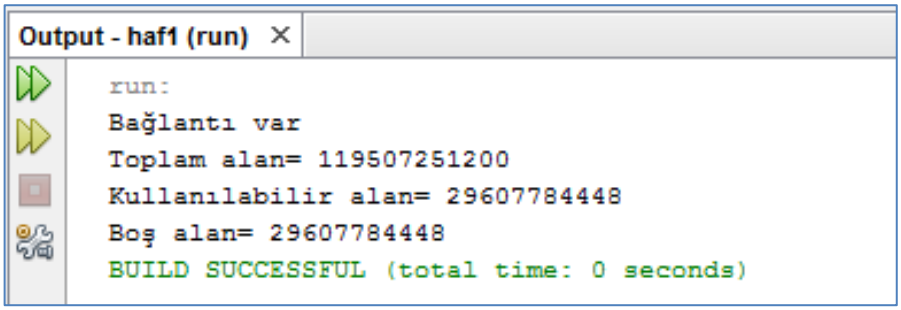
1. Hafta uygulamaları

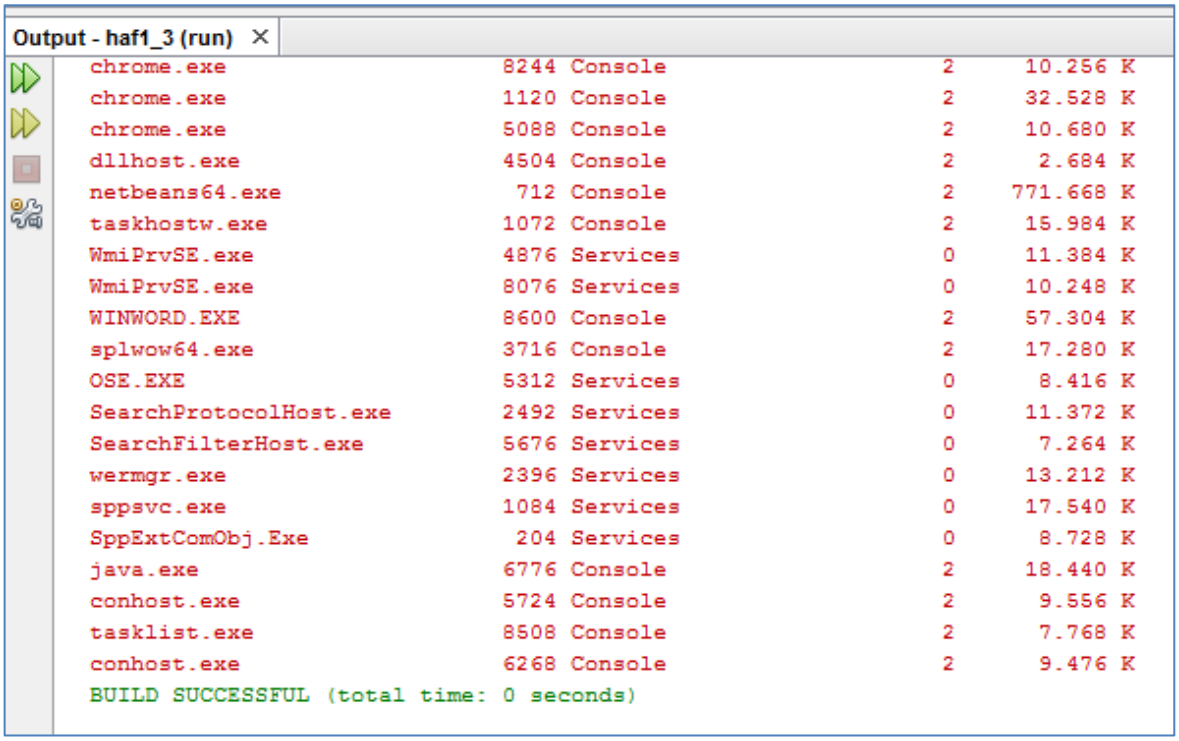
```
package haf1;
import java.io.File;
import java.io.IOException;
import java.net.URL;
import java.net.URLConnection;

public class Haf1 {

   public static void main(String[] args) {

      bag();
      disk();
   }

   public static void bag(){
      boolean connectivity;
      try{
         URL url = new URL("http://www.cs.sakarya.edu.tr");
         URLConnection conn =url.openConnection();
         conn.connect();
         connectivity = true;
      }
      catch(Exception ex){
         connectivity = false;
      }
      if(connectivity){
         System.out.println("Bağlantı var");
      }
      else{
         System.out.println("Bağlantı yok");
      }
   }
   public static void disk(){
      File f= new File("c:");
      long toplam_alan=f.getTotalSpace();
      long kullanilabilir_alan=f.getUsableSpace();
      long bos_alan=f.getFreeSpace();
      System.out.println("Toplam alan= " + toplam_alan);
      System.out.println("Kullanılabilir alan= " + kullanilabilir_alan);
      System.out.println("Boş alan= " + bos_alan);
   }
}
```



```
package haf1_2;

public class Haf1_2 {

   public static void main(String[] args) {

      uygulama();
   }
   public static void uygulama(){
      try{
         Process p = Runtime.getRuntime().exec("notepad");

      }
      catch(Exception ex){

      }

   }
}
```

