Hazal Gönen 131044028
HW06_VERİ YAPILARI RAPOR

İçindekiler

Bu program 3 farklı part icerir bunlar:

1. HuffmanCode encode method
2. BinarySearchTree iterator class
3. Implement priority queue classes

HUFFMANTREE ICIN SISTEM GEREKSINIMLERI:

Bu class huffmanTree'yi implement eden classdir.
Icinde buildTree , decode, printCode,toString metodlarini barindirir.
Benden istenen ise encode metodu yazmam.

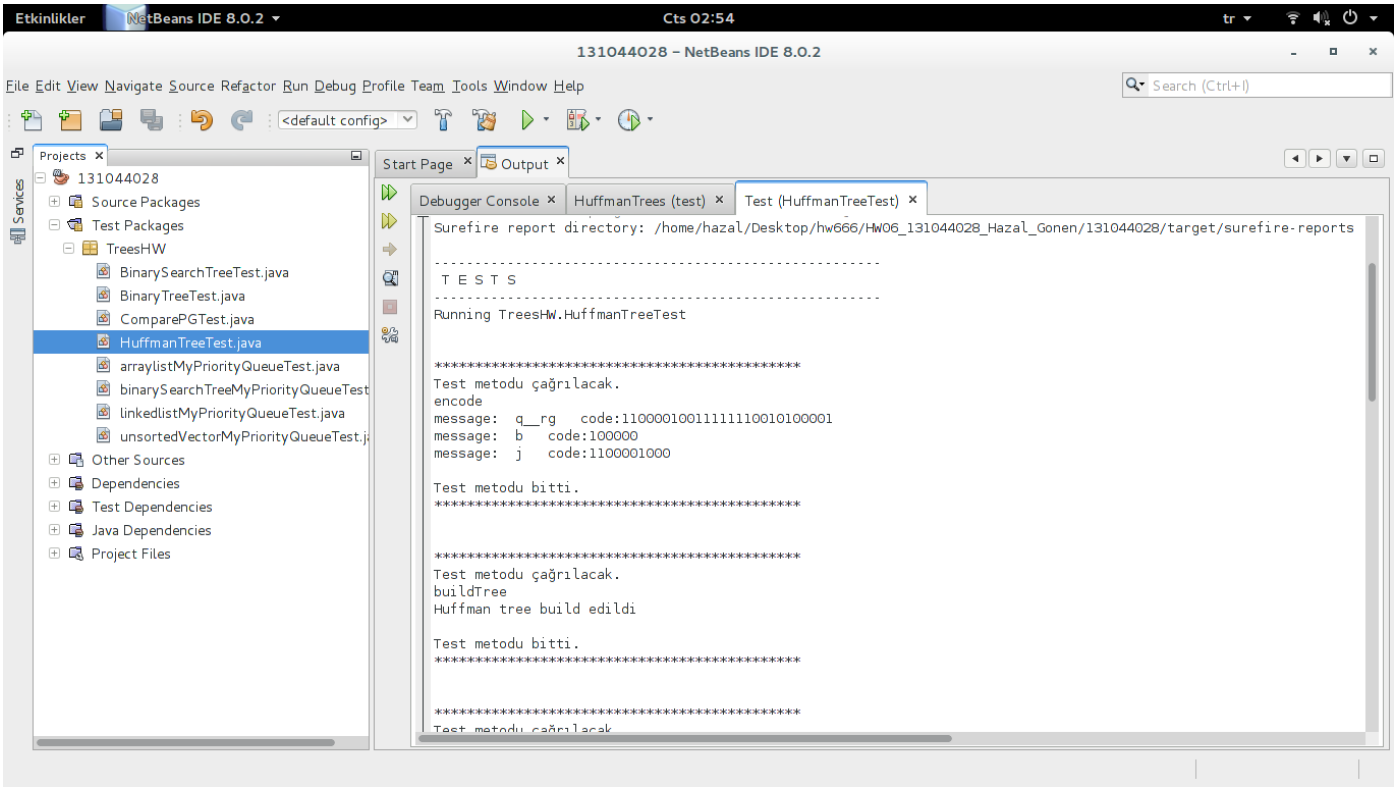
Bu treeyi kullanabilmek icin harfler ve onlarin ne kadar sik kullanildiklarini gosteren degerler olmali. her harf 1 ve 0larla temsil edilerek sikistirilmasini saglar.

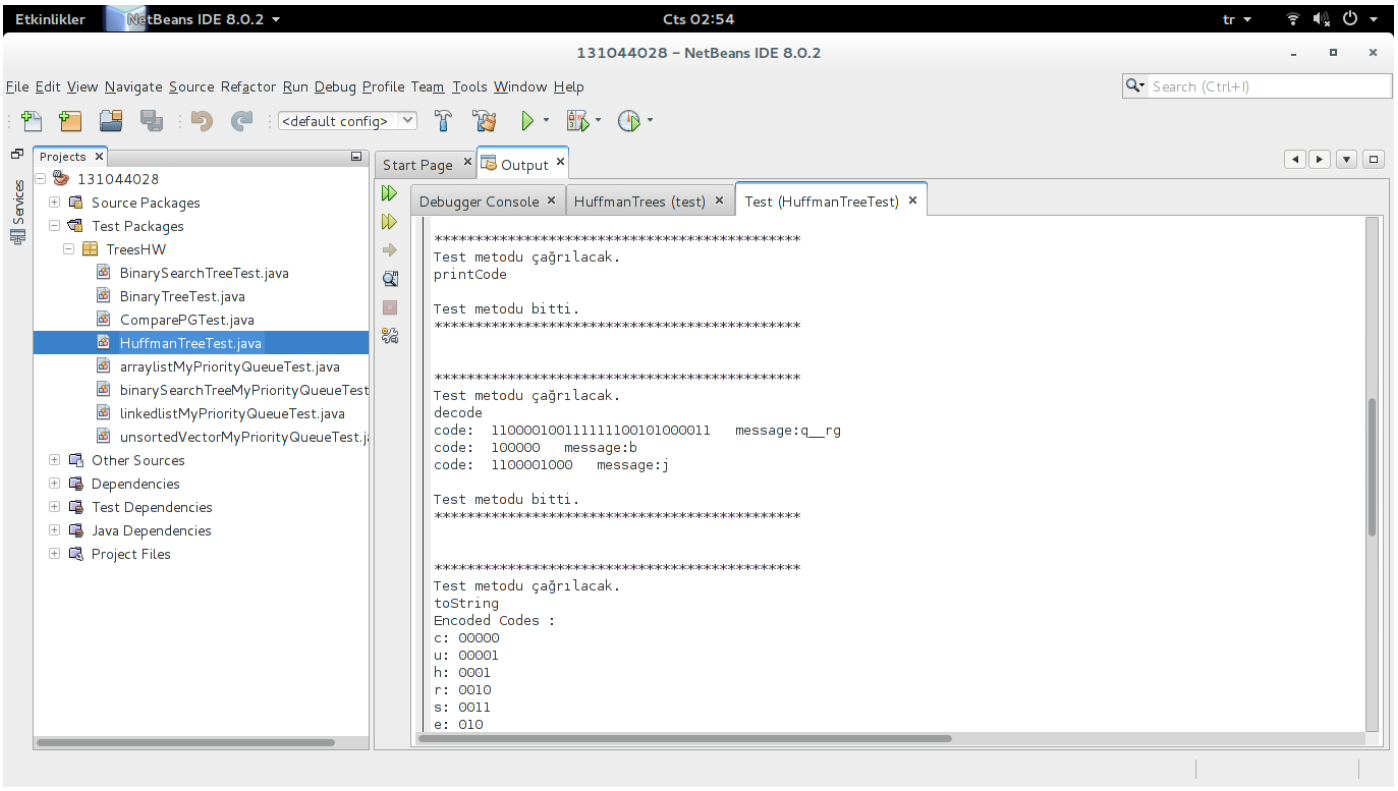
Yani ilk olarak bu harfler ve sikliklarini agaca buildTree ile build etmem lazim daha sonra build edilen agactan yararlanarak encode ve decode yapabilirim

