

编程入门基础

—— 线程互斥

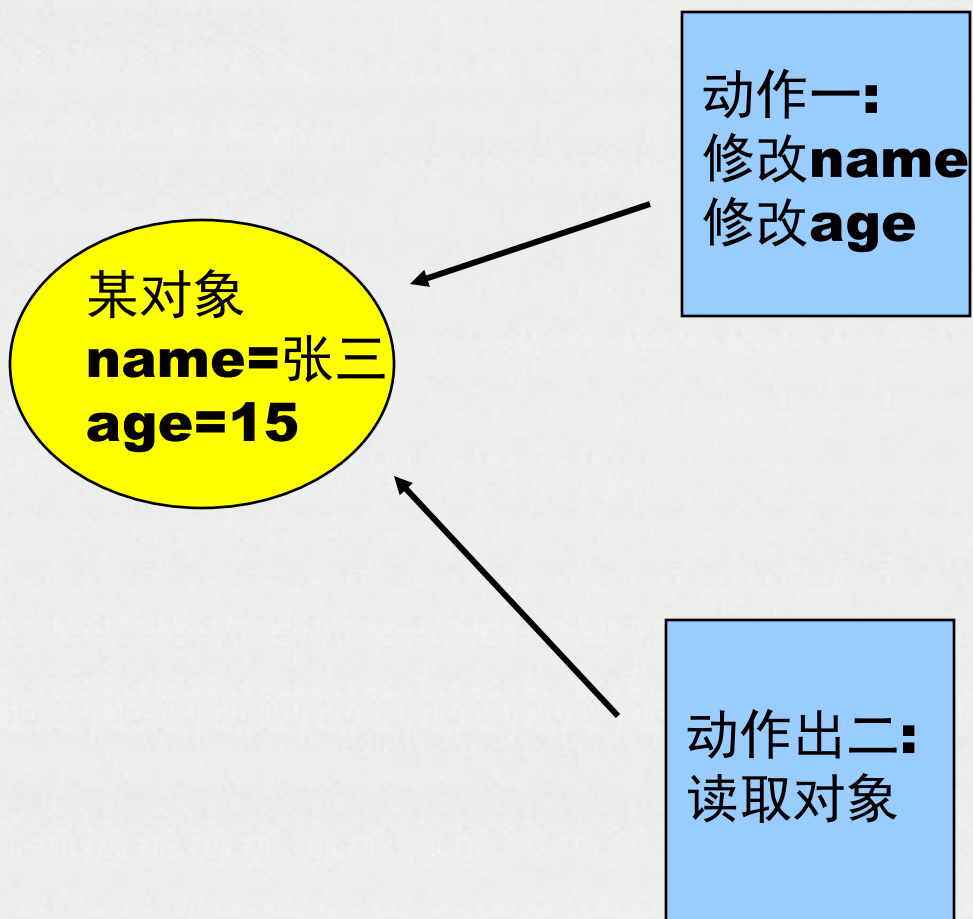
主讲教师：耿宇航



需要排斥的原因

- 线程与进程比，优点是共享内存，节约资源。
- 优点即是缺点
- 共享内存带来的问题：访问冲突！
- 如何防止同时操作同一对象

冲突示例



解决方法

- 操作的时候加锁
- 操作结束解锁
- 撞到锁的一方等待
- 解锁时唤醒等待一方

线程的排斥

- Object的继承类都可充当“锁”角色
- `synchronized(Object){ 需要排斥的操作; }`
- 选择一个公共的“锁”
 - 可以用this指针做排斥“锁”
 - 可以传入公共对象做排斥“锁”
 - 可以使用静态成员充当排斥“锁”
- 线程安全对象
 - this指针排斥是制作“线程安全”对象常用手法

synchronized(obj)动作

- { 的动作
 - 如果obj无人占有，则标为占用
 - 如果obj已被占用，则本线程在obj的等待队列中睡觉
- } 的动作
 - 如果obj脚下无人睡觉，则标记obj空闲
 - 如果obj脚下有人睡，则唤醒该线程

谢 谢 ！