



Programação em Java

18-08-2021



Tecnologia JAVA

Algoritmos
Celina Lopes



Sumário

- Threads
- Multi Threading



Threads

- ▶ *Threads* são diversas vezes denominadas por *lightweight processes*;
- ▶ As *threads* existem dentro de um processo;
- ▶ Cada processo tem pelo menos uma *thread*;
- ▶ As múltiplas *threads* de um processo partilham os seus recursos (memória, ficheiros abertos, ...).



Threads

- ▶ Cada thread está associada a uma instância da classe Thread;
- ▶ Existem duas estratégias base para utilizar objetos Thread na criação uma aplicação concorrente;
 - ▶ Controlar diretamente a gestão e a criação de threads, instanciando uma thread sempre que é necessário iniciar uma tarefa assíncrona;
 - ▶ passar a gestão abstrata de threads para um *executor*.



Threads - estados

- ▶ **new** – início do ciclo de vida de uma thread;
- ▶ **runnable** – a thread está a executar a sua tarefa;
- ▶ **waiting** – uma thread encontra-se neste estado enquanto espera que outra thread execute a sua tarefa;
- ▶ **time waiting** – uma thread *runnable* entra neste estado por um intervalo de tempo. Volta ao estado *runnable* quando o tempo expira;
- ▶ **blocked** – a thread passa ao estado *blocked* quando tenta executar uma tarefa que não pode ser terminada imediatamente e deve esperar temporariamente até poder terminar a tarefa;
- ▶ **terminated** – a thread entra no estado *terminated* quando completa uma tarefa ou termina de outra forma, por exemplo, através da ocorrência de um erro.



Threads

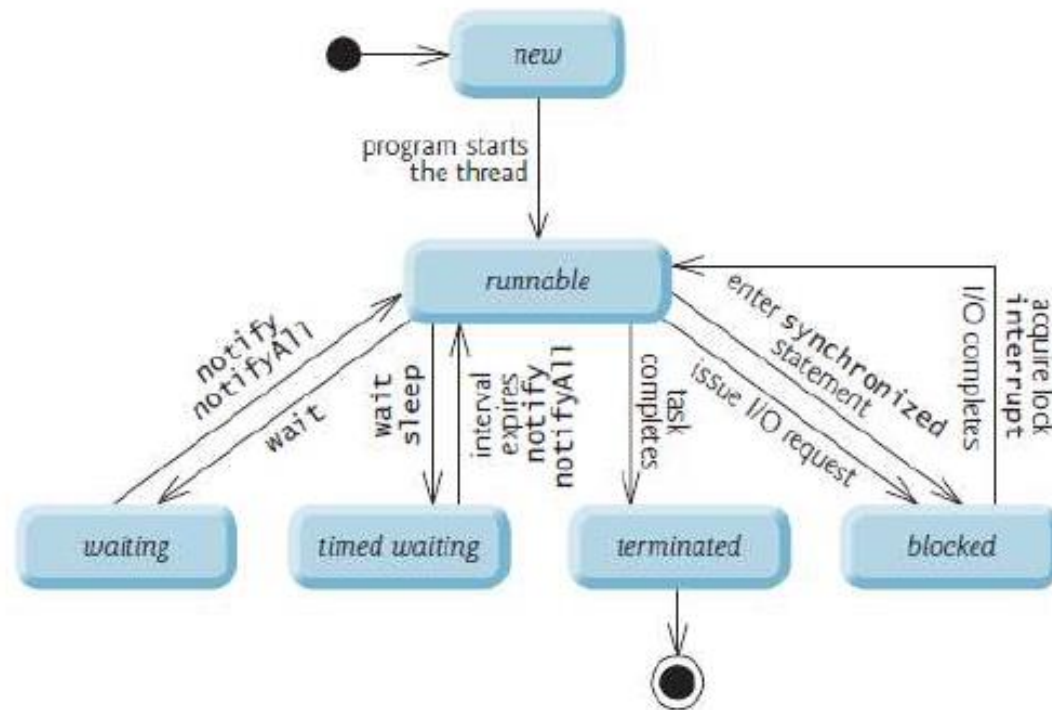
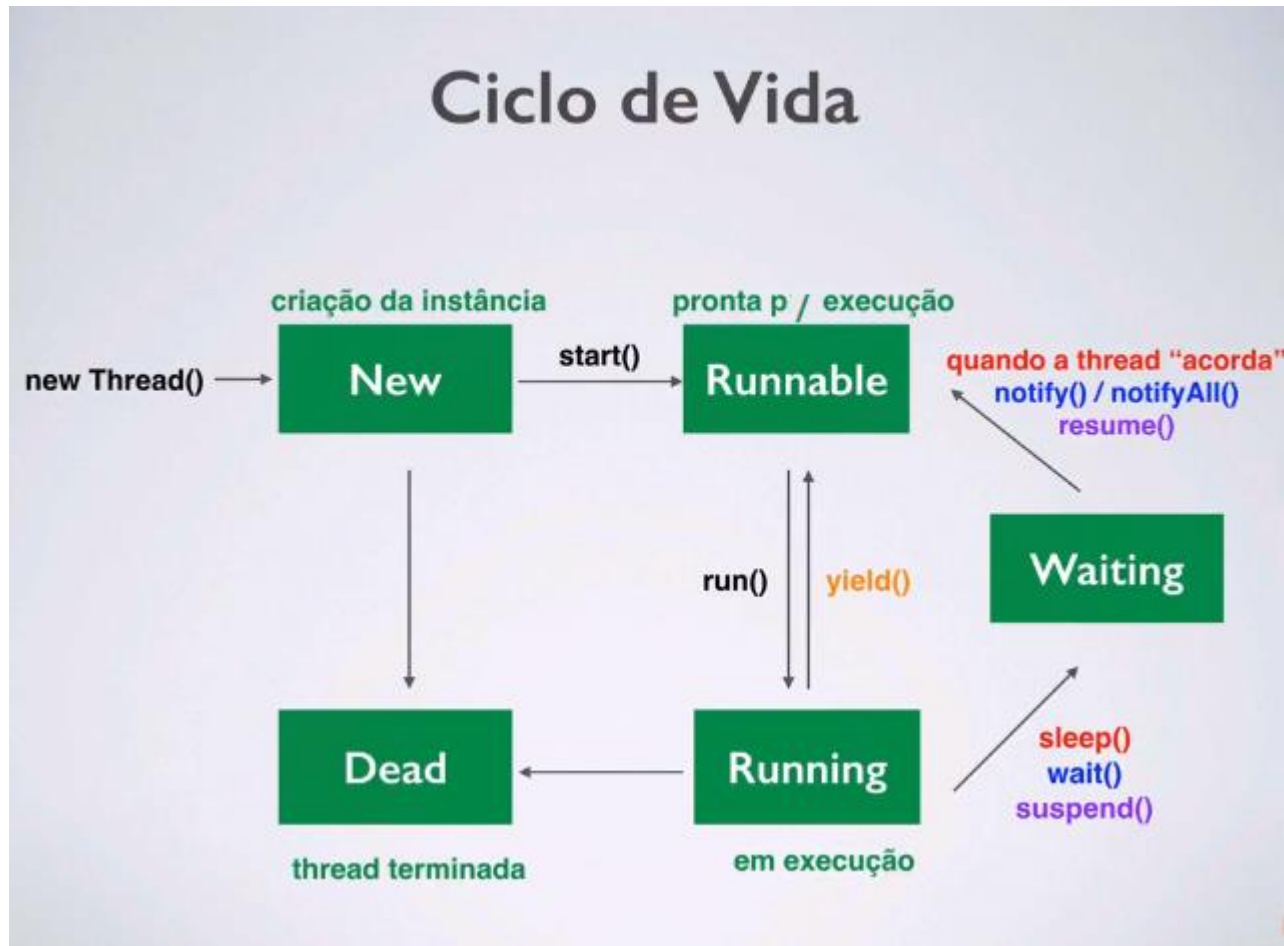


