

### *Questões de completar*

1. Em programação concorrente, uma thread é \_\_\_\_\_ dentro de um processo que pode ser executada de forma independente.
2. O método \_\_\_\_\_ da classe Thread em Java é usado para iniciar a execução de uma nova thread.

### *Questões objetivas*

3. Qual das alternativas melhor descreve o conceito de uma thread?
  - a) Uma thread é um processo independente que não compartilha memória com outras threads.
  - b) Uma thread é uma unidade de execução dentro de um processo, compartilhando recursos do processo.
  - c) Uma thread é uma função especial que roda paralelamente sem restrições de CPU.
  - d) Uma thread é um conjunto de processos que rodam em paralelo sem compartilhamento de memória.
4. No contexto de Java, qual das seguintes classes ou interfaces é usada para criar threads?
  - a) Runnable e Thread
  - b) Thread e Executor
  - c) Runnable e ExecutorService
  - d) Process e ThreadGroup
5. O que acontece quando chamamos o método .start() em um objeto de thread em Java?
  - a) A thread é finalizada imediatamente.
  - b) O método run() é executado diretamente no mesmo contexto da thread principal.
  - c) A JVM cria uma nova thread e executa o método run().
  - d) Um erro é gerado, pois start() só pode ser chamado uma vez.
6. Qual é um problema comum ao trabalhar com múltiplas threads compartilhando recursos?
  - a) O aumento da performance garantida.
  - b) O bloqueio mútuo (deadlock).
  - c) A execução determinística do código.
  - d) A impossibilidade de sincronização.

7. Como podemos evitar condições de corrida (race conditions) ao trabalhar com múltiplas threads?
- a) Evitando o uso de threads em sistemas complexos.
  - b) Garantindo que cada thread tenha sua própria cópia dos dados compartilhados.
  - c) Utilizando sincronização, como synchronized em Java.
  - d) Apenas rodando uma thread por vez na aplicação.
8. Em Java, qual das seguintes instruções pode ser usada para pausar temporariamente uma thread?
- a) Thread.stop()
  - b) Thread.yield()
  - c) Thread.sleep(millis)
  - d) Thread.run()

*Questão de Verdadeiro ou Falso*

9. ( ) O método run() de uma thread pode ser chamado diretamente, e isso terá o mesmo efeito que chamar .start().

*Questão subjetiva*

10. Explique o conceito de **sincronização de threads** e como ele pode ser utilizado para evitar problemas de concorrência. Dê um exemplo prático de implementação em Java.

---

Gabarito

- 1. uma unidade de execução
- 2. start()
- 3. **b)** Uma thread é uma unidade de execução dentro de um processo, compartilhando recursos do processo.
- 4. **a)** Runnable e Thread
- 5. **c)** A JVM cria uma nova thread e executa o método run().
- 6. **b)** O bloqueio mútuo (deadlock).
- 7. **c)** Utilizando sincronização, como synchronized em Java.
- 8. **c)** Thread.sleep(millis)
- 9. **Falso** – Chamar run() diretamente executa o código na mesma thread atual, enquanto start() cria uma nova thread.
- 10. Resposta subjetiva, mas deve abordar sincronização, synchronized, Lock, Semaphore, e exemplos práticos.