

Para saber mais: Obtendo locks de forma explícita

Nesse capítulo vimos o poder do bloco `synchronized` e como ele nos ajuda a trabalhar com tarefas atômicas. Como alternativa, podemos também conseguir esse bloqueio de forma *explícita* (programaticamente) através de uma classe chamada `ReentrantLock`.

E como podemos utilizar essa classe em nosso banheiro?

```
public class Banheiro {

    private Lock lock = new ReentrantLock();

    public void fazNumero1() {

        lock.lock();
        System.out.println("entrando no banheiro");
        System.out.println("fazendo coisa rapida");

        try {
            Thread.sleep(8000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

        System.out.println("dando descarga");
        System.out.println("lavando a mao");
        System.out.println("saindo do banheiro");
        lock.unlock();
    }

    public void fazNumero2() {

        lock.lock();
        System.out.println("entrando no banheiro");
        System.out.println("fazendo coisa demorada");

        try {
            Thread.sleep(15000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

        System.out.println("dando descarga");
        System.out.println("lavando a mao");
        System.out.println("saindo do banheiro");
        lock.unlock();
    }
}
```

Pesquise um pouco mais sobre essa classe. Você consegue enxergar alguma diferença de utilizar o bloco `synchronized` ?

