Hazal Gonen 131044028

Bu program SpecList isimli özellesmis bir classi implement eder.
SpecList classı LinkedList classını extend eder.
Generic bir classdir.
Her uyesini karsılaştırabilmek icin Comparable classini da extend eder.

SISTEM GEREKSINIMLERI:
Ilk olarak bir SpecList olusturulmalıdır.
Olusan Speclist objesine LinkedList ile yapilan butun hersey yapilabilir
Ek olarak 3 metod daha yapilabilir
Bunlar: addAllAtHead : listenin basina eleman ekler
getIntersectList : iki listenin kesisimini return eder
sortList : listeyi siralar.
: Siralama islemi parametre 0 ise buyukten kucuge 1 ise kucukten buyuge yapilir.

PROBLEM COZUM YONTEMLERI:
Problem LinkedListten ozellesmis bir class yazmamdi.
LinkedListten extend ettim ve boylece butun metodlarini kullanabilirim.
Butun tipler icin calismasi icin generic bir class yazdim
Generic tipleri siralarken karsilastirma yapabilmek icin Comparable classina ihtiyacim vardi bu sebeple onu da extend ettim.
addAllAtHead fonksiyonu en basa eklemeliydi bu sebeple addFirst metodunu kullandim
getIntersectList metodu kesisim kumesinde sadece unique elemanlar olmalıydı bu sebeple yardımcı bir fonksiyon yazdım. Eger daha once eklendiyse ekleme islemş yapmıyor.
sortList metodu hem decreasing hem increasing yapabilmesini if ile hallettim.

MAINDE YAZILAN TEST SENARYOLARI:

3 farkli tip icin (string, int , double) test fonksiyonu yazdim hepsinde yapilan islemler ayni tek fark icerdikleri bilgi. Bu sebeple sadece string olani acikliyorum.

Speclist objesi olusturdum.
add metodu ile 3 eleman ekledim.
icerigini ekrana bastim.

2.Speclist objesi olusturdum.
add ile eleman ekledim.
icerigini ekrana bastim.

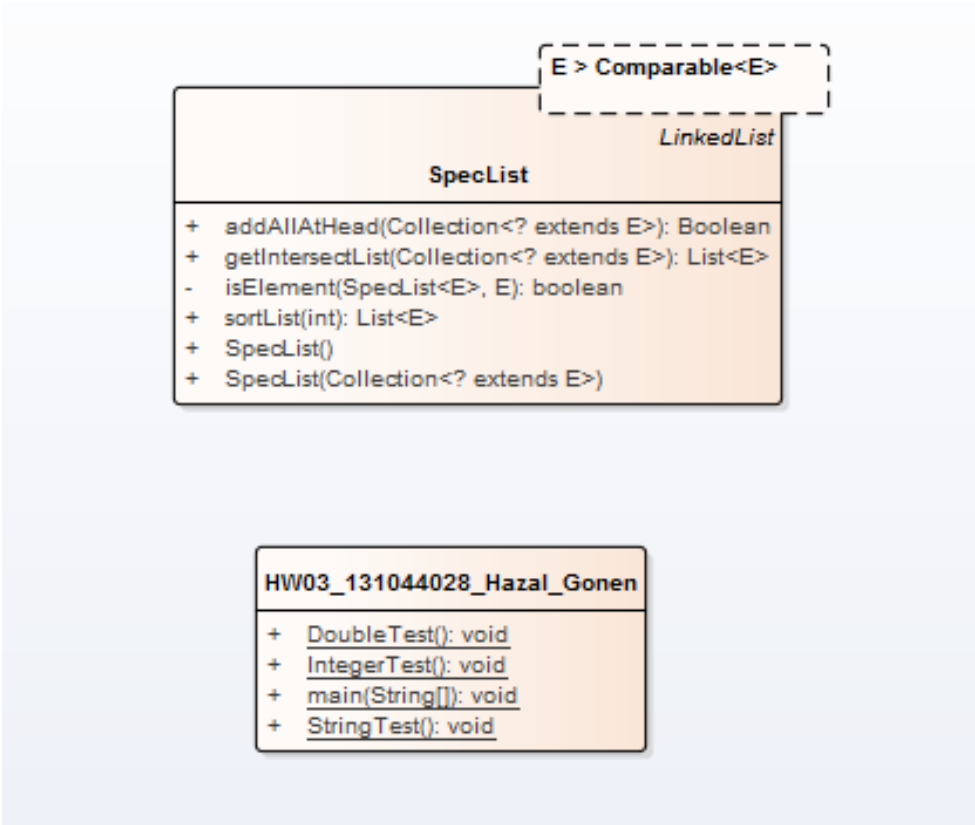
2.Speclist objesinde bulunan elemanlari ilkine addAllAtHead metoduyla ekledim.
eklemeden sonra icerigini ekrana bastim.

3.Speclist objesi olusturdum .
eleman ekledim .
icerigini ekrana bastim.

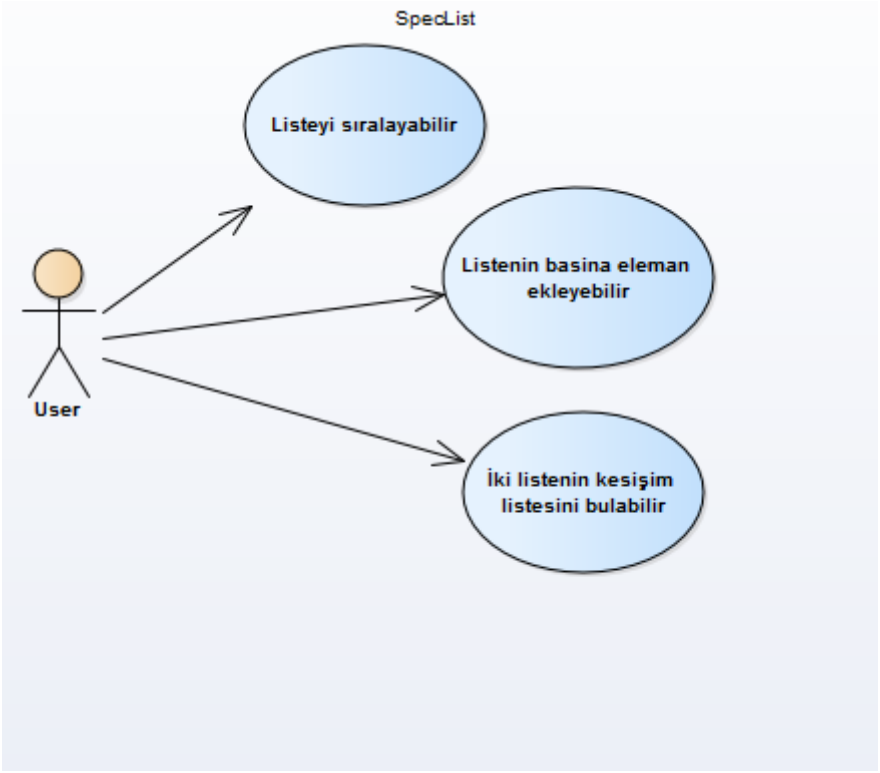
1.Speclist objesiyle kesisimlerini bulmak icin getIntersectList metodunu cagirdim ve yeni bir obje olusturup ona atadim.
kesisimleri iceren yeni objeyi ekrana bastim.

1.speclist objesini once buyukten kucuge daha sonra kucukten buyuge sortList metoduyla siraladim.
iceriklerini ekrana bastim.

