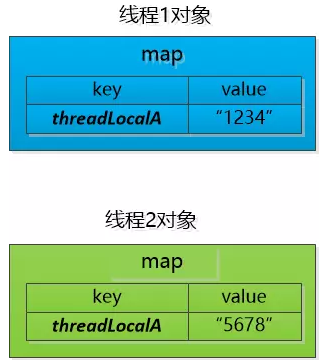
ThreadLocal<String> threadLocalA= new ThreadLocal<String>();

线程1： threadLocalA.set("1234");

线程2： threadLocalA.set("5678");

像‘1234’， ‘5678’这些值都会放到自己所属的线程对象中。”



“等你使用的时候，可以这么办：”

线程1： threadLocalA.get() --> "1234"

线程2： threadLocalA.get() --> "5678"

“明白了，相当于把各自的数据放入到了各自Thread这个对象中去了，每个线程的值自然就区分开了。 可是我不明白的是为什么那个数据结构是个map 呢？”

“你想想，假设你创建了另外一个threadLocalB：”

ThreadLocal<Integer> threadLocalB = new ThreadLocal<Integer>();

线程1： threadLocalB.set(30);

线程2： threadLocalB.set(40);

那线程对象的Map就起到作用了：

