

MultiThreading -> Last Minor Pillar of OOP

OOP

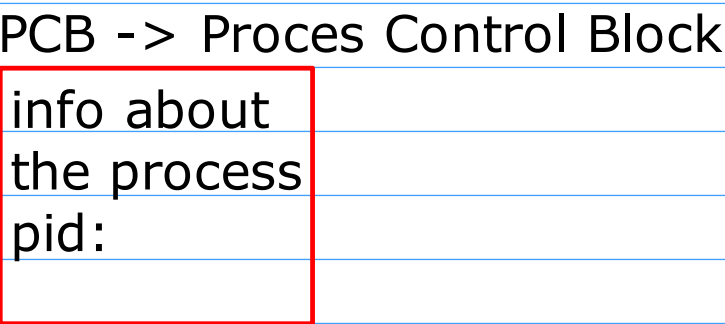
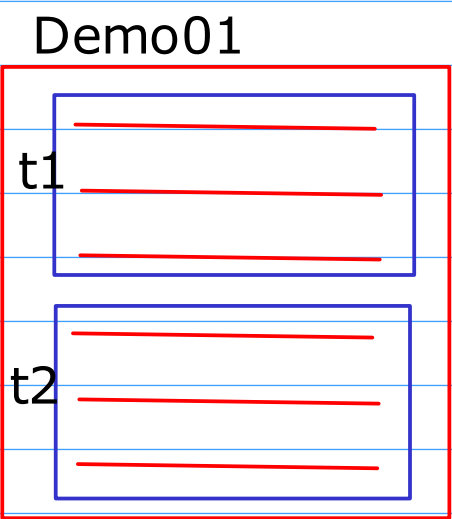
Major

Abstraction, Encapsulation
modularity, hirerachy

Minor

Polymorphism, Persistance
Concurrency

Process
- Program in Execution

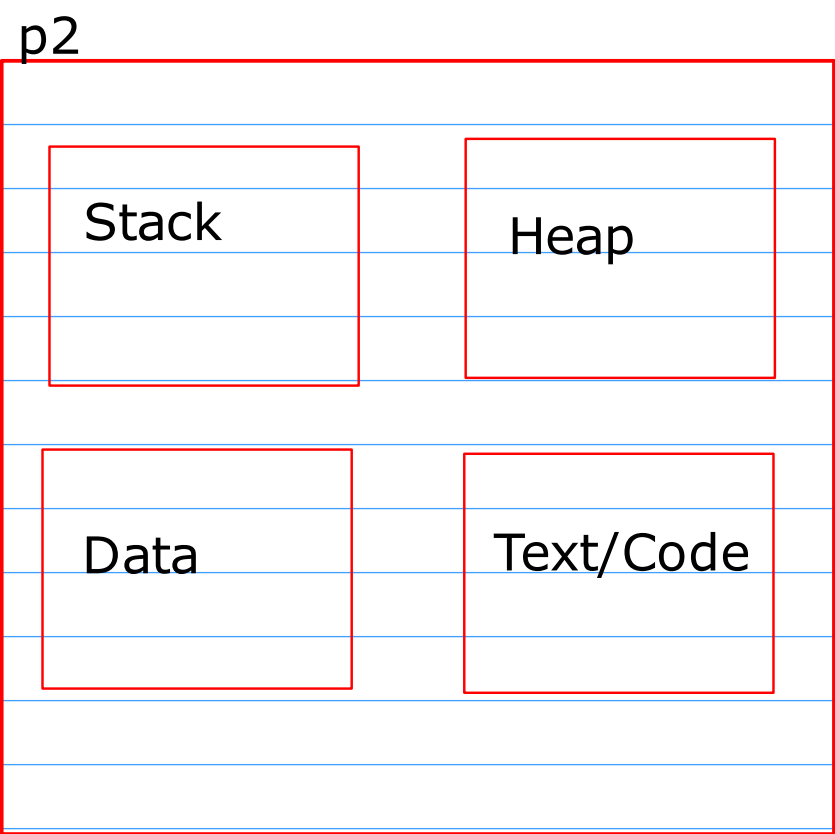
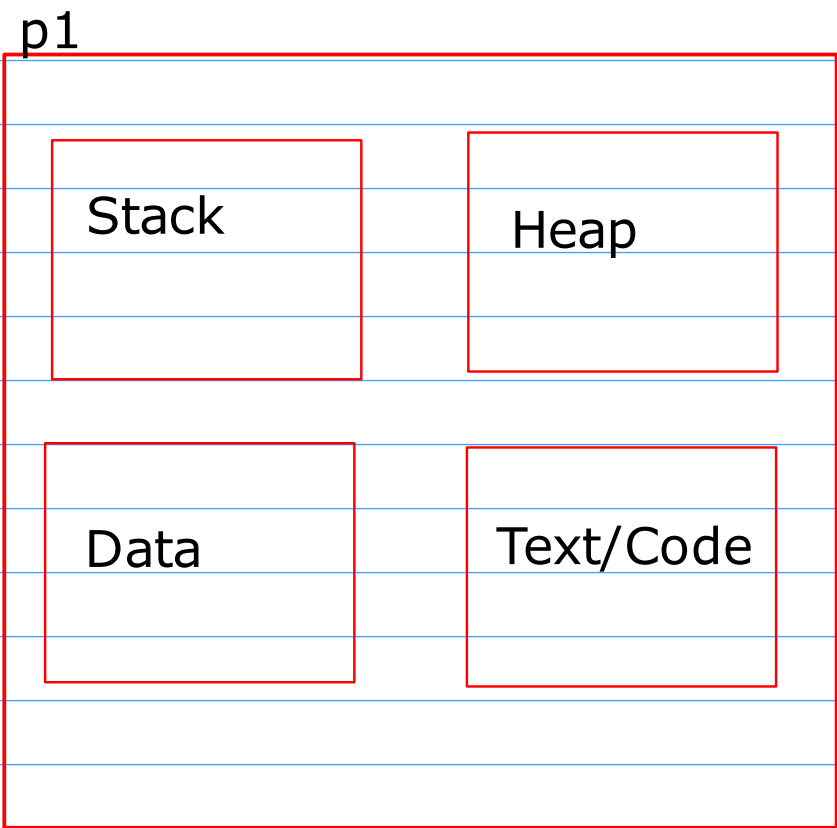