Sonuç:
Çalıştırıldığında “Not Defteri “ programını açar.

```
package haf1_3;

import java.io.BufferedReader;
import java.io.InputStreamReader;

public class Haf1_3 {

   public static void main(String[] args) {

      uygulama();
   }

   public static void uygulama(){
      try{
         String line;
         Process p = Runtime.getRuntime().exec("tasklist");
         BufferedReader input = new BufferedReader(new InputStreamReader(p.getInputStream()));

         while((line=input.readLine()) != null){
            System.err.println(line);
         }
      }
      catch(Exception ex){

      }

   }
}
```



2. Hafta uygulamaları

```
package haf2_2;

public class Haf2_2 {

   public static void main(String[] args) {

      uygula();
   }
   public static void uygula(){

      try{
         //ProcessBuilder pb = new ProcessBuilder();
         //pb.command("notepad");
         ProcessBuilder pb = new ProcessBuilder("mspaint");
         pb.start();

      }
      catch(Exception ex){
         //return;
      }
   }

}
```

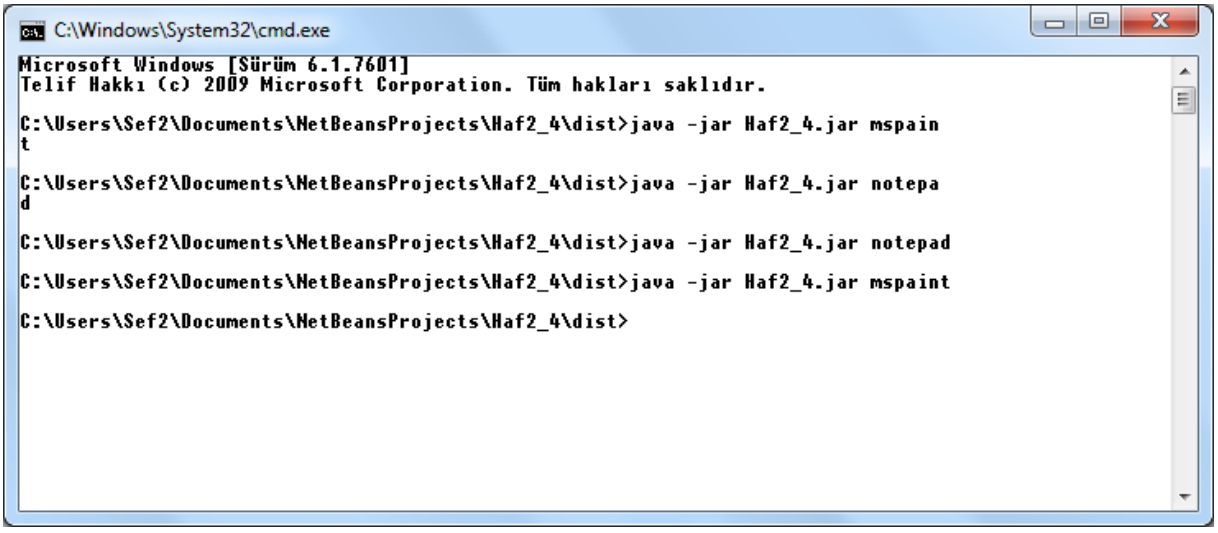
Sonuç: Çalıştırıldığında “Paint” programını açar

```
package haf2_3;

public class Haf2_3 {

   public static void main(String[] args) {

      uygula();
   }
   public static void uygula(){

      try{
         //ProcessBuilder pb = new ProcessBuilder();
         //pb.command("notepad");
         ProcessBuilder pb = new ProcessBuilder();
         pb.command("mspaint");
         pb.start();

      }
      catch(Exception ex){
         //return;
      }
   }
}
```