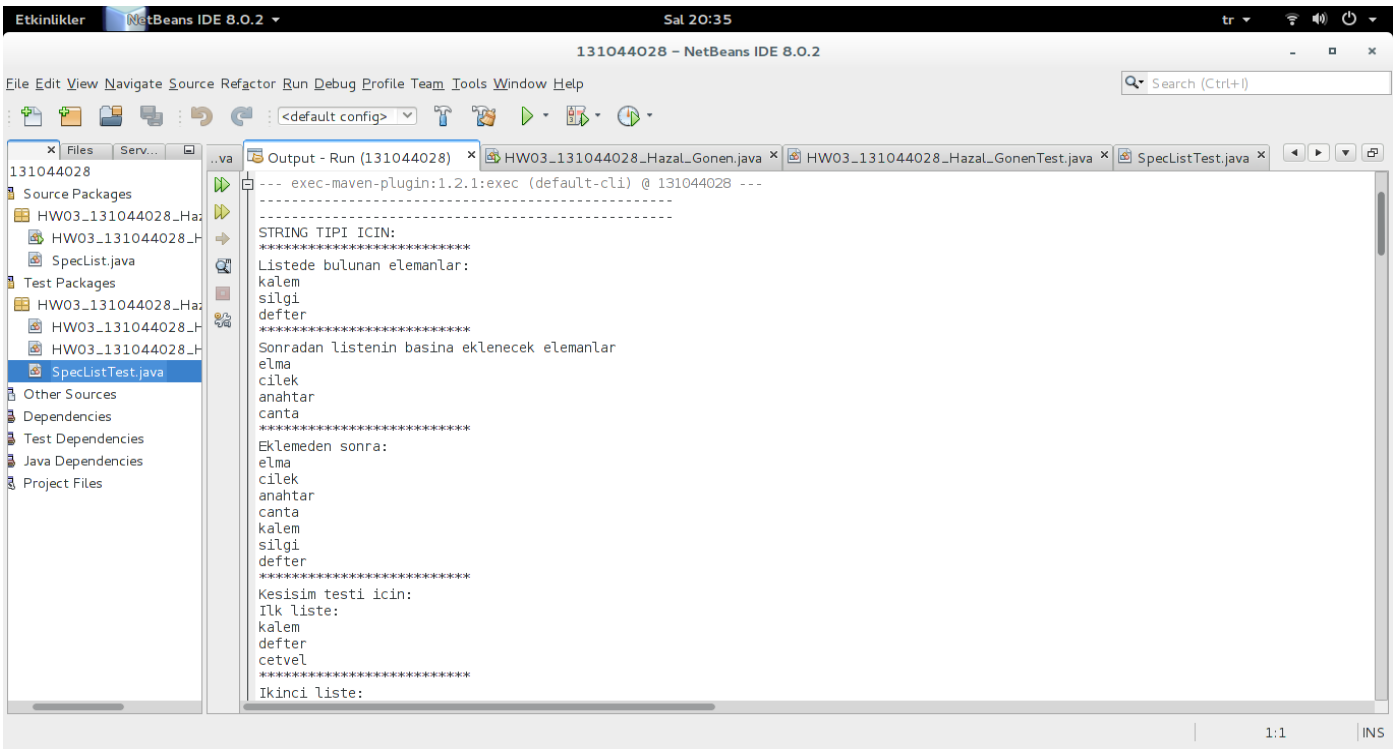
UML DIYAGRAMLARI

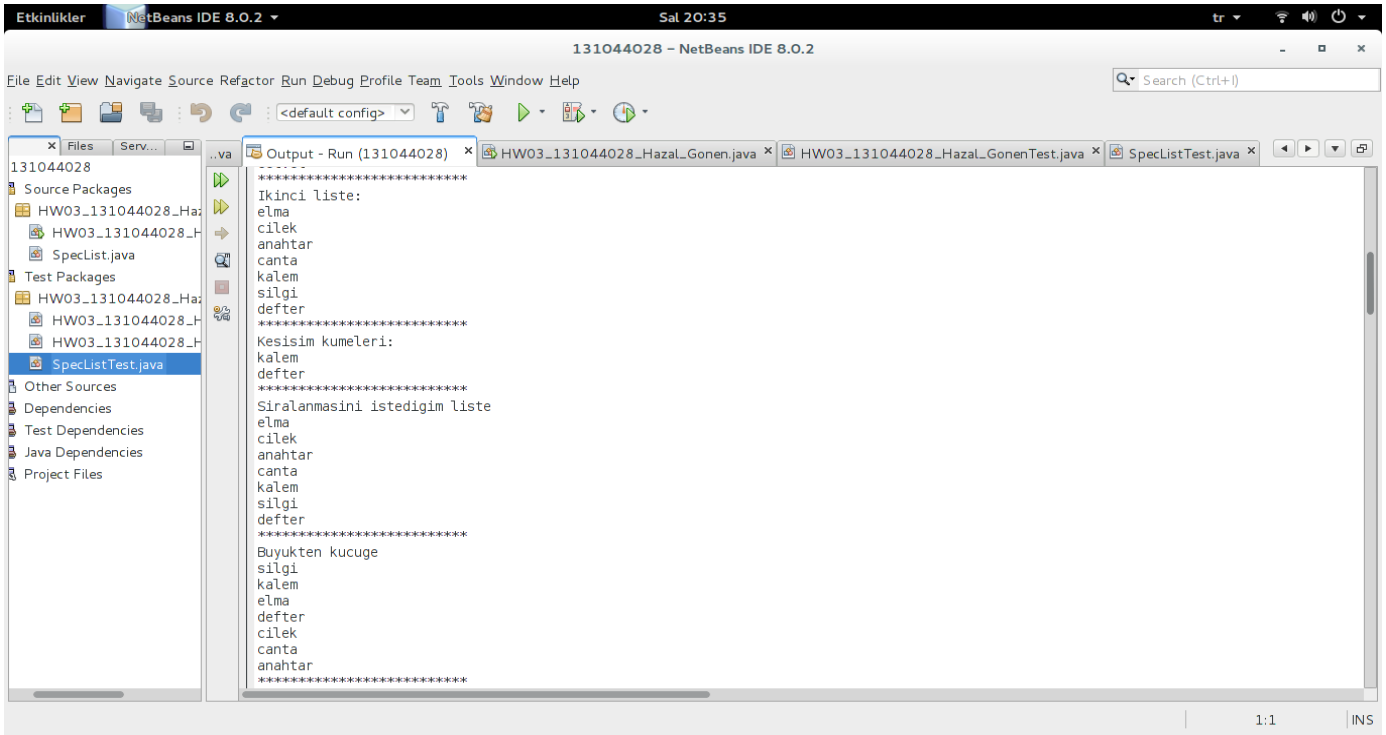


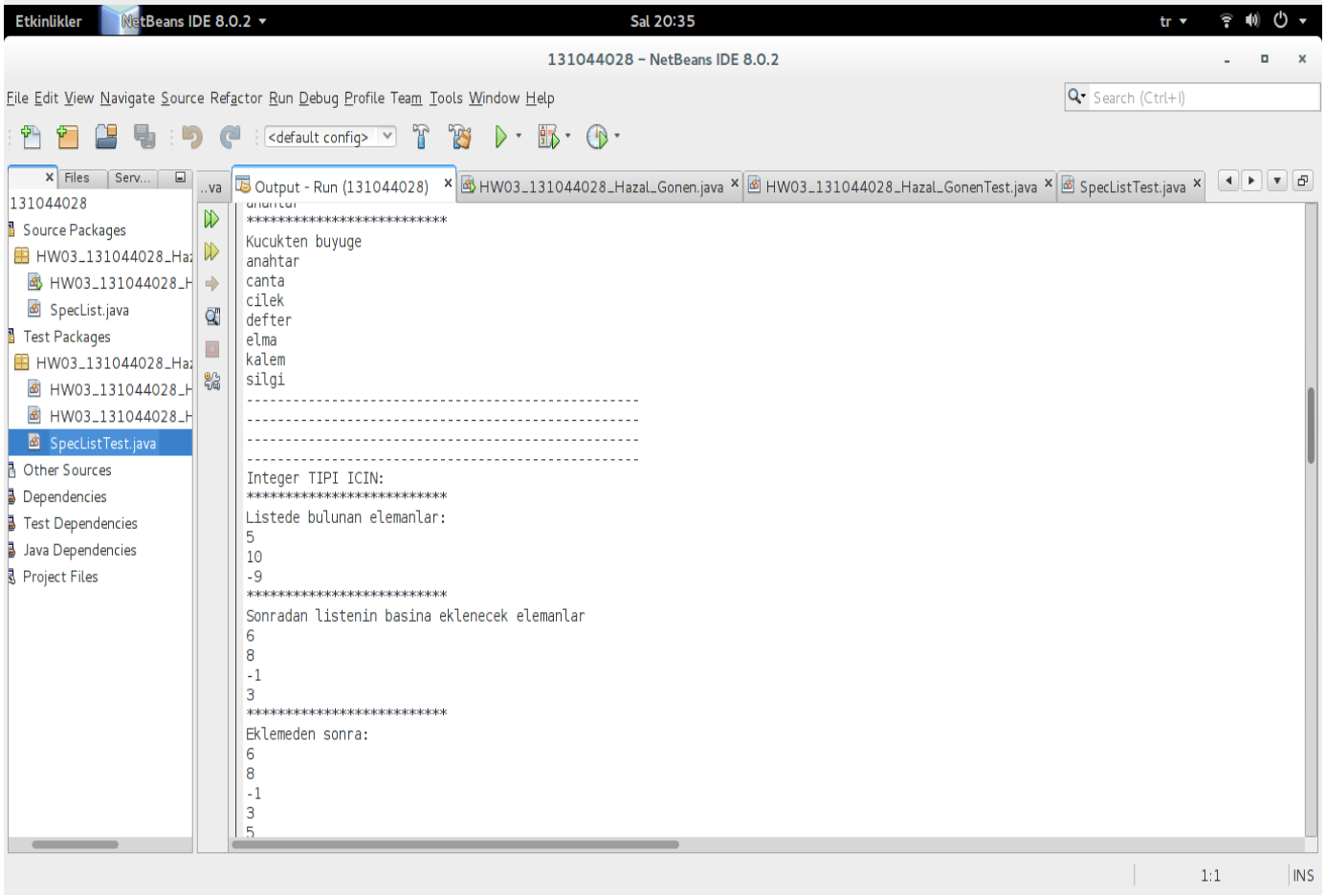