Program
JVM
OS

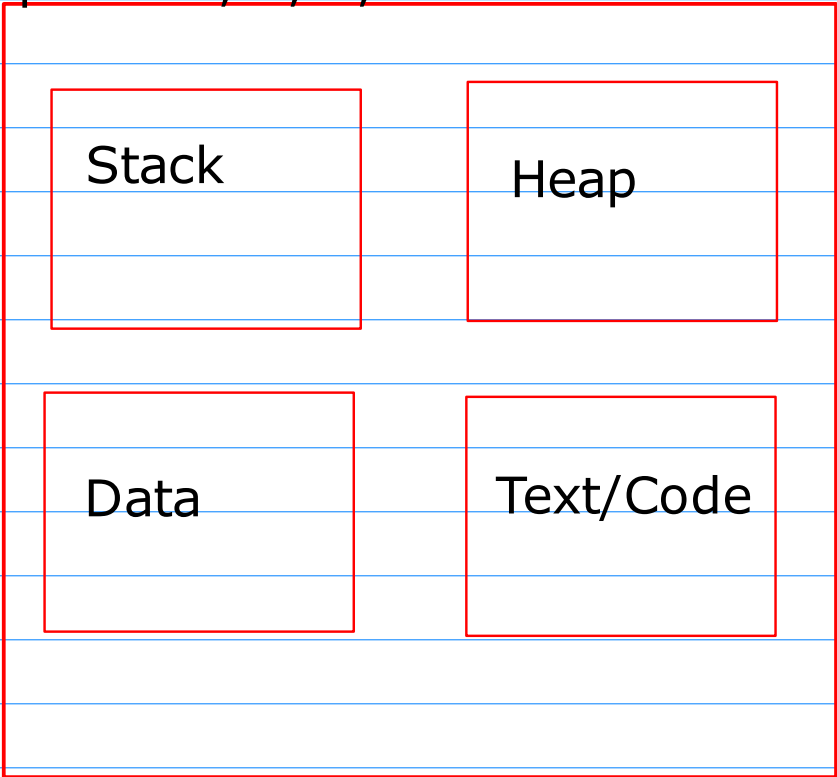
Red arrows point from 'Program' to 'JVM' and from 'JVM' to 'OS'.