Sonuç: Çalıştırıldığında “Paint” programını açar

```
public class Haf2_4 {

   /**
    * @param args the command line arguments
    */
   public static void main(String[] args) {
      try{
         ProcessBuilder pb= new ProcessBuilder(args[0]);
         Process p=pb.start();
         InputStream is=p.getInputStream();
         InputStreamReader isr= new InputStreamReader(is);
         BufferedReader br= new BufferedReader(isr);

         String line;
         while((line=br.readLine())!= null)
         {
            System.out.println(line);
         }
         br.close();
      }
      catch(Exception ex)
      {
         return;
      }
   }

}
```

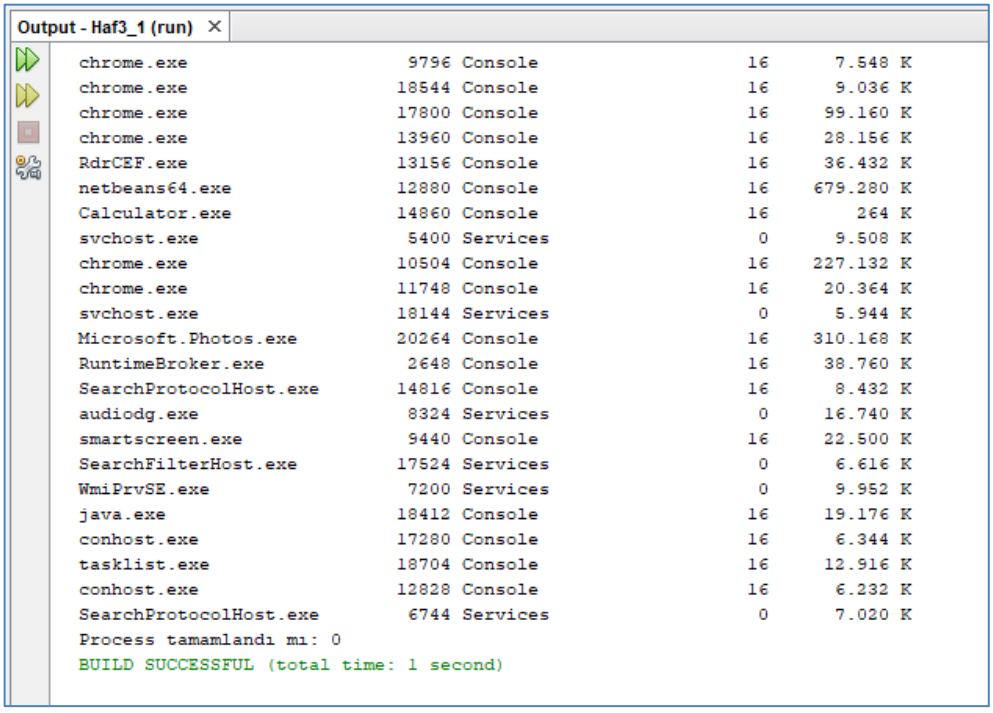
Haf2_4 Uygulamanın çalıştırılması



jar dosyası oluşturulduktan sonra komut istemi açılır.
Windows gezgininde klasör açıkken adres alanına “cmd” komutu yazılırsa, bulunduğumuz konumda komut İstemi açılacaktır.