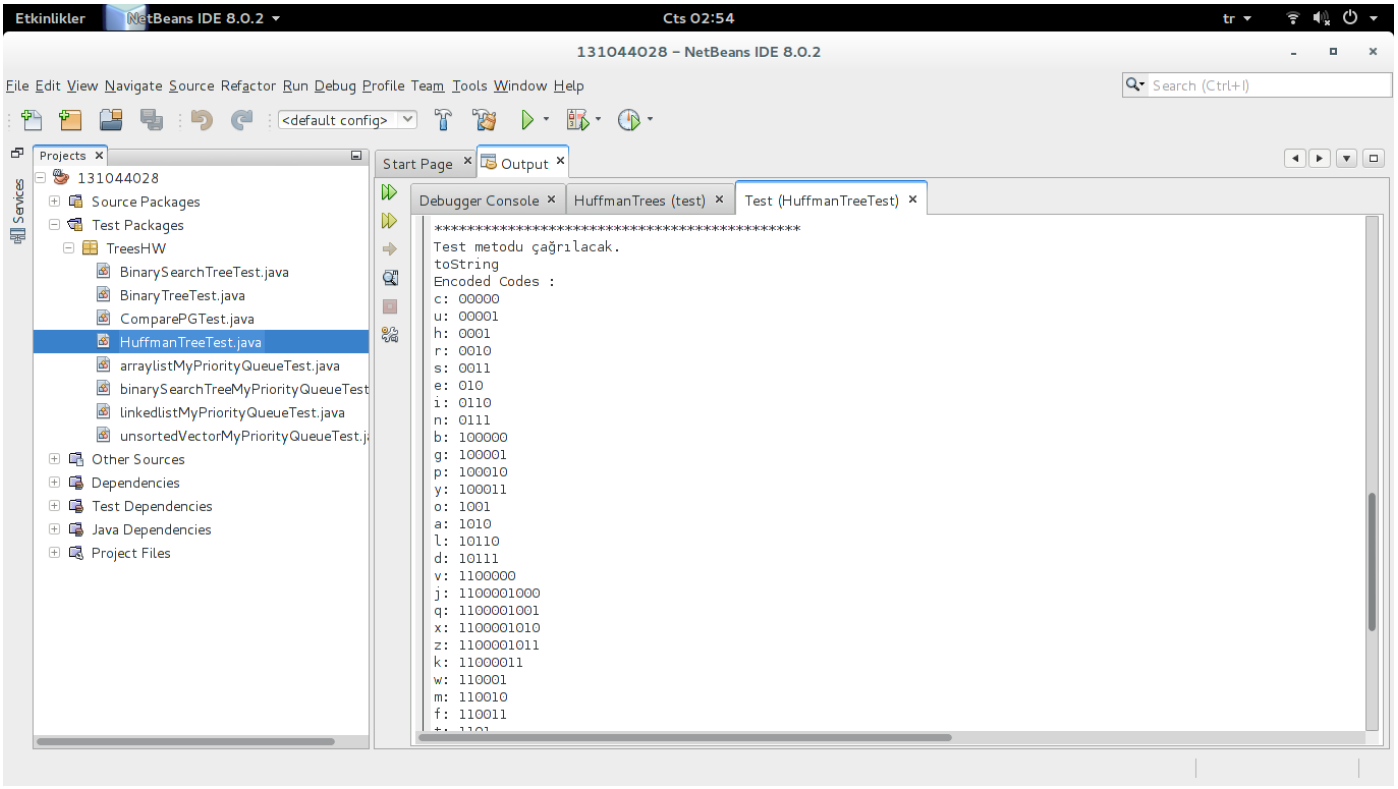
HUFFMANTREE Classı için PROBLEM COZUM YONTEMLERI:

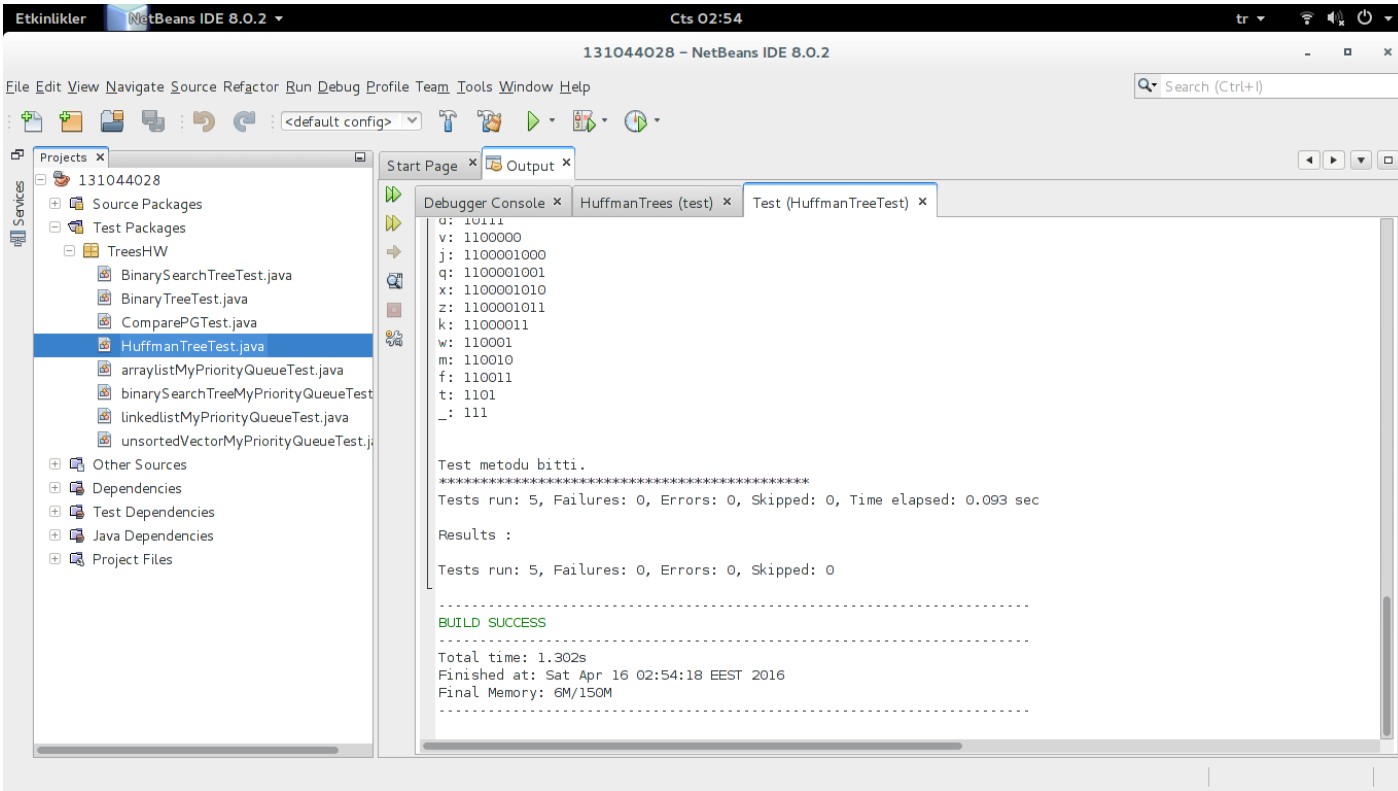
encode metodunu yazarken string ve tree parametre olarak aldim ve stringin her harfini tek tek recursive bir helper metoduna gonderdim bu helper metodu agaci tek tek gezerek harfin hangi dallar uzerinden gittigini bulup ona gore eger soldaysa 0 sagdaysa 1 ekleyerek kodlanmis halini return eder. harflerin hepsi leaf yani köktür bu sayede harflere geldigimi köke geldiğimde anlayabilirim