Thread
- It is a light weight process

Text/Code
Stack
Heap
Data



p1-> t1,t2,t3,t4



a.exe
java Program01

java Program01
- main
- GC

Process and Threads are managed by OS

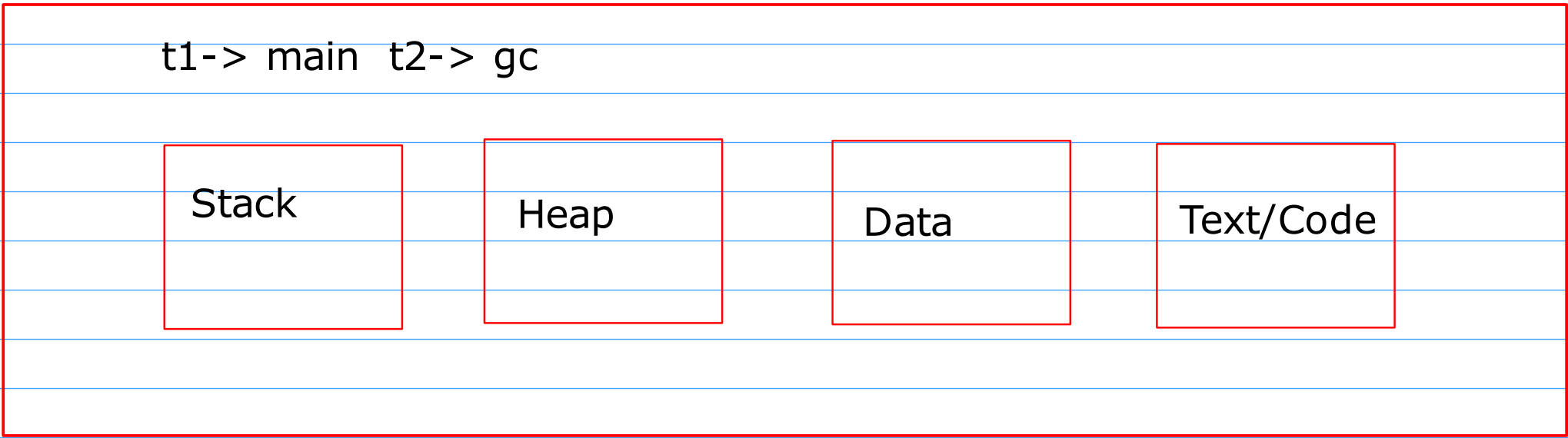
Thread Creation in java

- use a Thread class
- use a Runnable interface

clas extends anotherclass implements Runnable
class extends Thread

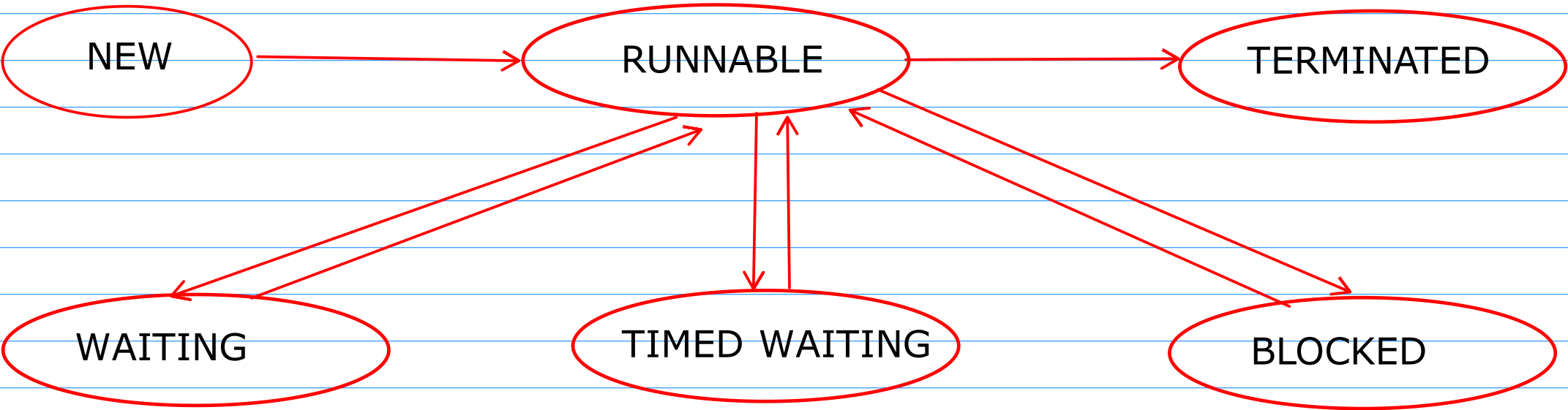
java Program
jvm -> execution on os
- t1(main thread)
Program.main();
- t3
- t4
- t2(gc thread)