USE CASE DIYAGRAMLARI



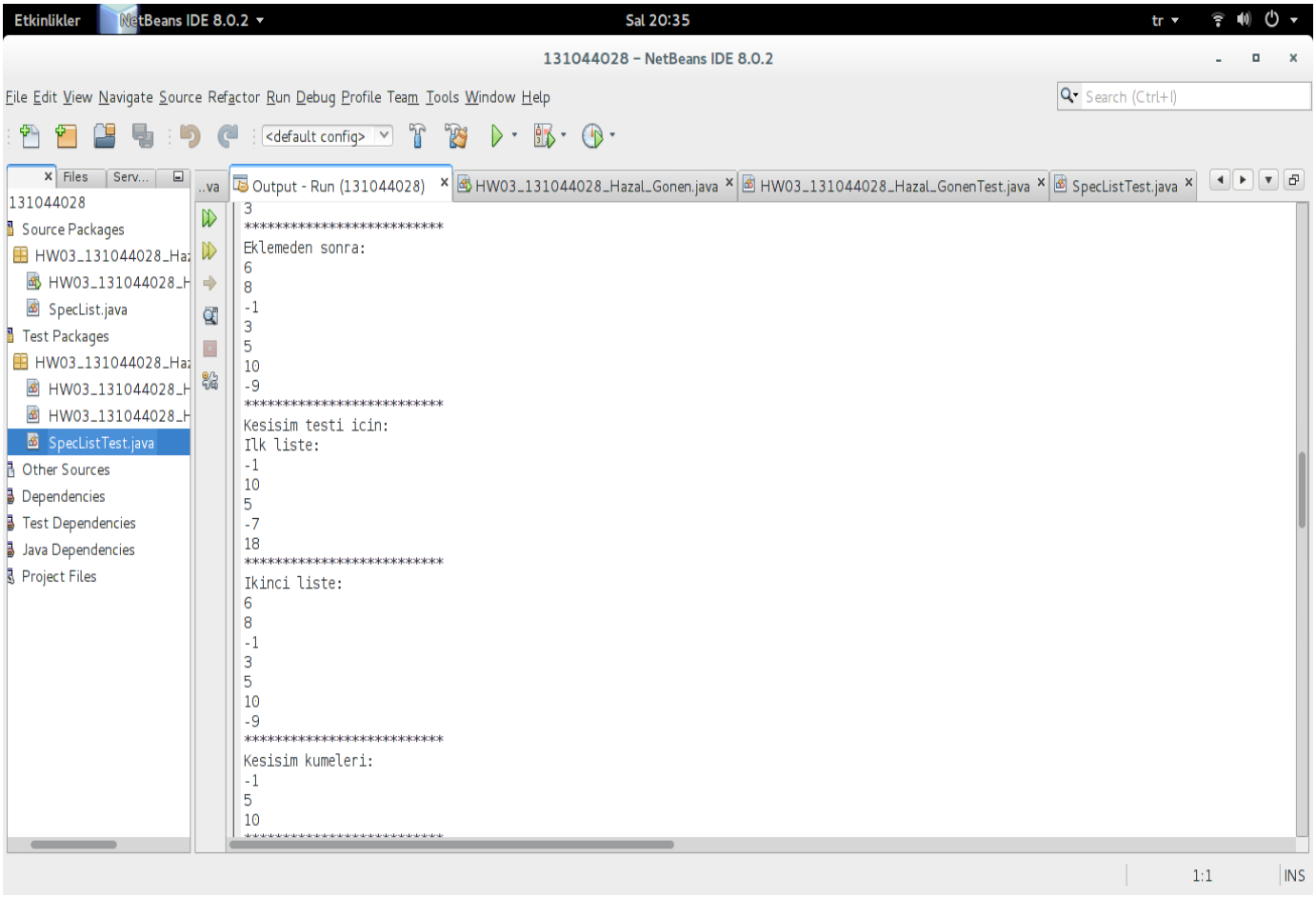
MAINDE YAZILAN TEST SENARYOLARININ SCREEN SHOTLARI