HuffmanTree Classinin J-UNIT ekran goruntuleri:






Etkinlikler
NetBeans IDE 8.0.2
Cts 02:54
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>
Projects
Services
131044028
Source Packages
Test Packages
TreesHW
BinarySearchTreeTest.java
BinaryTreeTest.java
ComparePGTest.java
HuffmanTreeTest.java
arraylistMyPriorityQueueTest.java
binarySearchTreeMyPriorityQueueTest
linkedlistMyPriorityQueueTest.java
unsortedVectorMyPriorityQueueTest.j
Other Sources
Dependencies
Test Dependencies
Java Dependencies
Project Files
Start Page
Output
Debugger Console
HuffmanTrees (test)
Test (HuffmanTreeTest)
****
Test metodu çağrılacak.
toString
Encoded Codes :
c: 00000
u: 00001
h: 0001
r: 0010
s: 0011
e: 010
i: 0110
n: 0111
b: 100000
g: 100001
p: 100010
y: 100011
o: 1001
a: 1010
l: 10110
d: 10111
v: 1100000
j: 1100001000
q: 1100001001
x: 1100001010
z: 1100001011
k: 11000011
w: 110001
m: 110010
f: 110011


Etkinlikler
NetBeans IDE 8.0.2
Cts 02:54
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>
Projects
Services
131044028
Source Packages
Test Packages
TreesHW
BinarySearchTreeTest.java
BinaryTreeTest.java
ComparePGTest.java
HuffmanTreeTest.java
arraylistMyPriorityQueueTest.java
binarySearchTreeMyPriorityQueueTest
linkedlistMyPriorityQueueTest.java
unsortedVectorMyPriorityQueueTest.j
Other Sources
Dependencies
Test Dependencies
Java Dependencies
Project Files
Start Page
Output
Debugger Console
HuffmanTrees (test)
Test (HuffmanTreeTest)
v: 1100000
j: 1100001000
q: 1100001001
x: 1100001010
z: 1100001011
k: 11000011
w: 110001
m: 110010
f: 110011
t: 1101
_: 111
Test metodu bitti.
****
Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.093 sec
Results :
Tests run: 5, Failures: 0, Errors: 0, Skipped: 0
BUILD SUCCESS
Total time: 1.302s
Finished at: Sat Apr 16 02:54:18 EEST 2016
Final Memory: 6M/150M

BinarySearchTree Class için SISTEM GEREKSINIMLERI:

Bu class icin iterator classi yazmam istendi. Iterator classinin icinde next ve hasNext metodlarini yazdim.

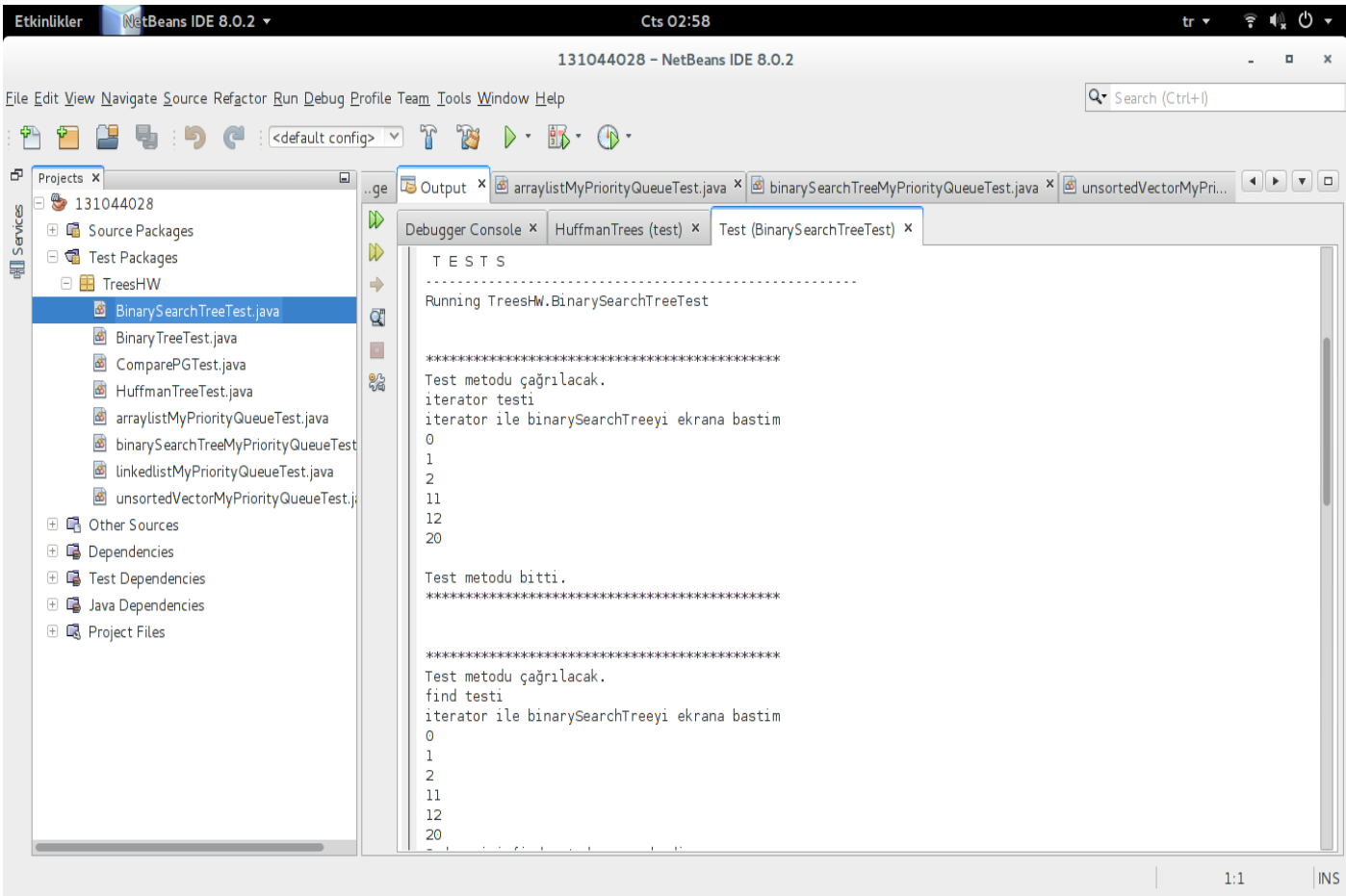
İlk olarak BinarySearchTree objesi olusturulmalı daha sonra bu objeye elemanlar add metodu ile eklenebilir. Herhangi bir gezme islemi icin Iterator objesi yapilip next ve hasNext ile gezilebilir. Ayrıca delete, find metodlari da vardir.