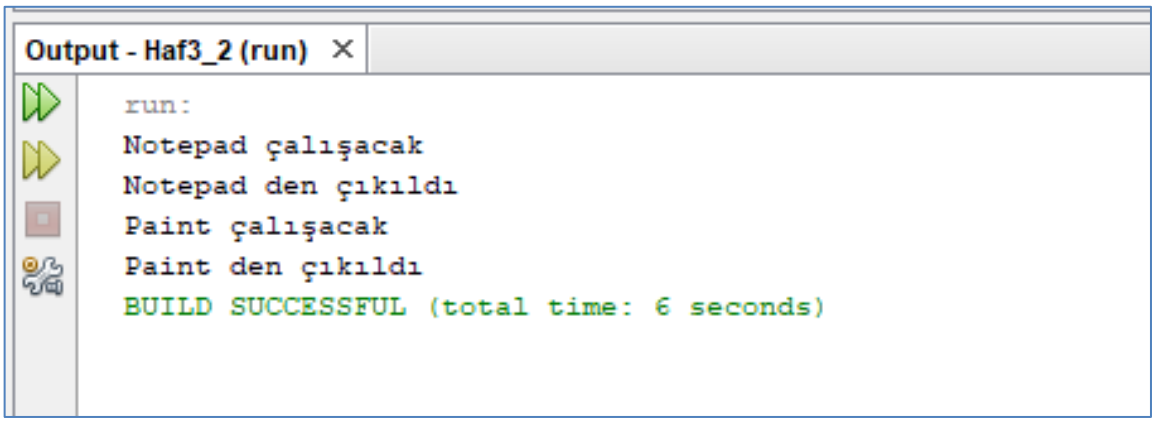
3. Hafta uygulamaları

```
package haf3_1;

import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;


public class Haf3_1 {

   public static void main(String[] args) {
      try{
         String line;
         Process p = Runtime.getRuntime().exec("tasklist");
         InputStream in = p.getInputStream();
         BufferedReader br = new BufferedReader(new InputStreamReader(in));

         while((line=br.readLine()) != null){
            System.out.println(line);
         }
         br.close();
         p.waitFor();
         int cikis = p.exitValue();
         System.out.println("Process tamamlandı mı: " + cikis);
      }
      catch(Exception ex){


      }
   }

}
```



