

```
kapasite = 5; // halattaki maksimum maymun sayısı
halatMaymun = 0; // aktif olarak halatta bulunan maymun
sayısı
Yön = 0; // batıya Gitme flag'i açık başlangıçta
Semaphore yönSema = new Semaphore(1);
Semaphore binarySemaphore = new Semaphore (1);
Semaphore doğu = new Semaphore (0);
Semaphore batı = new Semaphore (1); // başlangıçta batıya
gitme açık
Semaphore halatSema = new Semaphore (kapasite);
muzIsteyen=0;
uykuIsteyen =0;
```

```
psvm(String[] args)
{

    Thread[] maymunT=new Threads[100]
    Yön=1;
    for (int i=0;i<maymunT.length; i++)
        maymunT=new Maymun(kapasite,Yöne,yönSema,
binarySema .....);
        maymunT.start();
```

```
kapasite = 5; // halattaki maksimum maymun sayısı
halatMaymun = 0; // aktif olarak halatta bulunan maymun
sayısı
Yön = 0; // batıya Gitme flag'i açık başlangıçta
Semaphore yönSema = new Semaphore(1);
Semaphore binarySemaphore = new Semaphore (1);
Semaphore doğu = new Semaphore (0);
Semaphore batı = new Semaphore (1); // başlangıçta batıya
gitme açık
Semaphore halatSema = new Semaphore (kapasite);
muzIsteyen=0;
uykuIsteyen =0;
```

```
psvm(String[] args)
{

    Thread[] maymunT=new Threads[100]
    Yön=1;
    for (int i=0;i<maymunT.length; i++)
        maymunT=new Maymun(kapasite,Yöne,yönSema,
binarySema .....);
        maymunT.start();
```

}

```
Public class Maymun extends Thread{
```

```
// instance variablelar (driver Sınıfındaki aynısı)
```

```
// constructor instance variableların aynısı
```

```
public void batiyaGit(){
```

```
    yönSema.wait()
```

```
    if Yön ==1 // Yön doğu ise
```

```
        binarySemaphore.wait();
```

```
        if halatMaymun == 0
```

```
            Yön=0;
```

```
            yönSema.signal();
```

```
            halatMaymun ++;
```

```
            binarySemaphore.signal();
```

```
            muzYe();
```

```
        else halatMaymun>0
```

```
            muzIsteyen ++;
```

```
            yönSema.signal()
```

```
            if muzIsteyen == 1
```

```
                batı.wait();
```

```
                binarySemaphore.signal();
```

```
    else // zaten batıda ise
```

```
    halatSema.wait();  
    muzIsteyen++;  
    binarySemaphore.wait();  
    halatMaymun++;  
    binarySemaphore.signal();  
    muzye();
```

```
    yönSema.signal()
```

```
}
```

```
Public void muzye(){
```

```
    Random rObj = new Random();  
        Thread.Sleep (rObj.NextInt(3000));  
    binarySemaphore.wait();  
        halatMaymun--;  
        muzIsteyen--;  
        if muzIsteyen ==0 && uykuIsteyen>0
```

```
            dogu.signal();
```

```
        end
```

```
    binarySemaphore.signal();
```

```
    sout("Muz yiyor");  
    Thread.Sleep (rObj.NextInt(3000));  
        sout("Uyumaya gitmak için
```

```
hazırlanıyor”);
```

```
}
```

```
public void doguyaGit(){
    yönSema.wait()
    if Yön ==0 // Yön batı ise
        binarySemaphore.wait();
        if halatMaymun == 0
            Yön=1;
            yönSema.signal();
            halatMaymun ++;
            binarySemaphore.signal();
            uyu();

        else halatMaymun>0
            uykuIsteyen ++;
            binarySemaphore.signal();
            yönSema.signal()
            if uykuIsteyen==1
                dogu.wait();

    else // zaten doğuda ise

        uykuIsteyen++;
        halatSema.wait();
        binarySemaphore.wait();
        halatMaymun++;
        binarySemaphore.signal();
        uyu();
```

```
}
```

```
Public void uyu(){
```

```
    Random rObj = new Random();  
    Thread.Sleep (rObj.NextInt(3000));
```

```
    binarySemaphore.wait();  
    halatMaymun--;  
    uykuIsteyen--;  
    binarySemaphore.signal();  
    sout("Uyuyor");
```

```
    binarySemaphore.wait();  
    if uykuIsteyen ==0 && muzIsteyen>0  
        Yön=0;  
        bati.signal();  
    end  
    binarySemaphore.signal();
```

```
        Thread.Sleep (rObj.NextInt(3000));  
        sout("Muz yemek için hazırlanıyor");
```

```
}
```

```
Public void run()
```

```
{
```

```
    whileTrue(){  
        batiyaGit();  
        doguyaGit();  
    }
```

```
}
```

```
}
```