

Java Multithreading

A linguagem de programação Java é amplamente utilizada e tem sido instrumental na formação do mundo do desenvolvimento de software. Uma de suas características mais impressionantes é a multithreading, múltiplas trilhas, que permite aos desenvolvedores criar várias execuções de tarefas dentro de um único programa.

Para criar Threads em Java basta implementar a Interface Runnable ou estender a Classe Thread:

```
Thread t = new Thread(new Runnable(){
    @Override
    public void run() {
    }
});
```

Exemplo de extensão da classe Thread:

```
public class Tarefa extends Thread {

    private final long valorInicial;
    private final long valorFinal;
    private long total = 0;

    //método construtor que receberá os parâmetros da tarefa
    public Tarefa(int valorInicial, int valorFinal) {
        this.valorInicial = valorInicial;
        this.valorFinal = valorFinal;
    }

    //método que retorna o total calculado
    public long getTotal() {
        return total;
    }

    /*
    Este método se faz necessário para que possamos dar start() na Thread
    e iniciar a tarefa em paralelo
    */
    @Override
    public void run() {
        for (long i = valorInicial; i <= valorFinal; i++) {
            total += i;
        }
    }
}
```

Outro exemplo:

```
// Java code for thread creation by extending

// the Thread class

class MultithreadingDemo extends Thread {

    public void run()

    {

        try {
```

```

        // Displaying the thread that is running
        System.out.println(
            "Thread " + Thread.currentThread().getId()
            + " is running");
    }
    catch (Exception e) {
        // Throwing an exception
        System.out.println("Exception is caught");
    }
}
}

```

// Main Class

```

public class Multithread {
    public static void main(String[] args)
    {
        int n = 8; // Number of threads
        for (int i = 0; i < n; i++) {
            MultithreadingDemo object
                = new MultithreadingDemo();
            object.start();
        }
    }
}

```

0 mesmo exemplo, agora implementando a Interface Runnable:

#####

// Java code for thread creation by implementing

// the Runnable Interface

```

class MultithreadingDemo implements Runnable {

    public void run()

    {

        try {

            // Displaying the thread that is running

            System.out.println(

                "Thread " + Thread.currentThread().getId()

                + " is running");

        }

        catch (Exception e) {

            // Throwing an exception

            System.out.println("Exception is caught");

        }

    }

}

```

// Main Class

```

class Multithread {

    public static void main(String[] args)

    {

        int n = 8; // Number of threads

        for (int i = 0; i < n; i++) {

            Thread object

                = new Thread(new MultithreadingDemo());

            object.start();

        }

    }

}

```

Utilizando as expressões Lambda:

```
Runnable runnable = () -> System.out.println("Hello");
```

Para iniciar a execução, basta chamar o método start():

```
runnable.start();
```

As multi-threads são um conceito que permite ao programa executar múltiplas tarefas simultaneamente. Isso significa que um único programa pode realizar múltiplas tarefas ao mesmo tempo, como processamento de dados, envio de solicitações e recebimento de respostas. A linguagem Java fornece uma implementação excelente da multi-tarefas através da classe `Thread`, o que permite aos desenvolvedores criar tais threads, trilhas, com facilidade e eficiência.

Um dos mais importantes benefícios da multi-thread em Java é a melhorada responsividade. Em modelos de programação tradicionais, um programa pode apenas realizar uma tarefa de vez. Se um usuário deseja realizar múltiplas tarefas ao mesmo tempo, eles precisariam lançar programas separados para cada tarefa. Esta abordagem tem várias limitações, incluindo aumento da carga do sistema e desempenho geral reduzido.

Em contraste, a multi-thread em Java permite aos desenvolvedores criar um único programa que possa gerenciar as múltiplas tarefas simultaneamente. Isso garante que o usuário tenha uma experiência mais rápida e eficiente ao interagir com o sistema.

Além disso, multi-thread também pode melhorar a escalabilidade do aplicativo. Quando um programa precisa realizar múltiplas tarefas simultaneamente, permite que essas tarefas sejam executadas em paralelo, o que melhora significativamente a eficiência e a performance do sistema.

Multi-threads é uma característica fundamental da linguagem Java que pode melhorar a responsividade e escalabilidade dos aplicativos.

Referências:

<https://www.geeksforgeeks.org/multithreading-in-java/>

<https://www.devmedia.com.br/threads-paralelizando-tarefas-com-os-diferentes-recursos-do-java/34309>