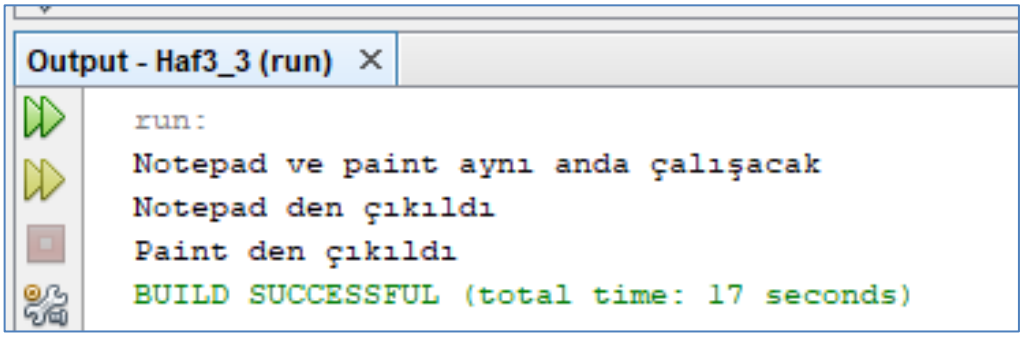
```
Output - Haf3_1 (run)
chrome.exe                    9796 Console                   16      7.548 K
chrome.exe                   18544 Console                   16      9.036 K
chrome.exe                   17800 Console                   16     99.160 K
chrome.exe                   13960 Console                   16     28.156 K
RdrCEF.exe                   13156 Console                   16     36.432 K
netbeans64.exe               12880 Console                   16    679.280 K
Calculator.exe               14860 Console                   16        264 K
svchost.exe                   5400 Services                   0      9.508 K
chrome.exe                   10504 Console                   16    227.132 K
chrome.exe                   11748 Console                   16     20.364 K
svchost.exe                  18144 Services                   0      5.944 K
Microsoft.Photos.exe         20264 Console                   16    310.168 K
RuntimeBroker.exe             2648 Console                   16     38.760 K
SearchProtocolHost.exe       14816 Console                   16      8.432 K
audiodg.exe                   8324 Services                   0     16.740 K
smartscreen.exe               9440 Console                   16     22.500 K
SearchFilterHost.exe         17524 Services                   0      6.616 K
WmiPrvSE.exe                  7200 Services                   0      9.952 K
java.exe                     18412 Console                   16     19.176 K
conhost.exe                  17280 Console                   16      6.344 K
tasklist.exe                 18704 Console                   16     12.916 K
conhost.exe                  12828 Console                   16      6.232 K
SearchProtocolHost.exe        6744 Services                   0      7.020 K
Process tamamlandı mı: 0
BUILD SUCCESSFUL (total time: 1 second)
```

```
package haf3_2;


public class Haf3_2 {


   public static void main(String[] args) {
      try{
         System.out.println("Notepad çalışacak");
         Process p1 = Runtime.getRuntime().exec("notepad");
         p1.waitFor();
         System.out.println("Notepad den çıkıldı");

         System.out.println("Paint çalışacak");
         Process p2 = Runtime.getRuntime().exec("mspaint");
         p2.waitFor();
         System.out.println("Paint den çıkıldı");

      }
      catch(Exception ex){

      }

   }

}
```



Çalışması: Önce “Not Defteri “ açılır. “Not defteri” programı kapatıldığında “Paint” programı açılır.

```
package haf3_3;

public class Haf3_3 {


   public static void main(String[] args) {
      try{
        System.out.println("Notepad ve paint aynı anda çalışacak");
        Process p1 = Runtime.getRuntime().exec("notepad");
        Process p2 = Runtime.getRuntime().exec("mspaint");

        p1.waitFor();
        System.out.println("Notepad den çıkıldı");

        p2.waitFor();
        System.out.println("Paint den çıkıldı");

      }
      catch(Exception ex){

      }
   }

}
```



Çalışması: “Not defteri” ve “paint” birlikte açılır.

```
package haf3_4;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.PrintWriter;

public class Haf3_4 {

   public static void main(String[] args)  {
      try{
         Runtime rt = Runtime.getRuntime();
         Process p = rt.exec("findstr java");
         OutputStream  out = p.getOutputStream();
         PrintWriter pw = new PrintWriter(out);

         pw.println("I love java");
         pw.println("I love tea");
         pw.println("I love coffee");
         pw.println("I like java");
         pw.println("I hate java");

         pw.close();
         InputStream in = p.getInputStream();
         BufferedReader br = new BufferedReader(new InputStreamReader(in));
         String line;

         while((line = br.readLine())!= null){
            System.out.println(line);
         }
         br.close();
         p.waitFor();
         int cikis = p.exitValue();
         System.out.println("Process tamamlandı mı: " + cikis);
      }

      catch(Exception ex){

   }
   }

}
```

```
Output - Haf3_4 (run)
   run:
   I love java
   I like java
   I hate java
   Process tamamlandı mı: 0
   BUILD SUCCESSFUL (total time: 0 seconds)
```

