ThreadLocal：线程局部变量，是一种多线程间并发访问变量的解决方案，与其synchronized等加锁的方式不同，ThreaddLocal完全不提供锁，而使用以空间换时间的手段，为每个线程提供变量的独立副本，以保证线程安全。

从性能上说，ThreadLocal不具有绝对的优势，在并发不是很高的时候，加锁的性能会更好，但是作为一套与锁完全无关的线程安全解决方案，在高并发或者竞争激烈的场景下，使用ThreadLocal可以在一定程度上减少锁竞争。

示例代码：

package com.bjsxt.base.conn010;

public class ConnThreadLocal {

public static ThreadLocal<String> th = new ThreadLocal<String>();

public void setTh(String value){

th.set(value);

}

public void getTh(){

System.out.println(Thread.currentThread().getName() + ":" + this.th.get());

}

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

final ConnThreadLocal ct = new ConnThreadLocal();

Thread t1 = new Thread(new Runnable() {

@Override

public void run() {

ct.setTh("张三");

ct.getTh();

}

}, "t1");

Thread t2 = new Thread(new Runnable() {

@Override

public void run() {

try {

Thread.sleep(1000);

ct.setTh("李四");

ct.getTh();

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}, "t2");

t1.start();

t2.start();

}

}

Result：t1：张三

t2：null