JVM-> Process->



OS-> manage the process and Threads
Process -> PCB
Thread -> TCB

Thread LifeCycle

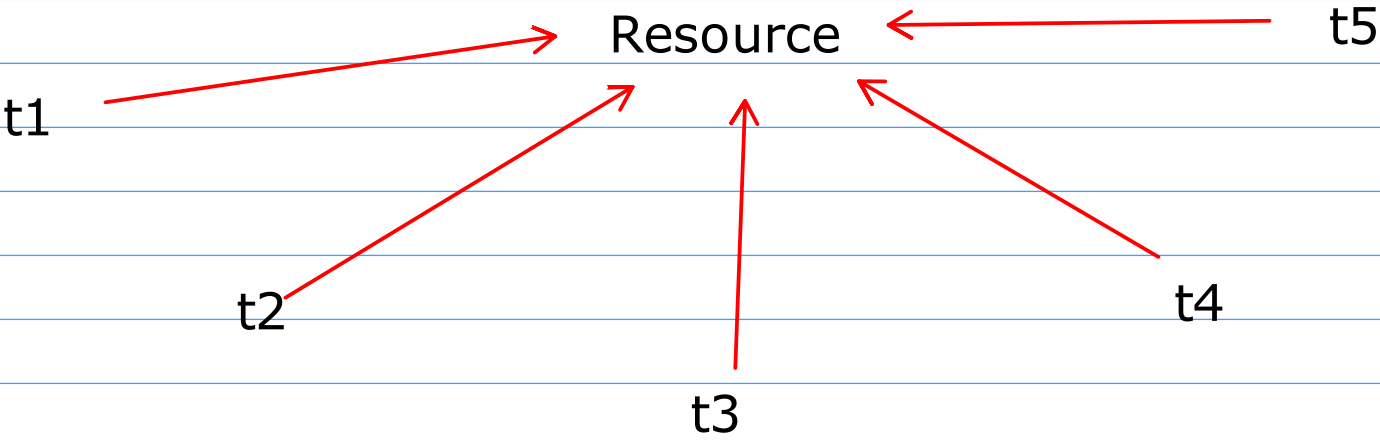


OS Multi-Tasking

OS(Concepts) -> before learning any language

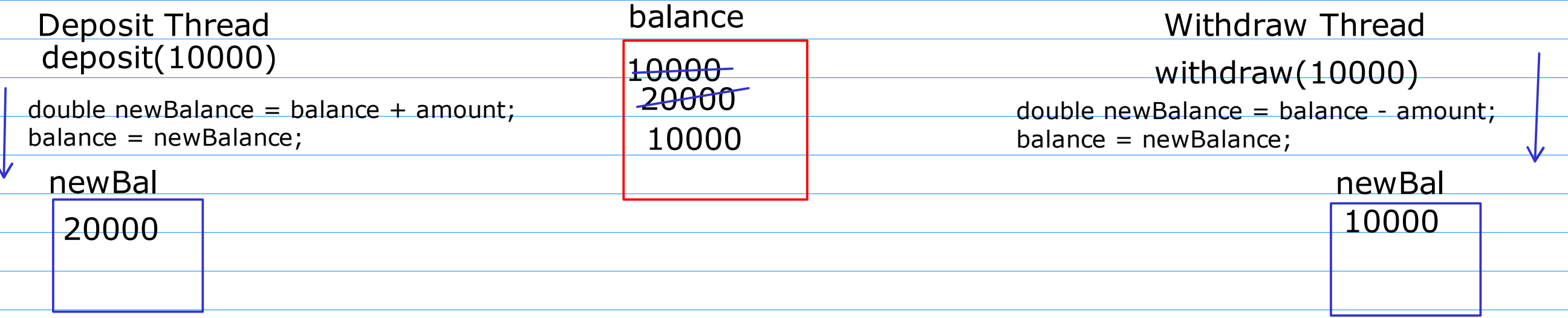
