

PRAXIS 54

MULTITHREADING

```

    public void processCommand(byte b)
    {
        //Move the robot based on the command.
    }

    public void run()
    {
        byte[] cmds;
        while(true)
        {
            synchronized(controller) { //1
                if (commands == null) //2
                {
                    try {
                        controller.wait(); //3
                    }
                    catch(InterruptedException e){} //Exception is ignored
                                                //purposefully.
                }
                cmds = new byte[commands.length];
                for (int i=0; i<commands.length; i++)
                    cmds[i] = commands[i];
                commands = null;
            }
            //Now we have commands for the robot.
            int size = cmds.length;
            for (int i=0; i<size; i++)
                processCommand(cmds[i]); //Move the robot.
        }
    }
}

class RobotController extends Thread
{
    private Robot robot1;
    private Robot robot2;
    public static void main(String args[])
    {
        RobotController rc = new RobotController();
        rc.start();
    }

    public void run()
    {
        robot1 = new Robot(this);
        robot1.start();
        robot2 = new Robot(this);
        robot2.start();
    }
    public synchronized void loadCommands(byte[] b) //4
    {
        Robot.storeCommands(b); //Give the commands to the Robot.
    }
}

```