

JAVA İŞ PARÇACIĞI

Thread sınıfı; İş parçacığını başlatan sınıf.

```
...avc Ben.java x JavaApplication4.java x JavaApplication5.java x
Source History
1  /*
2  * To change this license header, choose License Headers in Project Properties
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package javaapplication5;
7
8  class İsci extends Thread{
9      public void run(){
10         for(int i=0; i<5 ; i++){
11             System.out.println(this.getName()+"-->" +i);
12         }
13     }
14 }
15 public class JavaApplication5 {
16
17     public static void main(String[] args) {
18         // TODO code application logic here
19         İsci i1=new İsci();
20         İsci i2=new İsci();
21
22         i1.start();
23         i2.start();
24     }
25 }
26 }
27 }
```

İşlemcinin boş zamanına göre işlemleri yapar.

Her iş parçacığının öncelik sırası vardır. Öncelik sırası ilerleyen rütbelerde daha ayrıntılı alınır. Tek CPU'ya sahip bilgisayarda aynı anda bir parçacığı çalıştırır. Bu işlemleri yapan JSM (Java Sanal Makinesi) 'dir.

```
Start Page x JavaApplication6.java x Robot.java x
Source History
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package javaapplication6;
7
8  public class Robot extends Thread{
9      private int donguSayisi;
10     public Robot (String isim,int donguSayisi) {
11         super (isim);
12         this.donguSayisi=donguSayisi;
13     }
14     public void run() {
15         try{
16             if(donguSayisi==0){
17                 return;
18             }
19             for(int i=0; i<donguSayisi; i++){
20                 System.out.println(this.getName()+"-->" +i);
21             }
22         }
23         catch(Exception e){
24             System.err.println("Hata!");
25         }
26     }
27 }
28
```

```
Source History
1
2  package javaapplication6;
3
4  public class JavaApplication6 {
5
6      public static void main(String[] args) {
7          Robot a=new Robot ("A", 3);
8          Robot b=new Robot ("B", 2);
9          Robot c=new Robot ("C", 0);
10         Robot d=new Robot ("D", 4);
11
12         a.start();
13         b.start();
14         c.start();
15         d.start();
16     }
17 }
18
19
```

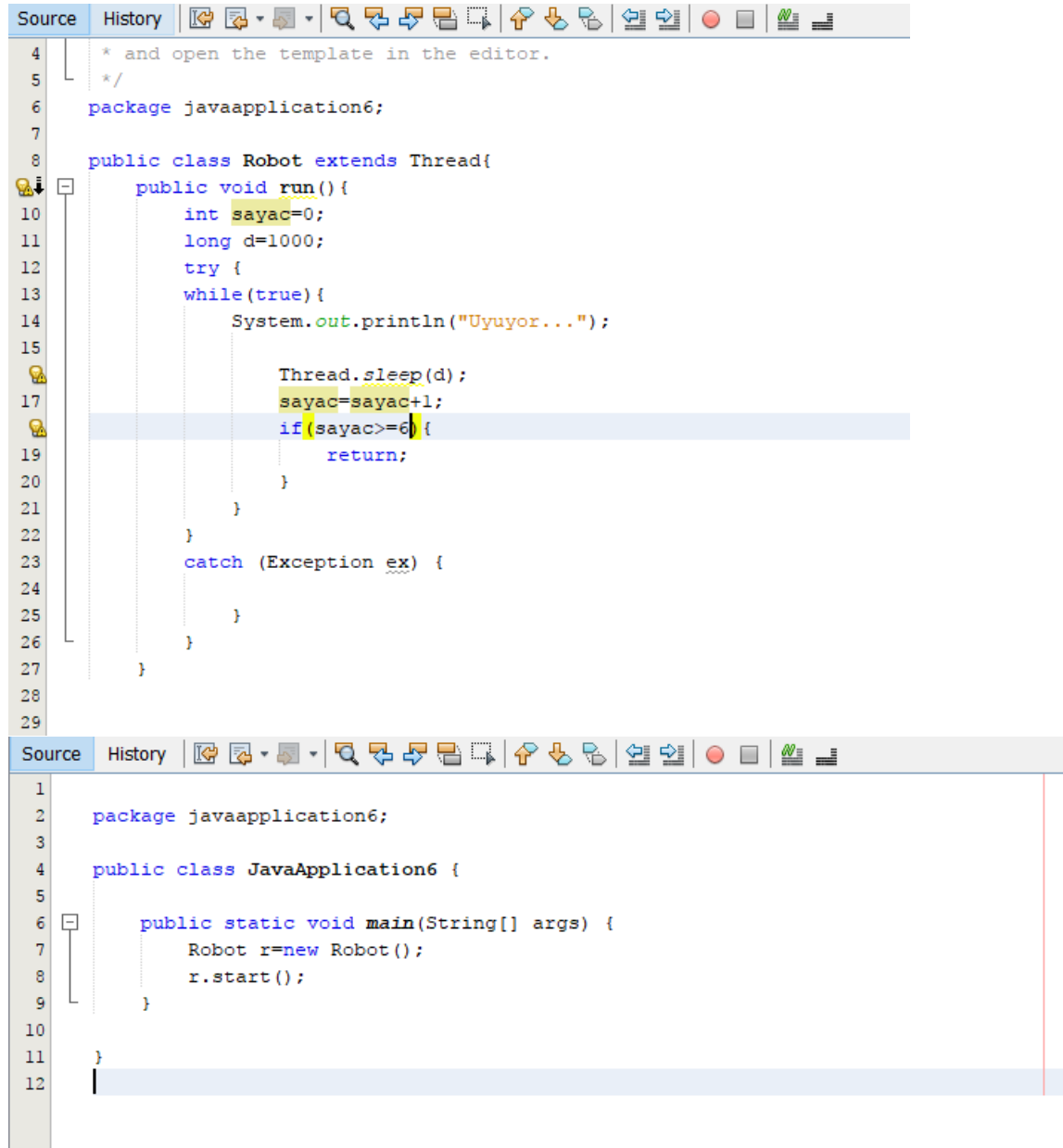
// ÖNEMLİ ÖRNEK

Start(); Thread'leri başlat.

Stop(); Thread'leri durdur.

Sleep(); Thread'leri belli zaman aralığında uyut.

//Aşağıdaki program ekrana 6 saniye boyunca 1 saniye aralıklarla uyuyor yazar.



```
4 | * and open the template in the editor.
5 | */
6 | package javaapplication6;
7 |
8 | public class Robot extends Thread{
9 |     public void run(){
10 |         int sayac=0;
11 |         long d=1000;
12 |         try {
13 |             while(true){
14 |                 System.out.println("Uyuyor...");
15 |
16 |                 Thread.sleep(d);
17 |                 sayac=sayac+1;
18 |                 if(sayac>=6){
19 |                     return;
20 |                 }
21 |             }
22 |         }
23 |         catch (Exception ex) {
24 |
25 |         }
26 |     }
27 | }
28 |
29 |
```

```
1 |
2 | package javaapplication6;
3 |
4 | public class JavaApplication6 {
5 |
6 |     public static void main(String[] args) {
7 |         Robot r=new Robot();
8 |         r.start();
9 |     }
10 |
11 | }
12 |
```

Thread' in aktif olup olmadığını gösteren yol isAlive();