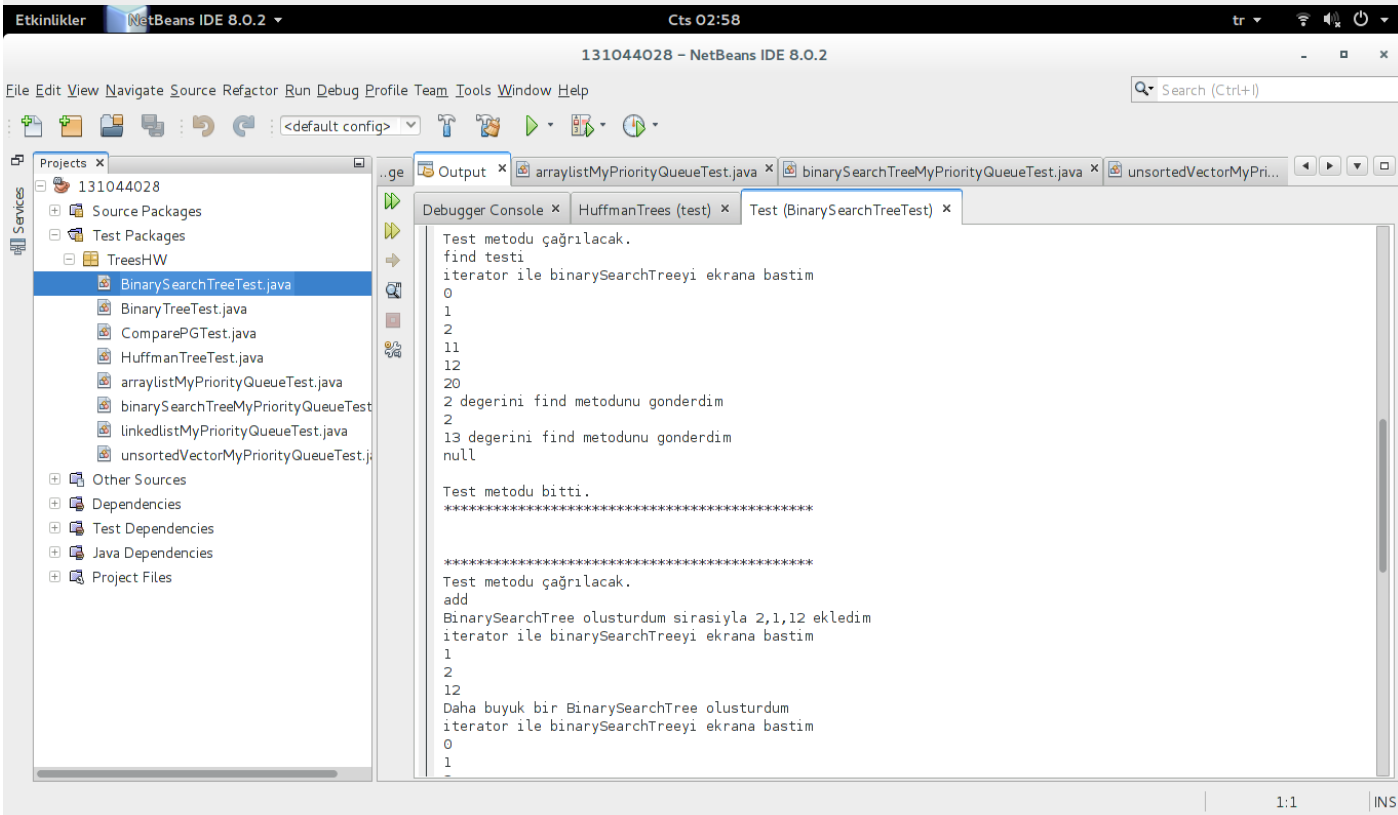
BinarySearchTree Class için PROBLEM COZUM YONTEMLERI:

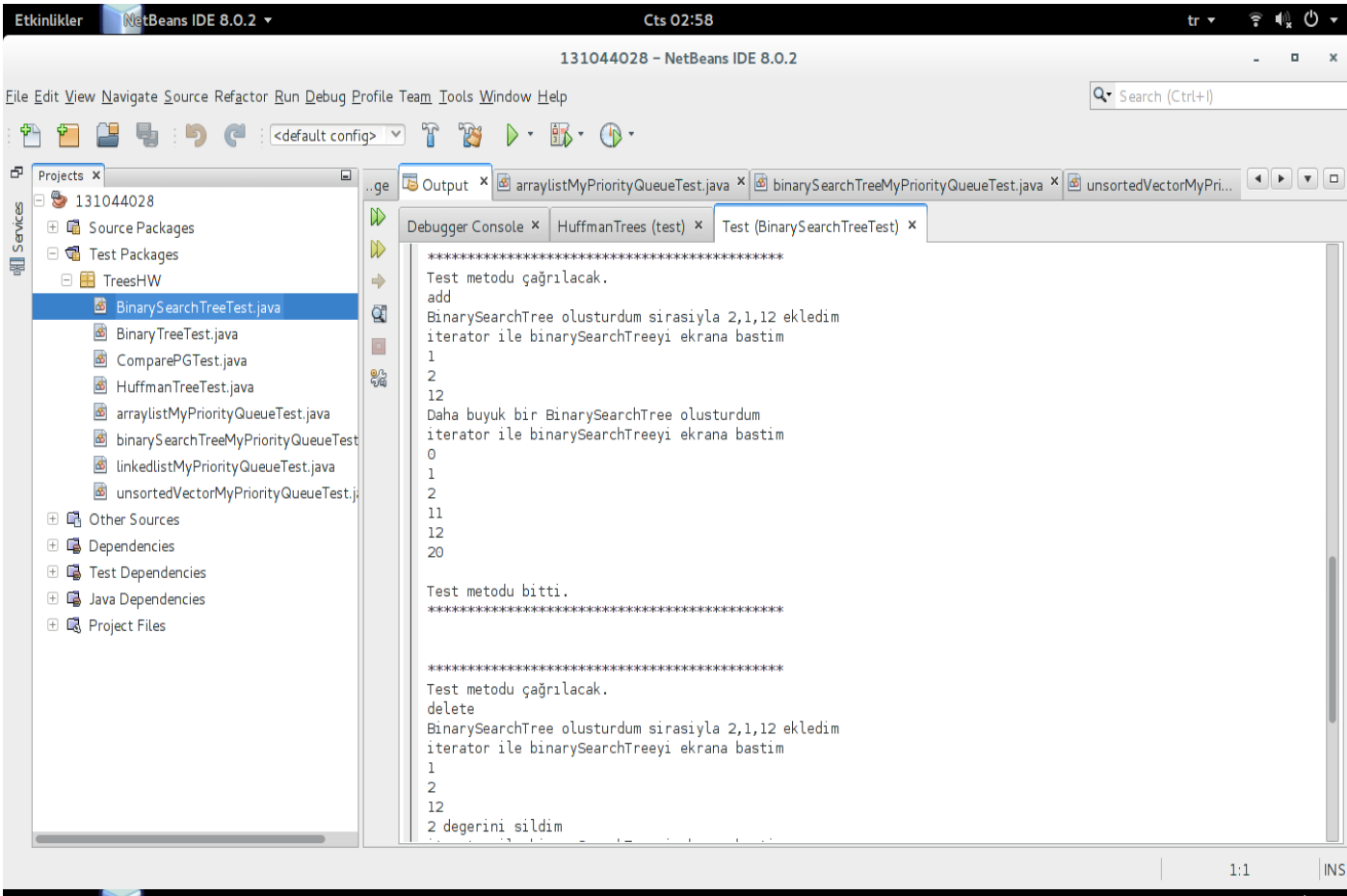
Bu classa iterator yazmam istendi ve ben myIterator adiyla private inner class yazdim bu class ayrica javanin Iterator classini da implements ediyor ve next,hasnext metodlarini override yazmam gerekiyor.
Iteratorle gezebilmek icin ilk olarak butun agaci linkedlist tipinde vi veriyapisina ekledim çünkü linkedlistte dolasmak daha kolay olacaktı linkedliste doldurma islemini yapan yardimci bir fillLinkedList metodu yazdim. bu metod recursive olarak sol ve sağ dallari da gezerek butun agaci sirasiyla listeye ekliyor ve next dedigi zaman ilk elemani removeFirst metodunu çağırarak return ediyor. hasNext metodu ise linkedlistin boş olup olmadığına bakıyor eger boşsa 0 değilse 1 donduruyor.

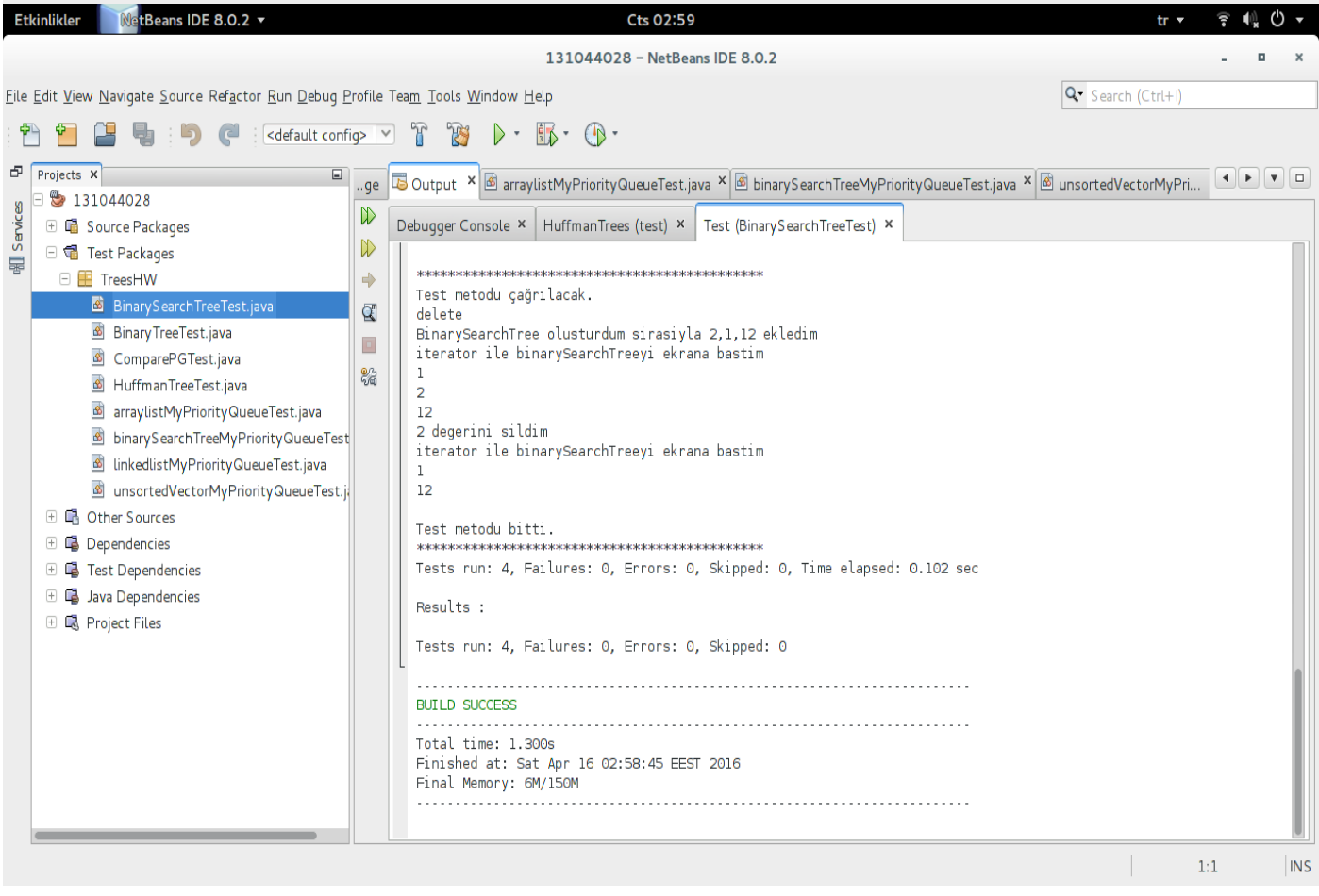
Bu inner classa dışarıdan erişmeyi saglayan public Iterator metodu yazdım ve bu sayede objeye ulaşan herkes iterator yardimiyla butun treeyi dolasabilir.