Diagrama de estados representando o ciclo de vida de uma thread

Fonte: Java How to Program, 9th Edition



Threads





Threads – interface Runnable

- ▶ O interface Runnable está definido na package java.lang;
- ▶ Um objeto *Runnable* representa uma tarefa que pode ser executada em paralelo com outras tarefas;
- ▶ O interface declara um único método, run;
- ▶ O método run contém o código que define a tarefa que o objeto Runnable deve executar;
- ▶ Quando uma thread passa do estado new para o estado runnable o método run é executado;



Threads – método Sleep

- ▶ O método sleep suspende a execução da thread atual por um determinado período;
- ▶ Permite tornar o processador acessível para outras threads ou aplicações;
- ▶ Existem duas versões do método sleep, a diferença entre as versões reside na unidade de tempo, uma versão recebe o valor considerando que a unidade é o milisegundo, a outra versão em nanosegundo.



Threads – sincronização

- ▶ Quando um objeto é partilhado entre threads e é alterado por uma ou mais, os resultados podem ser indeterminados.
- ▶ O problema pode ser solucionado dando acesso exclusivo ao objeto a uma thread.
- ▶ O processo designa-se por **thread synchronization**.



Threads

Números e Letras

1. Criar uma classe chamada: *ThreadNumerosLetras* que implemente a interface *Runnable* e com um atributo chamado *tipo* do tipo inteiro.
 - tipo = 1, mostra os números de 1 a 30
 - tipo = 2, mostra as letras de 'a' a la 'z'



Threads

Contadores

2. Criar uma classe chamada *Contador* que tenha:

- um atributo que seja um contador,
- outro que seja o nome da thread e
- outro que seja o limite do contador, isto é, onde deve acabar.
- Criar vários contadores e executálos.



Threads

3.



Threads

Corrida de Cavalos

4. Fazer um *JFrameForm* com uma *progressbar*.

- criar 4 threads que representam cavalos (que têm um nome e um caminho a percorrer ou percentagem percorrido).
- Quando clicarmos no botão iniciar, a corrida começará a avançar, atualizando-se a *progress bar* com um numero aleatorio entre 1 e 15.
- Ganha o primeiro que chegar aos 100.