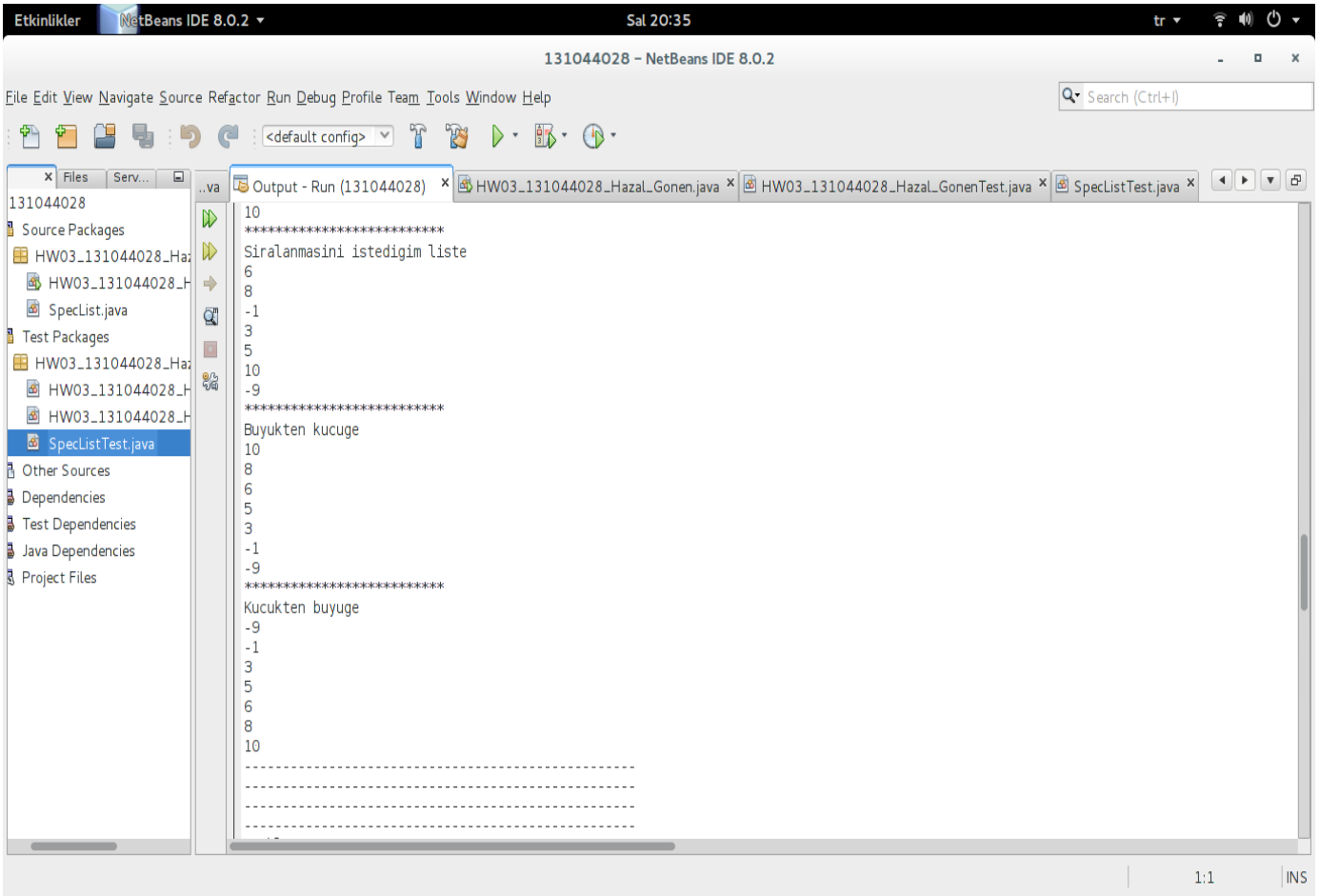


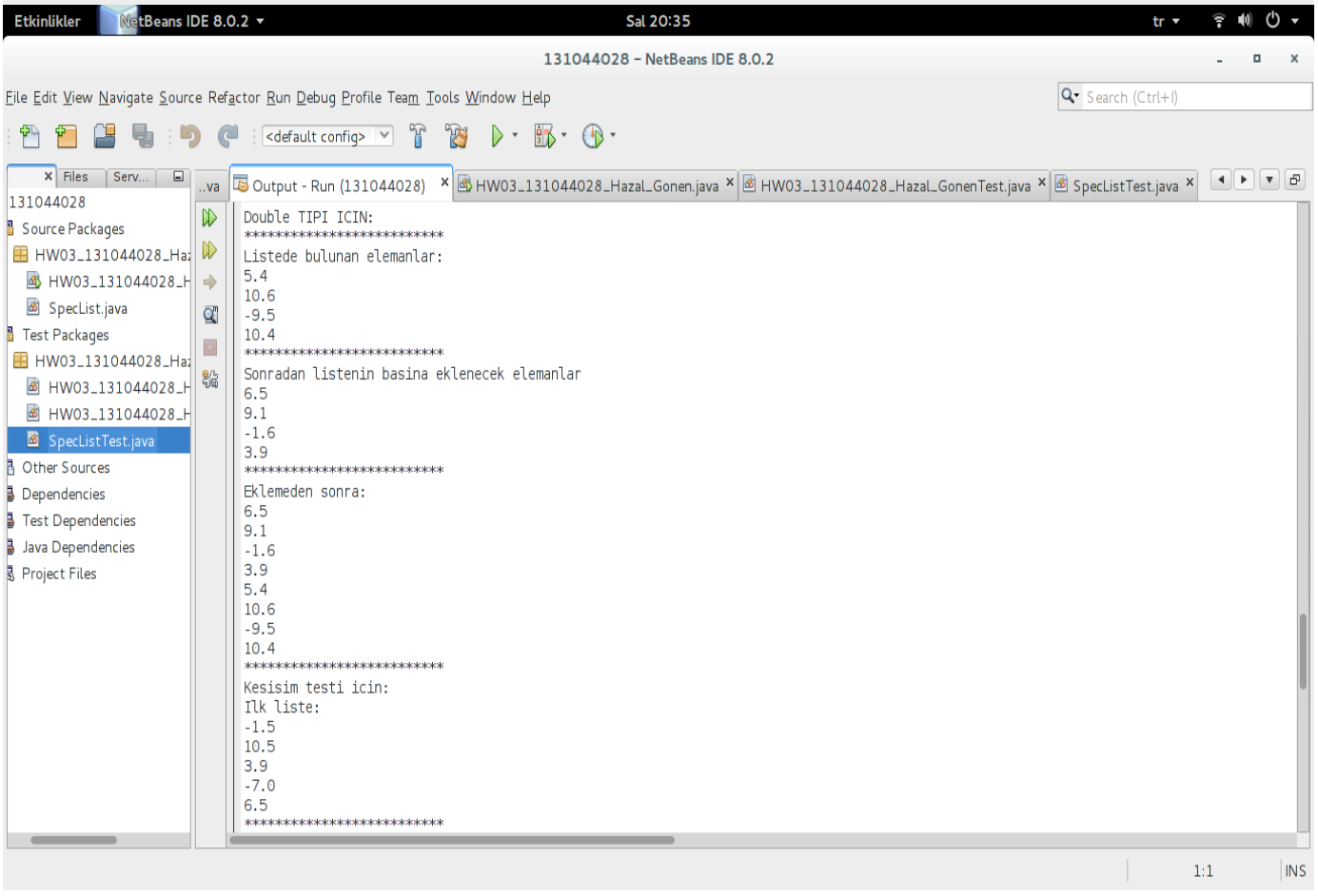

Etkinlikler
NetBeans IDE 8.0.2
Sal 20:35
tr
131044028 – NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Files
Serv...
131044028
Source Packages
HW03_131044028_Ha
HW03_131044028_H
SpecList.java
Test Packages
HW03_131044028_Ha
HW03_131044028_H
HW03_131044028_H
SpecListTest.java
Other Sources
Dependencies
Test Dependencies
Java Dependencies
Project Files
Output - Run (131044028)
HW03_131044028_Hazal_Gonen.java
HW03_131044028_Hazal_GonenTest.java
SpecListTest.java
****
Ikinci liste:
elma
cilek
anahtar
canta
kalem
silgi
defter
****
Kesisim kumeleri:
kalem
defter
****
Siralanmasini istedigim liste
elma
cilek
anahtar
canta
kalem
silgi
defter
****
Buyukten kucuge
silgi
kalem
elma
defter
cilek
canta
anahtar
****
1:1
INS