BinarySearchTree Classinin J-UNIT ekran goruntuleri:






Etkinlikler
NetBeans IDE 8.0.2
Cts 02:58
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>
Projects
131044028
Source Packages
Test Packages
TreesHW
BinarySearchTreeTest.java
BinaryTreeTest.java
ComparePGTest.java
HuffmanTreeTest.java
arraylistMyPriorityQueueTest.java
binarySearchTreeMyPriorityQueueTest
linkedlistMyPriorityQueueTest.java
unsortedVectorMyPriorityQueueTest.j
Other Sources
Dependencies
Test Dependencies
Java Dependencies
Project Files
Services
Output
arraylistMyPriorityQueueTest.java
binarySearchTreeMyPriorityQueueTest.java
unsortedVectorMyPri...
Debugger Console
HuffmanTrees (test)
Test (BinarySearchTreeTest)
****
Test metodu çağrılacak.
add
BinarySearchTree olusturdum sirasiyla 2,1,12 ekledim
iterator ile binarySearchTreeyi ekrana bastim
1
2
12
Daha buyuk bir BinarySearchTree olusturdum
iterator ile binarySearchTreeyi ekrana bastim
0
1
2
11
12
20
Test metodu bitti.
****
****
Test metodu çağrılacak.
delete
BinarySearchTree olusturdum sirasiyla 2,1,12 ekledim
iterator ile binarySearchTreeyi ekrana bastim
1
2
12
2 degerini sildim
1:1
INS


Etkinlikler
NetBeans IDE 8.0.2
Cts 02:59
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>
Projects
131044028
Source Packages
Test Packages
TreesHW
BinarySearchTreeTest.java
BinaryTreeTest.java
ComparePGTest.java
HuffmanTreeTest.java
arraylistMyPriorityQueueTest.java
binarySearchTreeMyPriorityQueueTest
linkedlistMyPriorityQueueTest.java
unsortedVectorMyPriorityQueueTest.j
Other Sources
Dependencies
Test Dependencies
Java Dependencies
Project Files
Services
Output
arraylistMyPriorityQueueTest.java
binarySearchTreeMyPriorityQueueTest.java
unsortedVectorMyPri...
Debugger Console
HuffmanTrees (test)
Test (BinarySearchTreeTest)
****
Test metodu çağrılacak.
delete
BinarySearchTree olusturdum sirasiyla 2,1,12 ekledim
iterator ile binarySearchTreeyi ekrana bastim
1
2
12
2 degerini sildim
iterator ile binarySearchTreeyi ekrana bastim
1
12
Test metodu bitti.
****
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.102 sec
Results :
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0
BUILD SUCCESS
Total time: 1.300s
Finished at: Sat Apr 16 02:58:45 EEST 2016
Final Memory: 6M/150M
1:1
INS

PriorityQueue için SISTEM GEREKSINIMLERI:

PriorityQueue'yu 4 farklı şekilde implement ettim. Bunlar arrayList, linkedList, BinarySearchTree, unsortedVector.

Hepsinde olması gereken 4 metod var bunlar: enqueue,dequeue,size,isEmpty.

PriorityQueue olan bu 4(arrayList,linkedList..) objeden biri olusturulabilirç Ve bu objeler bu 4 metodu yapabilir.

Onemli olan bu partta enqueue dequeue islemlerinin ne kadar surdugu. Bu bilgiler tabloda yer alıyor.

PriorityQueue Class için PROBLEM COZUM YONTEMLERI:

Bu parti yaparken 4 farkli class yazmaliyim arrayList, linkedList, BinarySearchTree, unsortedVector icin. Ve hepsinin 4 metoda sahip olacağını bildiğim için priorityqueue interface ile bunları birleştirdim.

4 metoddan bahsedecek olursam enqueue islemi ya da dequeue islemi yapildiktan sonra bile bu treenin priority özelliğini kaybetmemesi gerekiyor. priority önceliği olan elemanlarin ilk çıkmasi ve ekleme yapilirken önceliğine göre eklenmesi demektir.

ArrayList için enqueue ve dequeue yaparken kitaptaki koddan yardim aldim. parent child ilişkilerini barındırarak arrayliste ekleme yapildi ve bu sayede ekleme çıkarma en kötü logn zamana düştü. size metodu direk arrayListin size metodu ve isEmpty ise direk arrayListin isEmpty metoduyla yaptım. Ek olarak print metodu ekledim kolay görebilmek için. Constructorlar 3 çeşit ilki default direk arrayList objesi oluşturuyor diğeri comparator parametresi alıyor benim ComparatorPG isimli bir classim var ve bu class en kucuk olana en buyuk önceliği veriyor yani agaçtan ilk çıkacak olan listenin en küçük elemanı olmali her zaman. Birde capacity ve comparator alan constructor var bu ise capacity kadar yer ayırıyor arrayListe.

LinkedList için enqueue yaparken iterator yardimiyla ilk olarak butun agacı dolaşıyor ve agacı kucukten buyuge siralamis oluyor. Boylece dequeue yapmak çok kolay direk ilk elemani removeFirst metodu ile siliyor ve return ediyor. size metodu direk LinkedList size metodu ve isEmpty ise direk LinkedList isEmpty metoduyla yaptım. Ek olarak print metodu ekledim kolay görebilmek için bu metodda iterator yardimiyla butun listeyi ekrana basiyor. Constructorlar 3 çeşit ilki default direk LinkedList objesi oluşturuyor diğeri comparator parametresi alıyor benim ComparatorPG isimli az önce bahsettiğim class. Birde capacity ve comparator alan constructor var bu ise capacity kadar yer ayırmali LinkedListe fakat LinkedList böyle bir şey yapmasına gerek yok bu sebeple bu parametreyi kullanmadım.

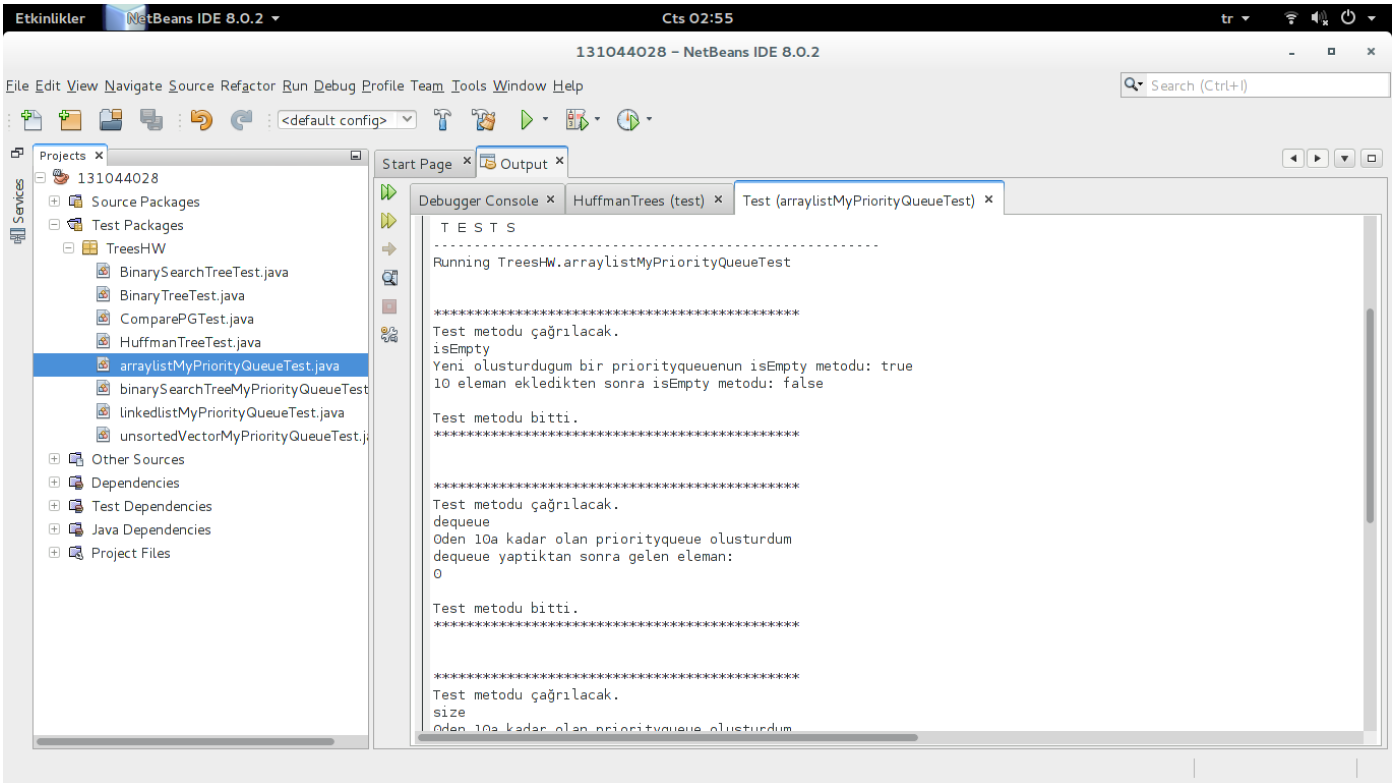
UnsortedVector için enqueue yaparken direk olarak ekledim çünkü sıralama istemiyor ve böylece eklenen eleman en sona koyuluyor. dequeue yapmak biraz daha zor çünkü önceliği en fazla olanı

