

## CountDown

### *Make CountDown Runnable:*

```
public class CountDown extends Applet
    implements Runnable
{
    ...
```

### *Start a new thread in the Applet's start() method:*

```
public void start()
{
    new Thread(this).start();
}
```

## *Add a run method that ticks*

```
public void run() {
    i_ = N;
    display_.setValue( i_ );
    while (notFinished()) {
        tick();
    }
    beep();
}
private boolean notFinished() {
    return i_ > 0;
}
private void tick() {
    i_--; ...
}
```

## RotatorPanel

### *Make RotatorPanel Runnable*

```
public class RotatorPanel extends Panel
    implements Runnable
{
```

### *Add an isRunning flag*

```
    boolean isRunning_ = false;
```

### *Start a new thread in the start method*

```
public void start()
{
    setZero();
    new Thread(this).start();
}
```

## *Add run() method that rotates while isRunning()*

```
public void run() {
    while(isRunning()) {
        arcPanel_.rotate();
        try { Thread.currentThread().sleep(100); }
        catch(InterruptedException e) { }
    }
}

public synchronized boolean isRunning() {
    while (!isRunning_) {
        try { wait(); }
        catch(InterruptedException e){ return false; }
    }
    return true;
}
```

## *Synchronize `isRunning()`, `setActive()` and `setInactive()`*

```
public synchronized void setActive()
{
    isRunning_ = true;
    run_.setEnabled( false );
    pause_.setEnabled( true );
    arcPanel_.setBackground( RUNCOLOR );
    notifyAll();
}
public synchronized void setInactive()
{
    isRunning_ = false;
    ...
    // notifyAll(); // Not necessary
}
```

## Garden

*Synchronize increment() in SafeCounter subclass of Counter*

```
public class SafeCounter extends Counter
{
    public SafeCounter( String title )
    { super( title ); }

    public SafeCounter( String title, Color bgc )
    { super( title, bgc ); }

    public synchronized void increment()
    { super.increment(); }
}
```

## *Change Garden.java to use SafeCounter*

```
public class Garden extends Applet
    implements ActionListener
{
    public void init()
    {
        ...
        Panel p1 = new Panel( new GridLayout(1,3) );
        total_ = new Counter SafeCounter( "Counter" );
        ...
    }
}
```

## Dining Philosophers

*Modify Philosopher to know whether he is even or odd*

```
public final class Philosopher
{ ...
  public boolean isEven() {
    return (id_ % 2) == 0;
  }
  ...
}
```

*Modify RunnablePhilosopher to pick up right fork if even, else left fork*

```
public class RunnablePhilosopher extends Thread {  
    public void run() { ...  
        phil_.setHungry();  
        if (phil_.isEven()) {  
            phil_.pickRight();  
            pause();  
            phil_.pickLeft();  
        } else {  
            phil_.pickLeft();  
            pause();  
            phil_.pickRight();  
        }  
        ... }  
}
```