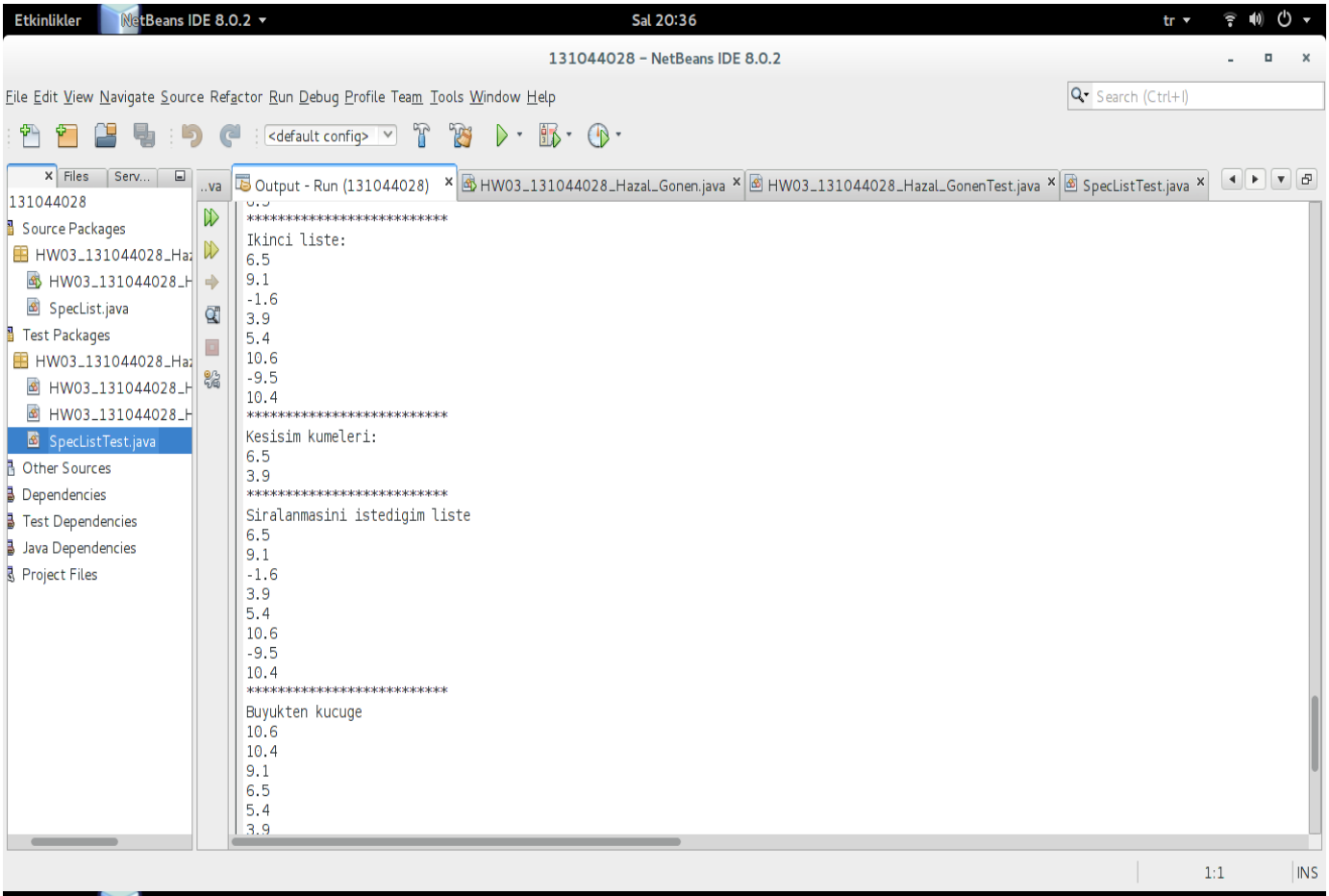

Etkinlikler
NetBeans IDE 8.0.2
Sal 20:35
tr
131044028 – NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Files
Serv...
131044028
Source Packages
HW03_131044028_Ha
HW03_131044028_H
SpecList.java
Test Packages
HW03_131044028_Ha
HW03_131044028_H
HW03_131044028_H
SpecListTest.java
Other Sources
Dependencies
Test Dependencies
Java Dependencies
Project Files
Output - Run (131044028)
HW03_131044028_Hazal_Gonen.java
HW03_131044028_Hazal_GonenTest.java
SpecListTest.java
****
Kucukten buyuge
anahtar
canta
cilek
defter
elma
kalem
silgi

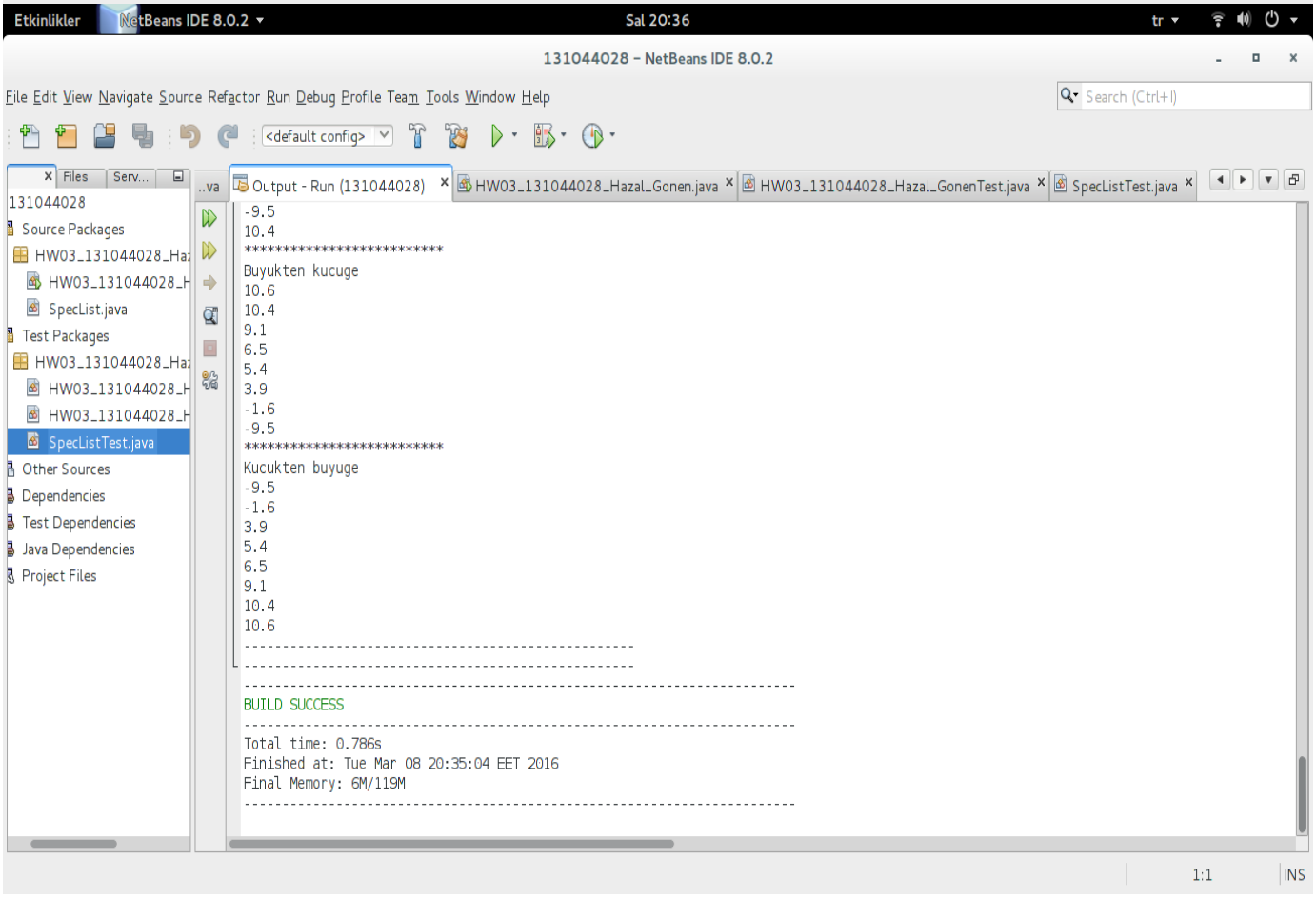
Integer TIPI ICIN:
****
Listede bulunan elemanlar:
5
10
-9
****
Sonradan listenin basina eklenecek elemanlar
6
8
-1
3
****
Eklemeden sonra:
6
8
-1
3
5
1:1
INS


Etkinlikler
NetBeans IDE 8.0.2
Sal 20:35
tr
131044028 – NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Files
Serv...
131044028
Source Packages
HW03_131044028_Haz
HW03_131044028_H
SpecList.java
Test Packages
HW03_131044028_Haz
HW03_131044028_H
HW03_131044028_H
SpecListTest.java
Other Sources
Dependencies
Test Dependencies
Java Dependencies
Project Files
..va
Output - Run (131044028)
HW03_131044028_Hazal_Gonen.java
HW03_131044028_Hazal_GonenTest.java
SpecListTest.java
3
****
Eklemeden sonra:
6
8
-1
3
5
10
-9
****
Kesisim testi icin:
Ilk liste:
-1
10
5
-7
18
****
Ikinci liste:
6
8
-1
3
5
10
-9
****
Kesisim kumeleri:
-1
5
10
****
1:1
INS


Etkinlikler
NetBeans IDE 8.0.2
Sal 20:35
tr
131044028 – NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Files
Serv...
131044028
Source Packages
HW03_131044028_Haz
HW03_131044028_H
SpecList.java
Test Packages
HW03_131044028_Haz
HW03_131044028_H
HW03_131044028_H
SpecListTest.java
Other Sources
Dependencies
Test Dependencies
Java Dependencies
Project Files
..va
Output - Run (131044028)
HW03_131044028_Hazal_Gonen.java
HW03_131044028_Hazal_GonenTest.java
SpecListTest.java
10
****
Siralanmasini istedigim liste
6
8
-1
3
5
10
-9
****
Buyukten kucuge
10
8
6
5
3
-1
-9
****
Kucukten buyuge
-9
-1
3
5
6
8
10
1:1
INS