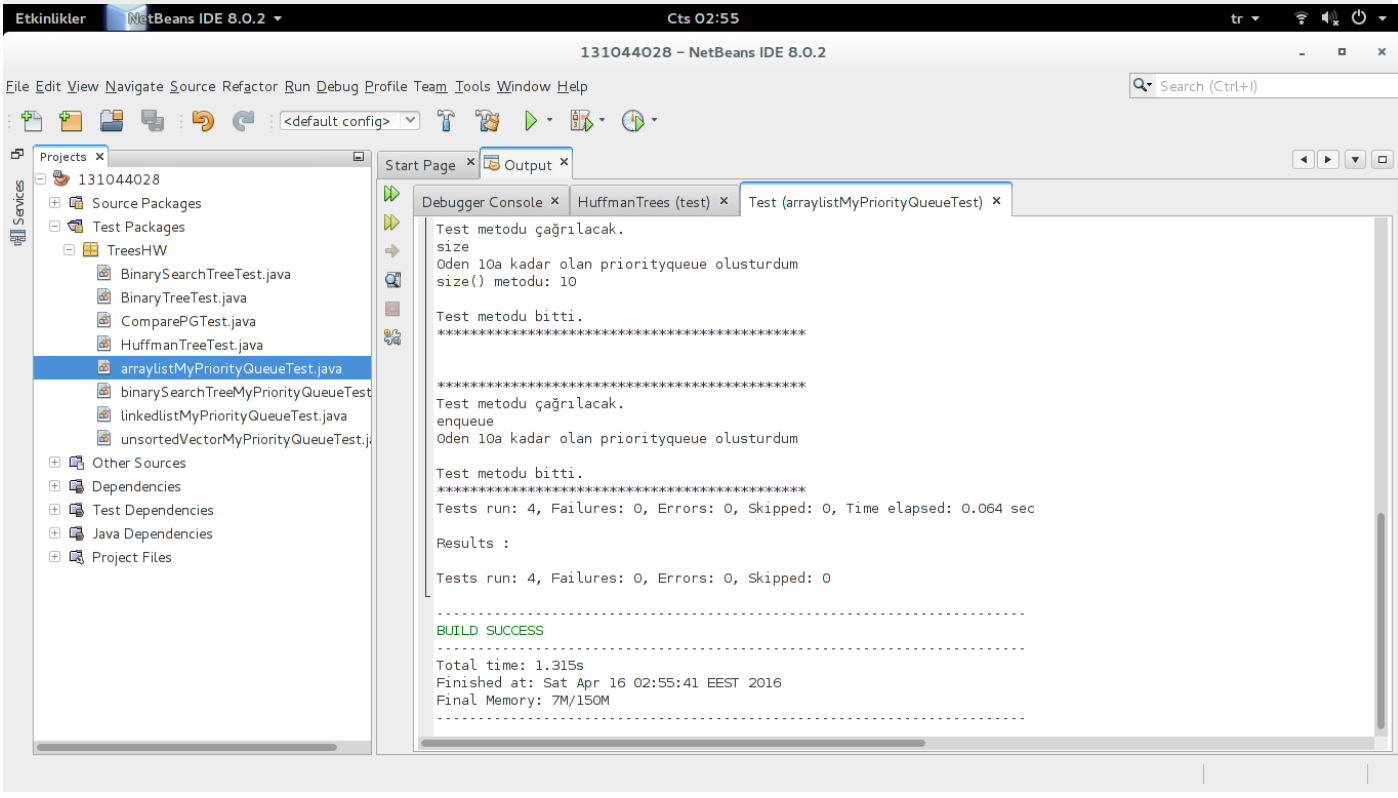
bulmak için butun listeyi gezmem gerekiyor ve önceliği en fazla olanı çıkarıp geri kalan listeyi kaydırmam gerekiyor ve size'ı da bir azaltmalıyım. size metodu direk Vector size metodu ve isEmpty ise direk Vector isEmpty metoduyla yaptım. Ek olarak print metodu ekledim kolay görebilmek için bu metodda donguyle butun listeyi ekrana basiyor. Constructorlar 3 çeşit ilki default direk UnsortedVector objesi oluşturuyor diğeri comparator parametresi alıyor benim ComparatorPG isimli az önce bahsettiğim class. Birde capacity ve comparator alan constructor var bu ise capacity kadar yer ayırmali.

BinarySearchTree için enqueue yaparken direk BinarySearchTreenin metodlarini kullandım. enqueue için add metodunu çağırdım ve direk öncelik sırasına gore ekleme yapildi. dequeue için delete metodunu çağırdım fakat delete işlemi yapilirken her zaman en üstteki data silinmeli çünkü her zaman en küçük olan o. size metodu direk iterator yardımıyla buldum ve isEmpty ise root null mı diye baktım. Constructorlar 3 çeşit ilki default direk BinarySearchTree objesi oluşturuyor diğeri comparator parametresi alıyor benim ComparatorPG isimli az önce bahsettiğim class. Birde capacity ve comparator alan constructor var bu ise capacity kadar yer ayırmali fakat BinarySearchTree böyle bir şey yapmasına gerek yok bu sebeple bu parametreyi kullanmadım.

