

Mãos à obra: Imprimindo os elementos

Nesse exercício vamos usar novamente a nossa lista caseira. O primeiro passo é voltar a utilizar a implementação da nossa lista, pois queremos alterá-la.

1) No projeto `experimento`, Na classe `TarefaAdicionarElemento` volte a usar a `Lista` em vez da `java.util.List`:

```
public class TarefaAdicionarElemento implements Runnable {

    private Lista lista;
    private int numeroDaThread;

    public TarefaAdicionarElemento(Lista lista, int numero) {
        this.lista = lista;
        this.numeroDaThread = numero;
    }

    @Override
    public void run() {
        for (int i = 0; i < 100; i++) {
            lista.adicionaElementos("Thread " + numeroDaThread + " - " + i );
        }
    }
}
```

2) Agora crie uma nova thread para imprimir os elementos da lista:

```
public class Principal {

    public static void main(String[] args) throws InterruptedException {

        Lista lista = new Lista();

        for (int i = 0; i < 10; i++) {
            new Thread(new TarefaAdicionarElemento(lista, i)).start();
        }

        new Thread(new TarefaImprimir(lista)).start();
    }
}
```

3) Também crie a classe da tarefa:

```
public class TarefaImprimir implements Runnable {

    private Lista lista;

    public TarefaImprimir(Lista lista) {
        this.lista = lista;
    }
}
```

```
@Override
public void run() {
    for (int i = 0; i < lista.tamanho(); i++) {
        System.out.println(i + " - " + lista.pegaElemento(i));
    }
}
```

4) Abra a classe `Lista` e adicione um `Thread.sleep(10)` no método `adicionaElementos`:

```
public class Lista {

    //atributos omitidos

    public synchronized void adicionaElementos(String elemento) {
        this.elementos[indice] = elemento;
        this.indice++;

        try{
            Thread.sleep(10);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }

    //outros métodos omitidos
}
```

5) Rode a classe `Principal`. Tudo está ok?