Etkinlikler
NetBeans IDE 8.0.2
Sal 20:35
tr
131044028 – NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Files
Serv...
131044028
Source Packages
HW03_131044028_Haz
HW03_131044028_H
SpecList.java
Test Packages
HW03_131044028_Haz
HW03_131044028_H
HW03_131044028_H
SpecListTest.java
Other Sources
Dependencies
Test Dependencies
Java Dependencies
Project Files
..va
Output - Run (131044028)
HW03_131044028_Hazal_Gonen.java
HW03_131044028_Hazal_GonenTest.java
SpecListTest.java
Double TIPI ICIN:
****
Listede bulunan elemanlar:
5.4
10.6
-9.5
10.4
****
Sonradan listenin basina eklenecek elemanlar
6.5
9.1
-1.6
3.9
****
Eklemeden sonra:
6.5
9.1
-1.6
3.9
5.4
10.6
-9.5
10.4
****
Kesisim testi icin:
Ilk liste:
-1.5
10.5
3.9
-7.0
6.5
****
1:1
INS


Etkinlikler
NetBeans IDE 8.0.2
Sal 20:36
tr
131044028 – NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Files
Serv...
131044028
Source Packages
HW03_131044028_Haz
HW03_131044028_H
SpecList.java
Test Packages
HW03_131044028_Haz
HW03_131044028_H
HW03_131044028_H
SpecListTest.java
Other Sources
Dependencies
Test Dependencies
Java Dependencies
Project Files
..va
Output - Run (131044028)
HW03_131044028_Hazal_Gonen.java
HW03_131044028_Hazal_GonenTest.java
SpecListTest.java
****
Ikinci liste:
6.5
9.1
-1.6
3.9
5.4
10.6
-9.5
10.4
****
Kesisim kumeleri:
6.5
3.9
****
Siralanmasini istedigim liste
6.5
9.1
-1.6
3.9
5.4
10.6
-9.5
10.4
****
Buyukten kucuge
10.6
10.4
9.1
6.5
5.4
3.9
1:1
INS

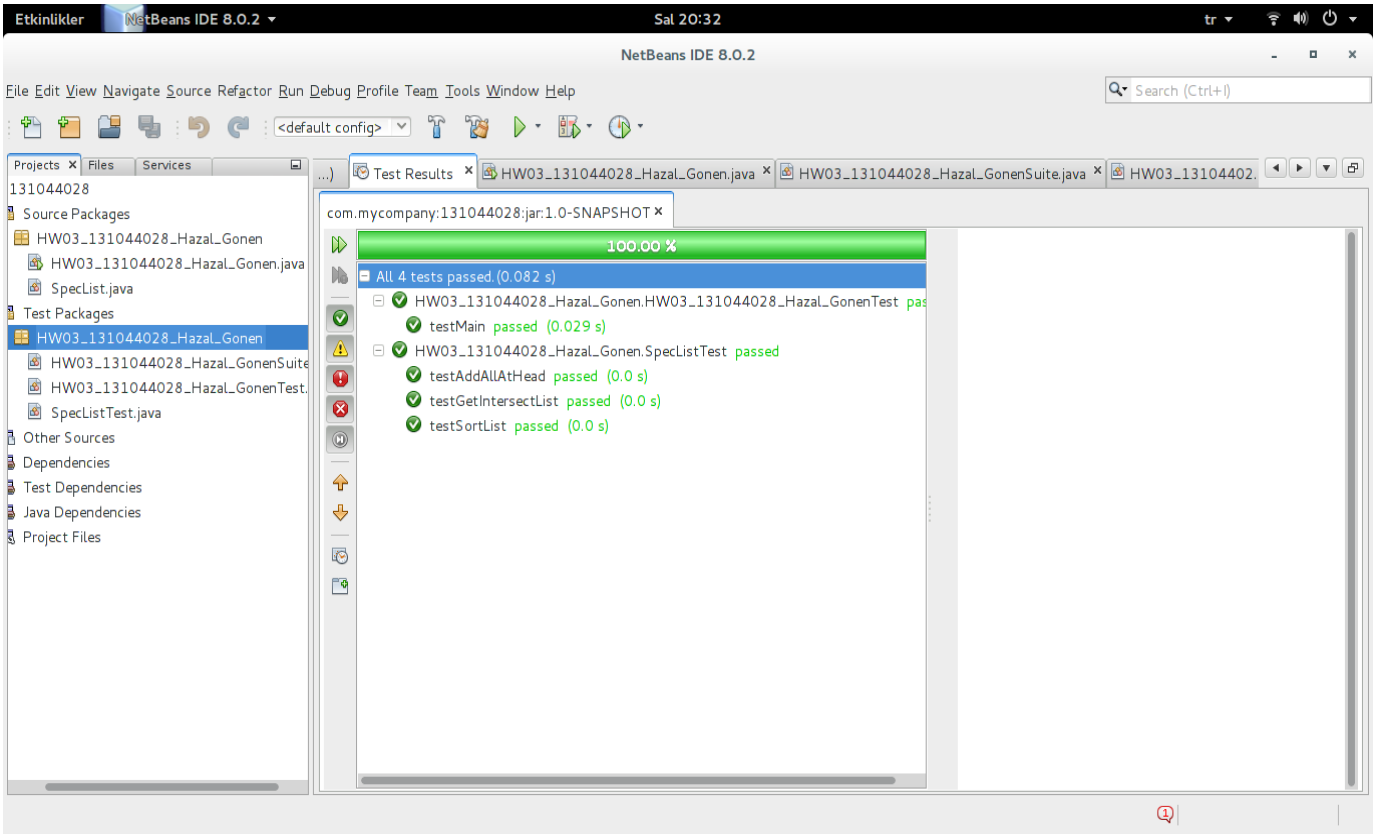

Etkinlikler
NetBeans IDE 8.0.2
Sal 20:36
tr
131044028 – NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Files
Serv...
131044028
Source Packages
HW03_131044028_Haz
HW03_131044028_H
SpecList.java
Test Packages
HW03_131044028_Haz
HW03_131044028_H
HW03_131044028_H
SpecListTest.java
Other Sources
Dependencies
Test Dependencies
Java Dependencies
Project Files
..va
Output - Run (131044028)
HW03_131044028_Hazal_Gonen.java
HW03_131044028_Hazal_GonenTest.java
SpecListTest.java
-9.5
10.4
****
Buyukten kucuge
10.6
10.4
9.1
6.5
5.4
3.9
-1.6
-9.5
****
Kucukten buyuge
-9.5
-1.6
3.9
5.4
6.5
9.1
10.4
10.6