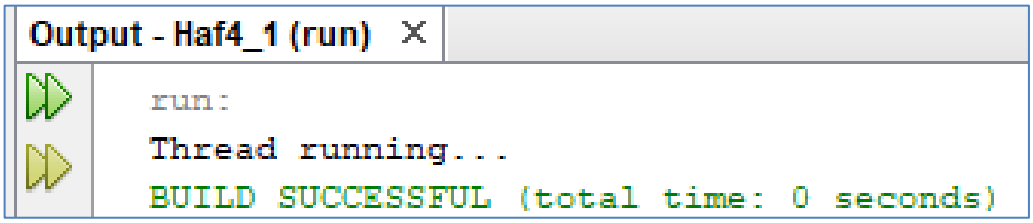
Çalışması: “pw” nesnesine verilen cümlelerden içinde “java” kelimesi geçenleri “findstr” ile bulur.

```
package haf3_5;
import java.io.IOException;
import java.io.OutputStream;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.BufferedReader;
import java.io.PrintWriter;

public class Haf3_5 {
   public static void main(String[] args) {
      try{
      Runtime rt = Runtime.getRuntime();
      Process p1 = rt.exec("ipconfig");
      InputStream in = p1.getInputStream();

      Process p2 = rt.exec("findstr 10");
      OutputStream out = p2.getOutputStream();

      int b;
      while((b = in.read())!= -1)
         out.write(b);
      p1.waitFor();
      in.close();
      out.close();

      in = p2.getInputStream();
      while((b=in.read())!= -1)
         System.out.write(b);
      p2.waitFor();
      in.close();
      }
      catch(Exception ex){
      }
   }
}
```

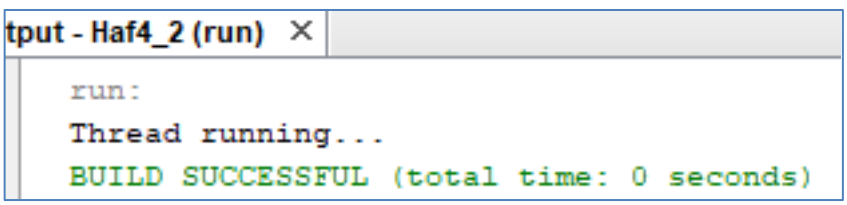
Çalışması: “ipconfig” ile gelen bilgilerden içinde “10” geçen satırları “findstr” ile bulur.

4. Hafta uygulamaları

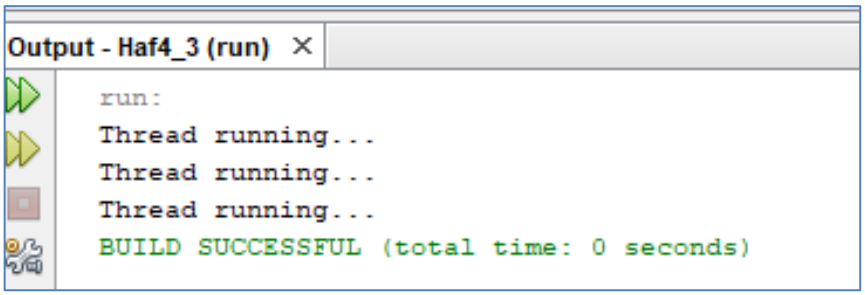
```
package haf4_1;

/**
 *
 * @author Sef2
 */
public class Haf4_1 extends Thread{

   public void run(){
      System.out.println("Thread running...");
   }
   public static void main(String[] args) {
      // TODO code application logic here

      Haf4_1 t = new Haf4_1();
      t.start();
   }

}
```




```
package haf4_2;

/**
 *
 * @author Sef2
 */
public class Haf4_2 implements Runnable{

   public void run(){
      System.out.println("Thread running...");
   }

   public static void main(String[] args) {
      Haf4_2 t = new Haf4_2();
      Thread x = new Thread(t);
      x.start();
   }

}
```



```
package haf4_3;

/**
 *
 * @author Sef2
 */
public class Haf4_3 extends Thread{

    public void run(){
        System.out.println("Thread running...");
    }

    public static void main(String[] args) {

        Haf4_3 t  = new Haf4_3();
        Haf4_3 t2  = new Haf4_3();
        Haf4_3 t3  = new Haf4_3();
        t.start();
        t2.start();
        t3.start();

    }

}
```

