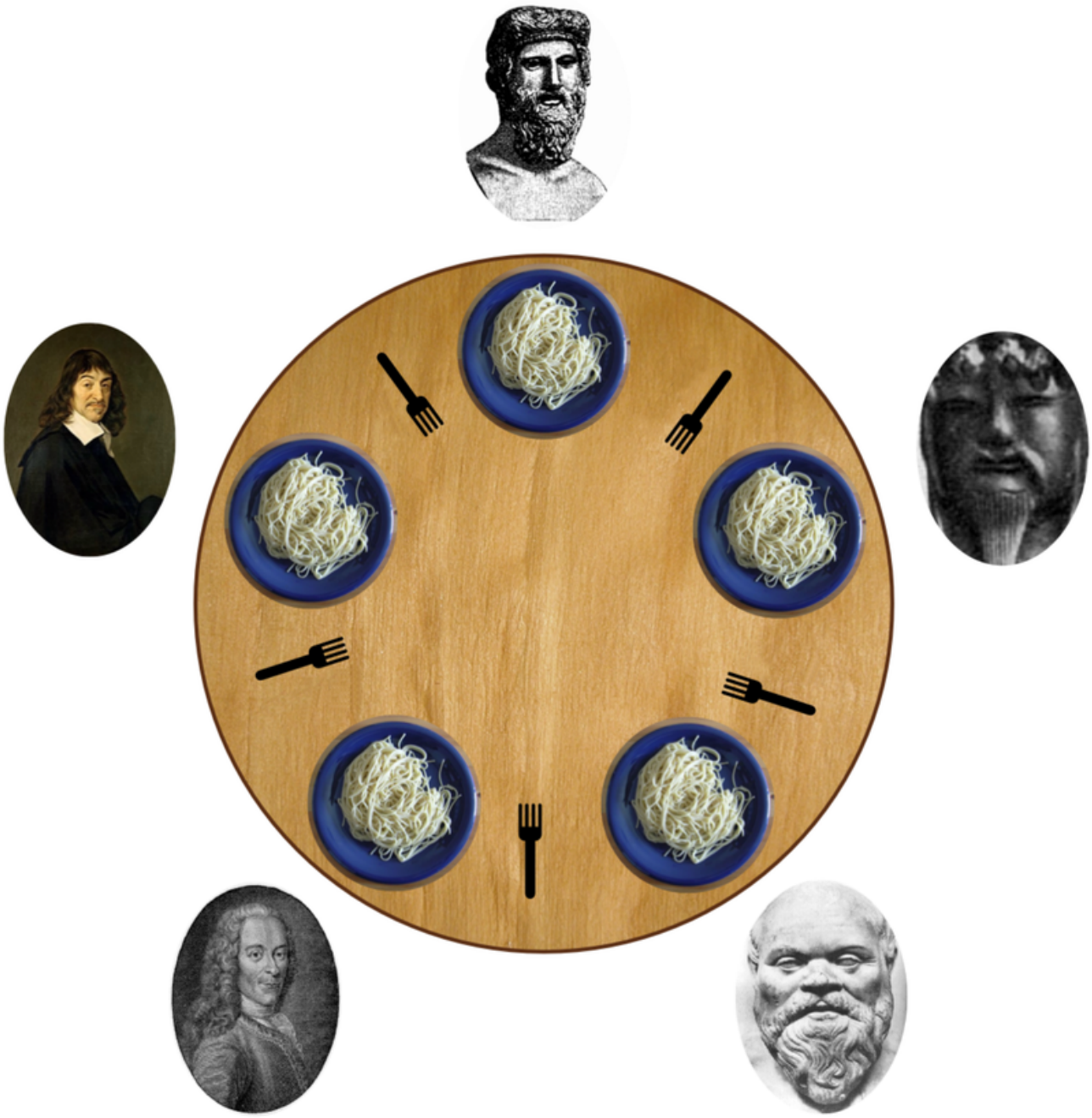
Implement the classic Dining Philosophers problem:

- 5 philosophers sitting around a circular table
- 5 forks (one between each pair of philosophers)
- Each philosopher alternates between thinking and eating
- To eat, a philosopher needs both left and right forks
- Prevent deadlock and starvation

Requirements:

- Each philosopher should eat at least 3 times
- Use proper synchronization to avoid deadlock
- Display philosopher states (thinking/hungry/eating)
- Implement timeout mechanism to break potential deadlocks

- Run simulation for 2 minutes and show statistics



Submission Guidelines:

- Create separate Java files for each task
- Include comments explaining your approach
- Add README.md with execution instructions
- Submit all source code and output screenshots
- Deadline: One week from assignment date