BUILD SUCCESS

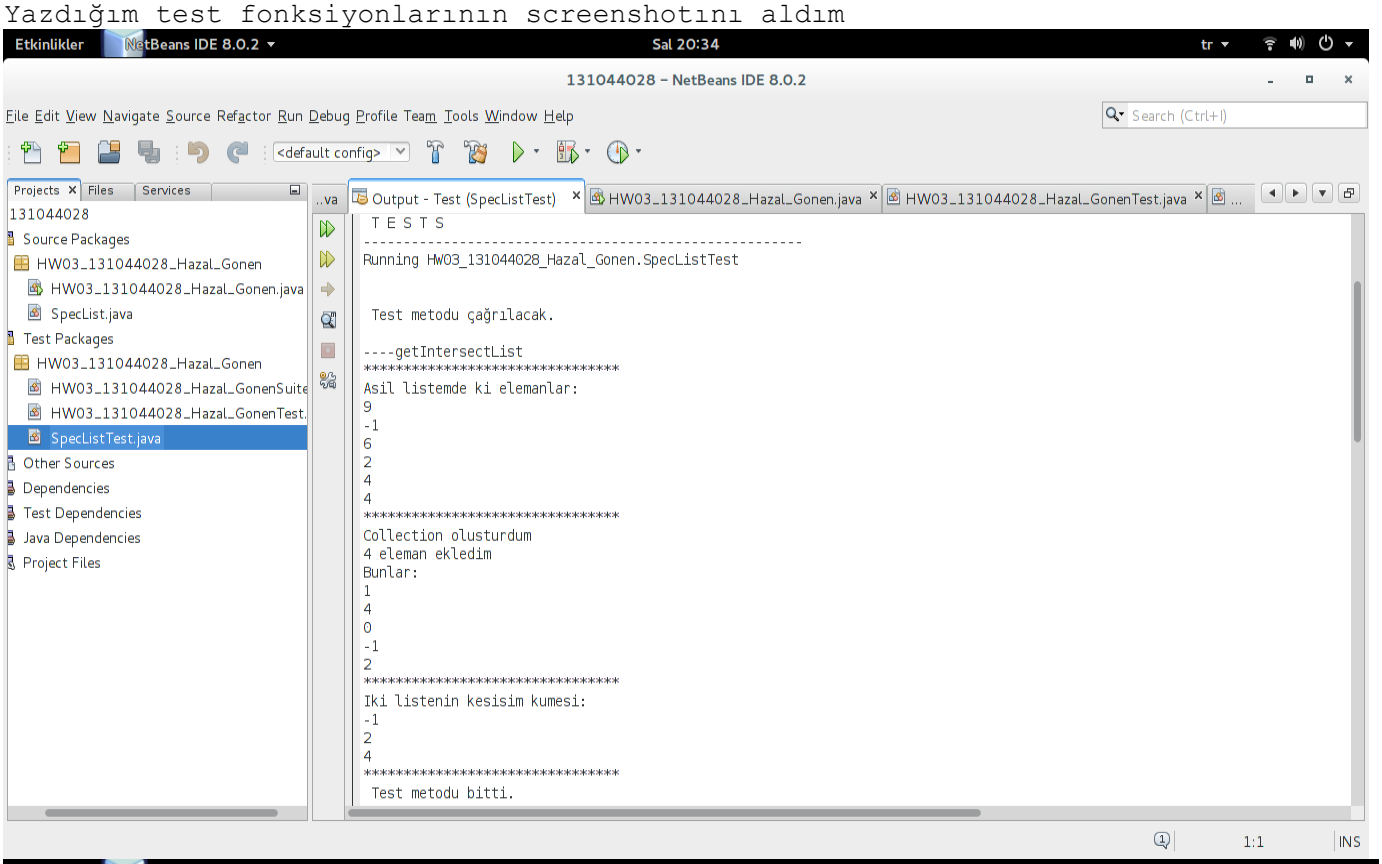
Total time: 0.786s
Finished at: Tue Mar 08 20:35:04 EET 2016
Final Memory: 6M/119M

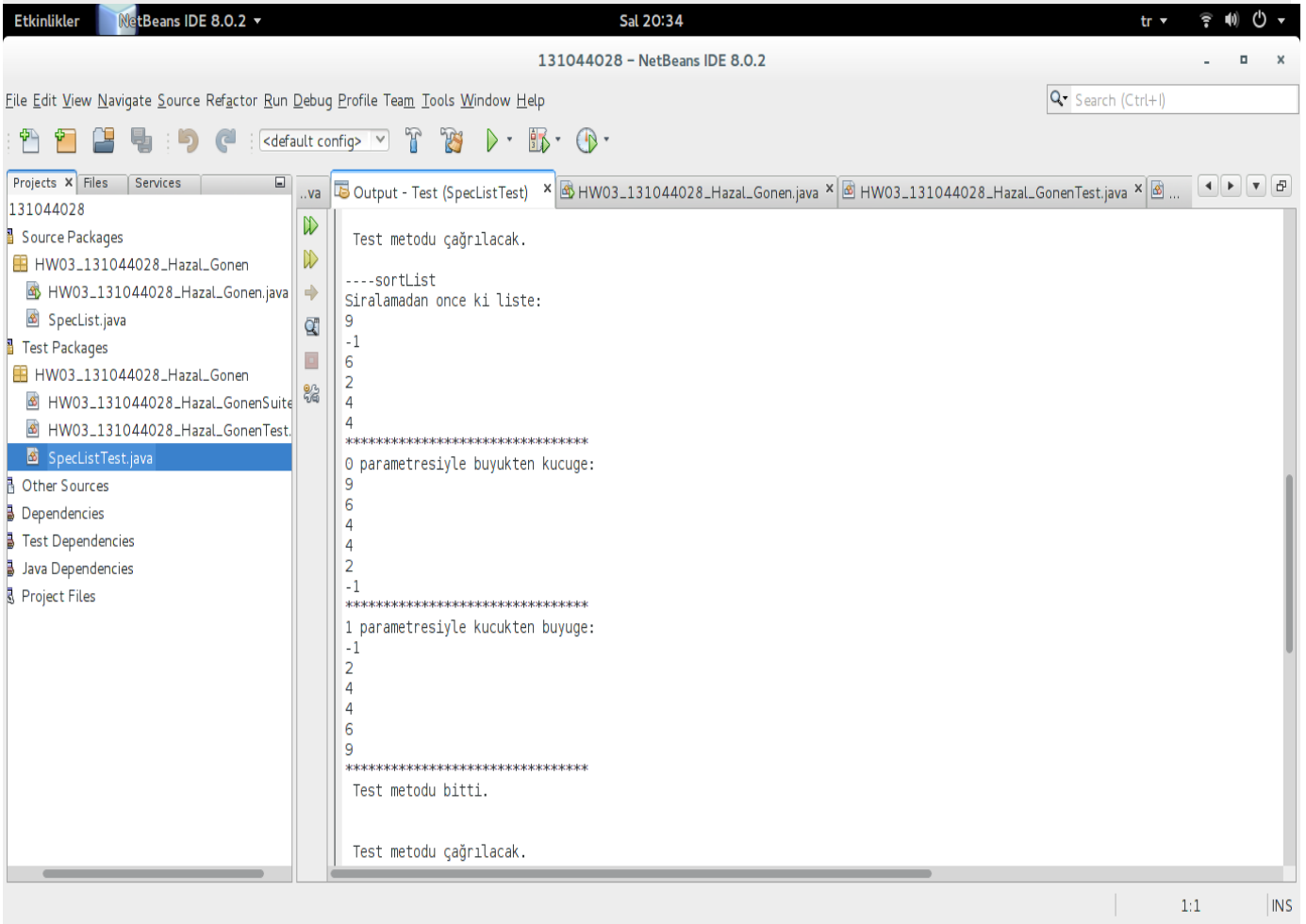
1:1
INS

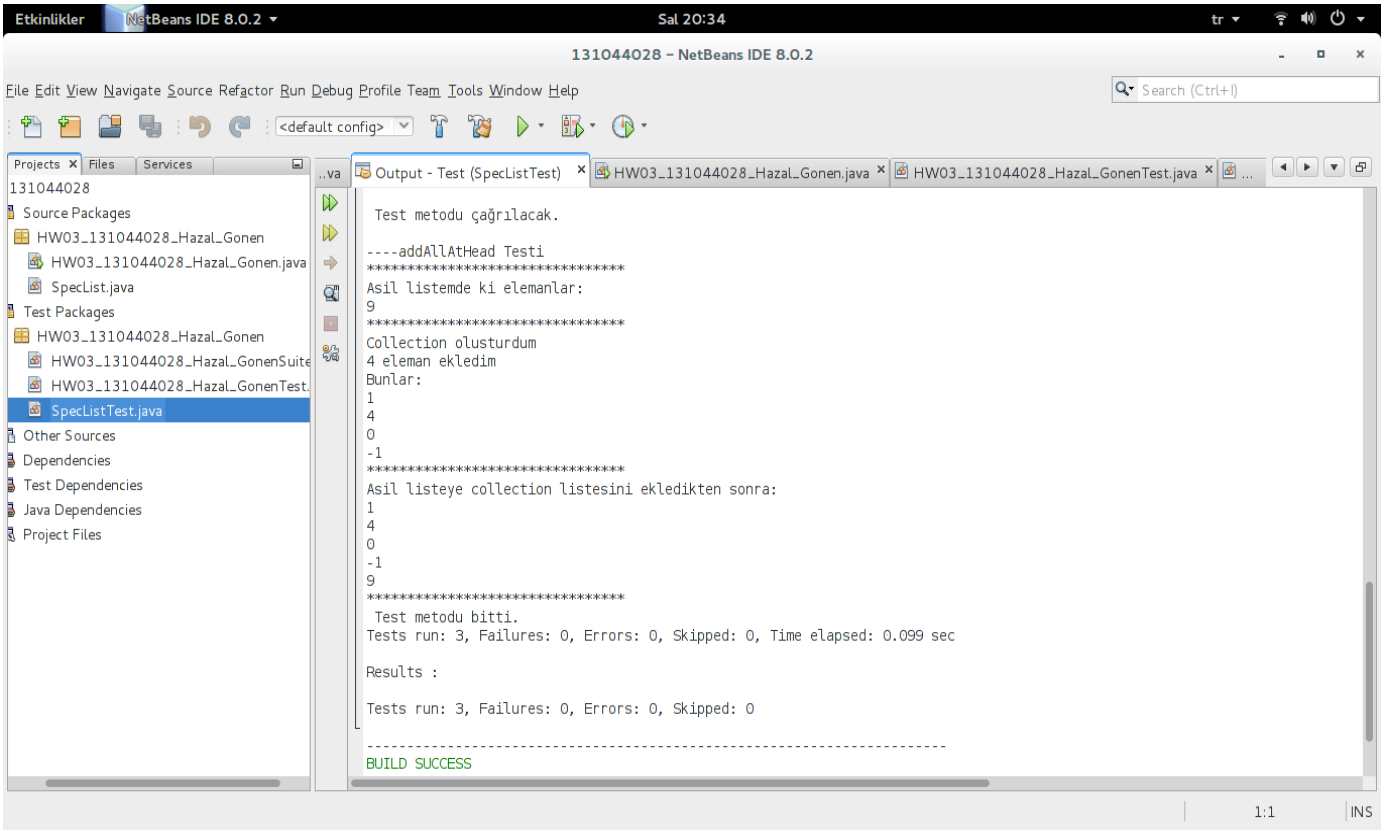
JUNIT TESLERI
Butun testler basarıyla sonuçlandı.