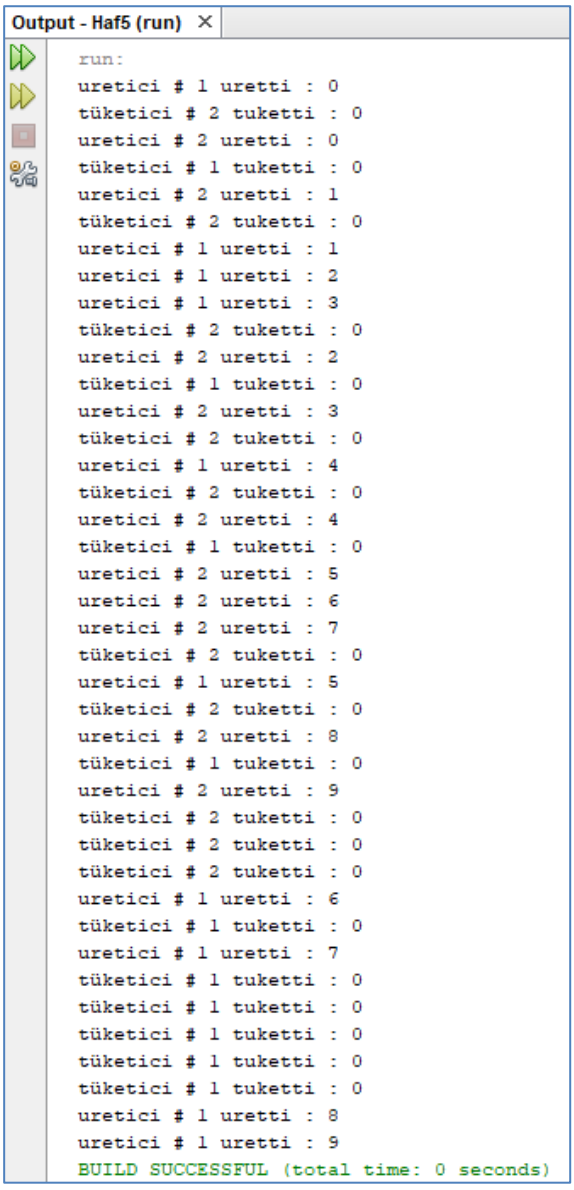


```
package haf4_4;

public class Haf4_4 {

    public static void main(String[] args) {
        PrintChar A = new PrintChar('A', 10);
        PrintChar B = new PrintChar('B', 10);
        PrintNum n = new PrintNum(10);

        A.start();
        B.start();
        n.start();
    }
}

class PrintChar extends Thread{
    char CharToPrint;
    int times;

    public PrintChar(char c, int t){
        CharToPrint = c;
        times=t;
    }
    public void run(){
        for(int i=0; i < times ; i++){
            System.out.println(CharToPrint);
        }
    }
}

class PrintNum extends Thread {
    int sayi;
    public PrintNum(int t){
        sayi= t;
    }

    public void run(){
        for(int i=0; i <= sayi ; i++)
            System.out.print(" " + i);
    }
}
```