OS + Java

Process -> new Process()
OS -> exec(), fork()

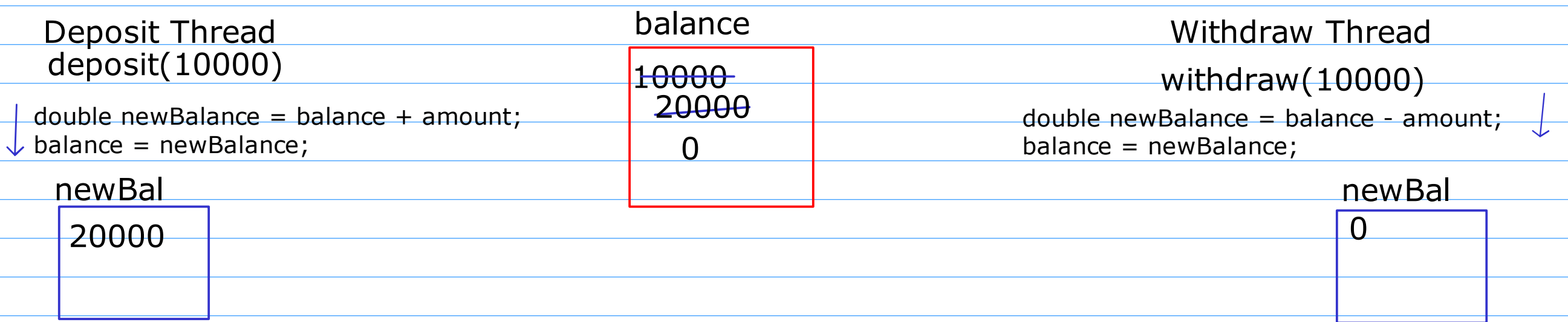


Race Condition
Synchronization

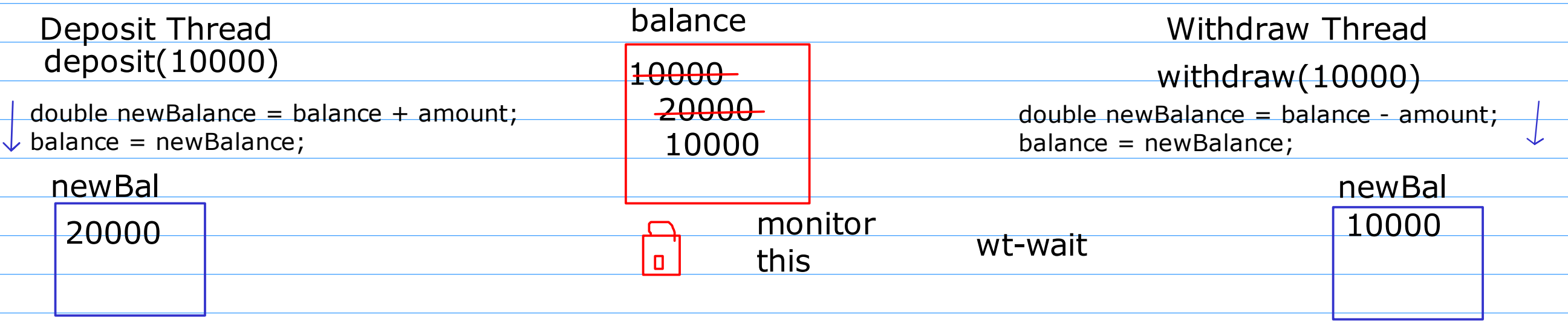
Ideal Situation



Race Condition



Synchronization



Thread safe, Not Thread Safe