

Módulo 1 - Laboratório 3

Introdução à programação concorrente em Java

Programação Concorrente (ICP-361)
Prof. Silvana Rossetto

¹Instituto de Computação/CCMN/UFRJ - 2023-1

Introdução

O objetivo deste Laboratório é **introduzir a programação concorrente em Java**. Para cada atividade, siga o roteiro proposto e responda às questões colocadas.

Atividade 1

Objetivo: Mostrar como criar um programa concorrente em Java. Em Java, a classe `java.lang.Thread` oferece métodos para criar threads, iniciá-las, suspendê-las e esperar pelo seu término.

O primeiro passo para criar uma aplicação concorrente em Java é **criar uma classe que implementa a interface Runnable**. Essa interface define o método `run()`, responsável pelo código que deverá ser executado pela thread.

O segundo passo é **transformar o objeto Runnable em uma thread**, chamando o construtor da classe `java.lang.Thread` com o objeto `Runnable` como argumento.

O terceiro passo é iniciar as threads criadas, usando o método `start()` da classe `Thread`.

Abra o arquivo **HelloThread.java** e siga o roteiro abaixo.

1. Leia o programa e tente entender o que ele faz (**acompanhe a explicação da professora na vídeo-aula**).
2. Compile o programa fazendo `javac HelloThread.java` no terminal.
3. Execute o programa **várias vezes** (fazendo `java HelloThread`) e observe os resultados impressos na tela. **Há mudanças na ordem de execução das threads? Por que isso ocorre?**

Atividade 2

Objetivo: Mostrar outra forma de criar threads em Java, usando herança.

Roteiro: Outra forma de criar programas concorrentes em Java é estendendo a classe `Thread`. Abra o arquivo **OlaThread.java** e siga o roteiro abaixo.

1. Primeiro, encontre as principais diferenças em relação ao programa `HelloThread.java` (**acompanhe a explicação da professora na vídeo-aula**).
2. Compile e execute o programa **várias vezes**, e observe os resultados impressos.

Atividade 3

Objetivo: Implementar um **programa concorrente em Java** para calcular a série que gera o valor de π mostrada abaixo. Divida a tarefa entre as threads de forma balanceada.

$$\pi = 4 \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots + (-1)^n \frac{1}{2n+1} + \cdots \right)$$

Roteiro:

1. Na thread *main*, crie e dispare as threads, aguarde todas as threads terminarem e verifique se o cômputo final está correto (compare com o valor de π);
2. Teste seu programa variando o valor de n e o número de threads.

Entrega: Disponibilize o código implementado na **Atividade 3** em um ambiente de acesso remoto (GitHub ou GitLab). **Use o formulário de entrega desse laboratório para enviar o link do repositório do código implementado.**