5. Hafta uygulamaları

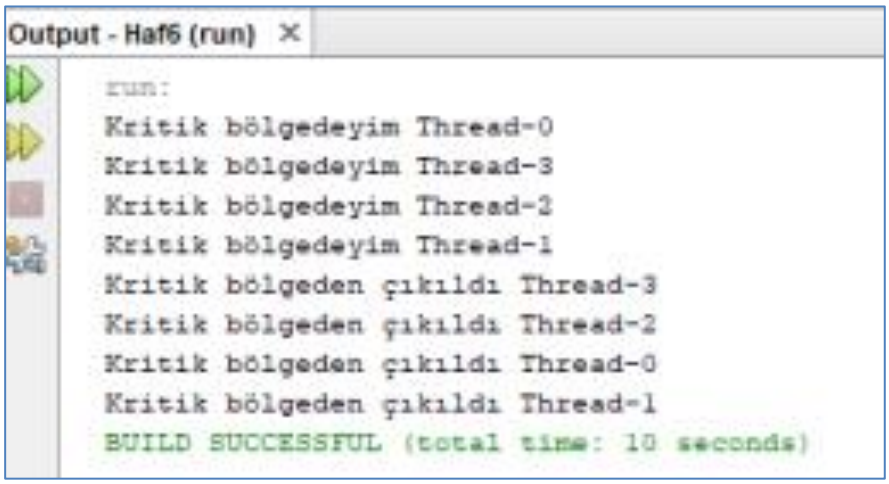
```
package haf5;
public class Haf5 {

   public static void main(String[] args) {
      uretTuket ut = new uretTuket();
      uretici p1 = new uretici(ut, 1);
      tuketici t1 = new tuketici(ut, 1);

      uretici p2 = new uretici(ut, 2);
      tuketici t2 = new tuketici(ut, 2);

      p1.start();
      p2.start();
      t1.start();
      t2.start();
   }

}

class uretTuket{
   private int urun;
   private boolean available= false;

   public synchronized void uret(int value){
      while(available==true){
         try{
            wait();
         }
         catch(Exception ex){

         }
         urun = value;
         available = true;
         notifyAll();
      }
   }

   public synchronized int tuket(){
      notifyAll();
      return urun;
   }
}

class uretici extends Thread{
   private uretTuket ut;
   private  int number;

   public uretici(uretTuket a, int sayi){
      ut = a;
      number = sayi;
   }
```

```
public void run(){
      for(int i=0; i < 10; i++){
         ut.uret(i);
         System.out.println("uretici # " + number + " uretti : " + i);
         try{
            //sleep(500);
         }
         catch(Exception ex){

         }
      }
   }

}

class tuketici extends Thread{
   private uretTuket ut;
   private  int number;

   public tuketici(uretTuket a, int sayi){
      ut = a;
      number = sayi;
   }

   public void run(){
      int value =0;
      for(int i=0; i < 10; i++){
         value = ut.tuket();
         System.out.println("tüketici # " + number + " tuketti : " + value);

      }
   }

}
```



6. Hafta uygulamaları

```
package haf6;
import java.util.concurrent.Semaphore;

public class Haf6 extends Thread{

   /**
    Kritik bölge problemini semaforla çözme
    */

   Semaphore sem = new Semaphore(3);

   public void run(){
      try{
         sem.acquire(); //Kritik bölge başlangıç
      }
      catch(Exception ex){

      }
      System.out.println("Kritik bölgedeyim " + getName());
      try{
       Thread.sleep(5000);
      }
      catch(Exception ex){

      }
      sem.release();
      System.out.println("Kritik bölgeden çıkıldı " + getName());
      try{
       Thread.sleep(5000);
      }
      catch(Exception ex){

      }
   }
   public static void main(String[] args) {
      Haf6 t1 = new Haf6();
      Haf6 t2 = new Haf6();
      Haf6 t3 = new Haf6();
      Haf6 t4 = new Haf6();

      t1.start();
      t2.start();
      t3.start();
      t4.start();

   }

}
```



```
package haf6_1;

import java.util.concurrent.Semaphore;

public class Haf6_1 {


   public static void main(String[] args) {
      Semaphore sem = new Semaphore(1);

      MyThread t1 = new MyThread(sem, "A");
      MyThread t2 = new MyThread(sem, "B");
      MyThread t3 = new MyThread(sem, "C");
      MyThread t4 = new MyThread(sem, "D");

      t1.start();
      t2.start();
      t3.start();
      t4.start();
   }
}

class MyThread extends Thread{
   Semaphore sem;
   String threadName;

   public MyThread(Semaphore sem, String name){
      this.sem = sem;
      threadName = name;
   }
   public void run(){
      System.out.println("Başlıyor " + threadName);

      try{
         System.out.println(threadName + " izin için bekliyor");
         sem.acquire();
         System.out.println(threadName + " izin aldı");

         for(int i=0; i<5; i++){
            System.out.println(threadName + " : " + i);
            //Thread.sleep(500);
         }
      }
      catch(Exception ex){

      }

      System.out.println(threadName + " izin bıraktı");
      sem.release();
   }

}
```