Yazdığım test fonksiyonlarının screenshotını aldım







FONKSIYONLARIN CALISMA SURESI ANALIZI

```
Boolean addAllAtHead(Collection<? extends E> c){
①     try{
②         SpecList<E> newList = (SpecList<E>) c;          → 1
③         for(int i =newList.size-1; i>=0; --i){          } n
④             this.addFirst(newList.get(i));              → 1 }
⑤         }
⑥     } catch (NullPointerException | IllegalStateException |
⑦             ClassCastException | IllegalArgumentException e)
⑧             System.out.println("ERROR!! HATA OLUSTU!!");  → 1
⑨             return false;                               → 1
⑩     }
⑪ return true;                                            → 1
  }
```

4. satırda addFirst metodu constant zaman alır.

3. satırdaki döngünün çalışması n kadar sürer

$\sum_{0}^{n-1} 1 = n$ $O(n)$ mi diye bakıyorum $f(n) \leq g(n) \cdot c$

$n \leq 2n$, $c=2$ olsun

$c=1$ için $n=1$ olsun $1 \leq 2$

$n=k$ için $k \leq 2k$ olduğunu varsayalım

$n=k+1$ için $k+1 \leq 2k+2$ → $k \leq 2k+1$ ispatladım.

Dolayısıyla fonksiyon worste case : $O(n)$

amortized : $\frac{O(n)}{n}$

Eğer fonksiyon catch bloğuna girerse

8. satır constant zaman alır

9. satır constant zaman alır

Toplam = 2 = O(1) 'dir. (ispatlamaya gerek yok.)

Böylece Worst Case = O(n)

Best Case = O(1)

Amortized = $\frac{O(n)}{n}$

```
List<E> getIntersectList(Collection<? extends E> c){
① try{
②     SpecList<E> castLinkedList = (SpecList<E>)c;          -> 1
③     SpecList<E> intersectList = new SpecList();          -> 1
④     for(int i=0; i<this.size; ++i){
⑤         for(int j=0; j<castLinkedList.size(); j++){
⑥             if(this.get(i).equals(castLinkedList.get(j))){   -> 2n+1
⑦                 if( isElement(intersectList, this.get(i)) ==true){   -> n^2+n
⑧                     intersectList.add(this.get(i));          -> n+1
                  }                                            (1)  (n)
              }
          }
      }
⑨     return intersectList;                                    -> 1
⑩ }catch( .. ){
      //elsik satirlar var
  }
⑪ return null;
}
```

2. satirin calismasi : constant zaman alir.
3. satirin calismasi : constant zaman alir.
8. satirin calismasi : LinkedList'in add metodu - n+1 zaman alir
7. satirin calismasi : isElement fonksiyonu n^2+n zaman alir. (ileride aciklayacagim)
6. satirin calismasi : get'ler n zaman sürer equals metodu constant zaman sürer toplam 2n+1

6, 7, 8. satır toplam $n+1 + n^2+n+2n+1 = n^2+4n+2$

4 ve 5. satırlar : $\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} n^2+4n+2$

$\sum_{j=0}^{n-1} n^2.(n) + 4n(n) + 2(n)$

n^3+4n^2+2n

$\sum_{i=0}^{n-1} n^3+4n^2+2n$

$= n^4+4n^3+2n^2$

Toplam calisma süresi $= n^4+4n^3+2n^2$

$O(n^4)$ 'mu diye bakıyorum

$n^4+4n^3+2n^2 \leq c \cdot n^4$ $c=7$ için

① $n=1$ için $1+4+2 \leq 1.7$ sağlanır.

② $n=k$ için $k^4+4k^3+2k^2 \leq 7k^4$ varsaydım $4k^3+2k^2 \leq 6k^4$

③ $n=2k$ için $16k^4+32k^3+8k^2 \leq 112k^4$ $32k^3+8k^2 \leq 96k^4$

2. satırı 18 ile carparsan $32k^3+16k^2 \leq 48k^4$

$32k^3+8k^2 \leq 96k^4$

iki tarafa $8k^2$ ekle olur $32k^3+16k^2-8k^2 \leq 96k^4$

Böylece worst case $= O(n^4)$ burası $48k^2$'den küçük dolayısıyla sağlandı

best case $= O(1)$ $\Rightarrow$ catch bloğu

amortized $= \frac{O(n^4)}{n}$

```
List<E> sortList(int dec_in){
①     boolean swapped;                                    → 1
②     specList<E> newSortList = newSpecList<>();          → 1
      try{
③       newSortList.addAll(this);                         → n
        do{
④         swapped =false;
⑤         for(int i=0; i<newSortList.size()-2; ++i){
⑥           if((( dec_in ==1 && (newSortList.get(i).compareTo(
⑦               newSortList.get(i+1)))>0) || dec_in==0 &&
⑧             (newSortList.get(i).compareTo(newSortList.get(i+1)))<0)){
⑨             E temp= newSortList.get(i);                  → n
⑩             newSortList.set(i, newSortList.get(i+1));   → n+1
⑪             newSortList.set(i+1, temp);                 → 1
⑫             swapped true;                               → 1
            }
          }
⑬         if(!swapped){                                   → 1
⑭           break;                                        → 1
          }
⑮         swapped = false;                                → 1
⑯         for(int i=newSortList.size()-2; i>=0;i--){
⑰           if(((dec_in==1 && (newSortList.get(i).compareTo(
⑱             newSortList.get(i+1)))>0) || dec_in==0 &&
⑲           (newSortList.get(i).compareTo(newSortList.get(i+1)))<0)){
```

2n+2

2n+3

n

$4n^2+n$

```
20)                 E  temp = newSortList.get(i);
21)                    newSortList.set(i, newSortList.get(i+1));
22)                    newSortList.set(i+1, temp);
23)                    swapped = true;
                    }
                }
            } while(swapped);
        } catch( . . . ){
24)     }
        return newSortList;
    }
```

3. satırda addAll metodu n zaman alır

9. satırda get : n zaman

10. satırda set ve get : n+1 zaman

11. satırda set = 1

12 satır 1

6-7-8. satır 2n+2 zaman

} 4n+5

5. satır: $\sum_{j=0}^{n-2} 4n+5 = (4n+5)(n-1) = (4n^2+n-5)$ ..

17 satırdan itiboren aynı şey tekrar edilir $= (4n^2+n-5)$

do-while olduğu için $\sum_{i=0}^{n-1} 8n^2 + 2n - 10 = 8n^3+2n^2-10n$

$8n^3+2n^2-10n \Rightarrow O(n^3)$ mü ?

$8n^3+2n^2-10n \leq c.n^3$ $\quad c=1 \quad n=1$ için

$8+2-10 \leq 1$

$0 \leq 1$ olduğundan $O(n^3)$'dür.

Best case try catch olursa $O(1)$'dir.