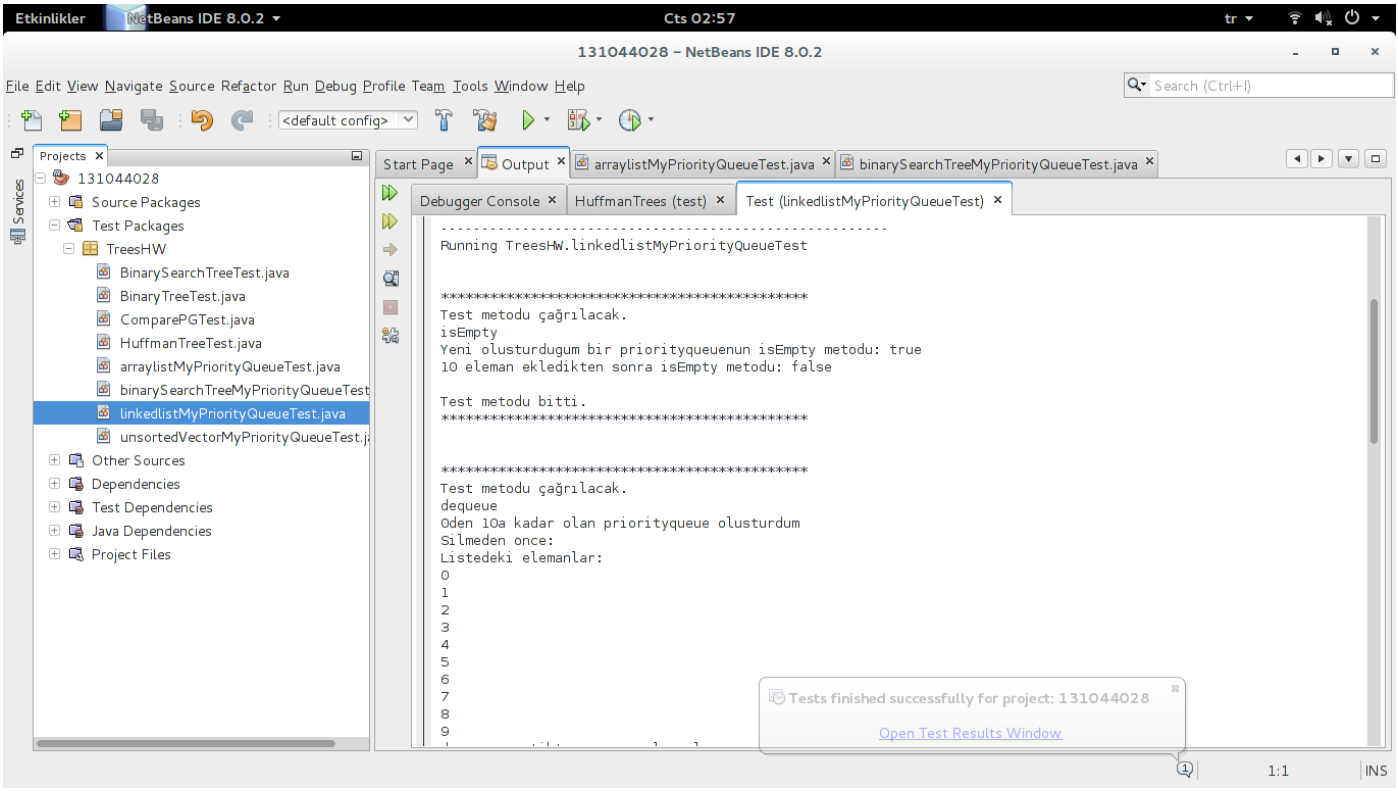
PriorityQueue Classinin J-UNIT ekran goruntuleri:

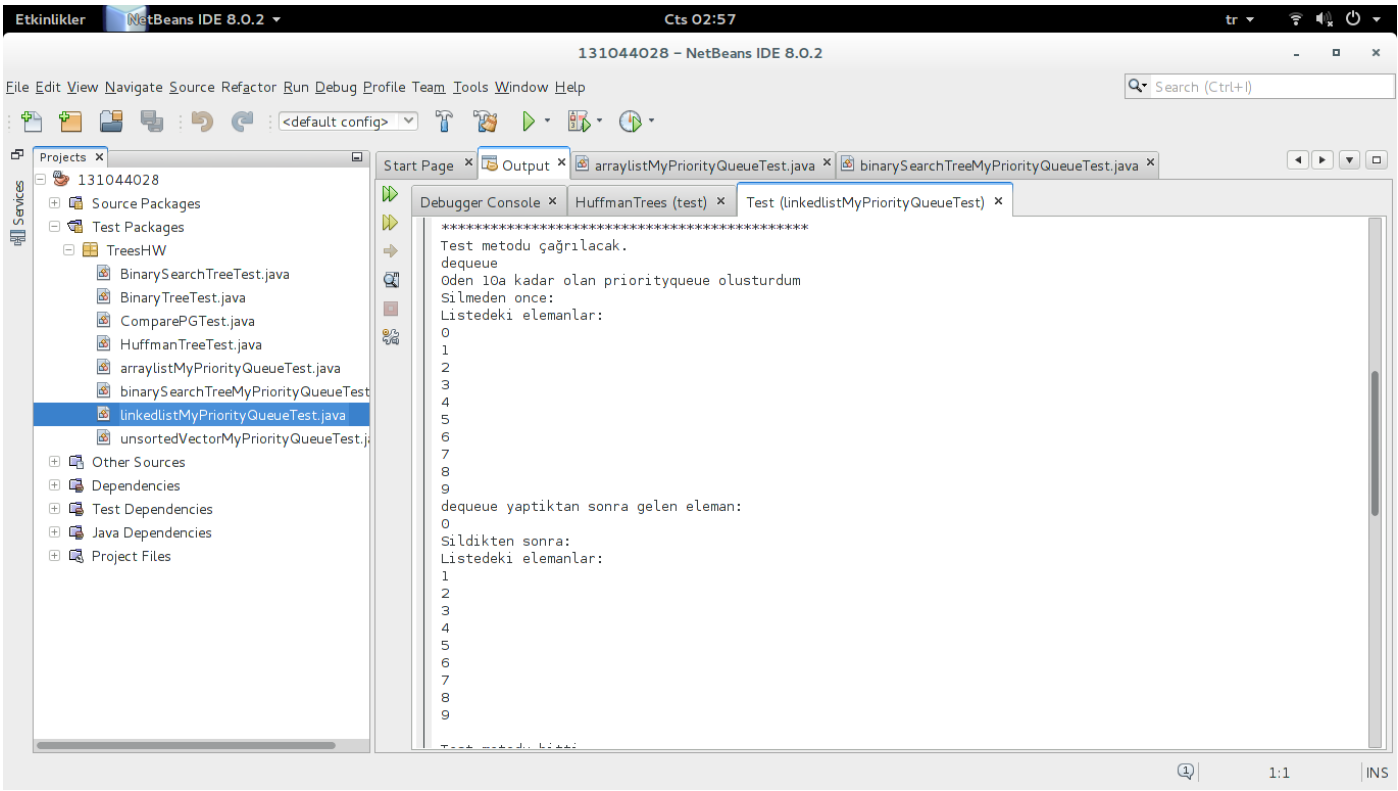
ArrayList için:

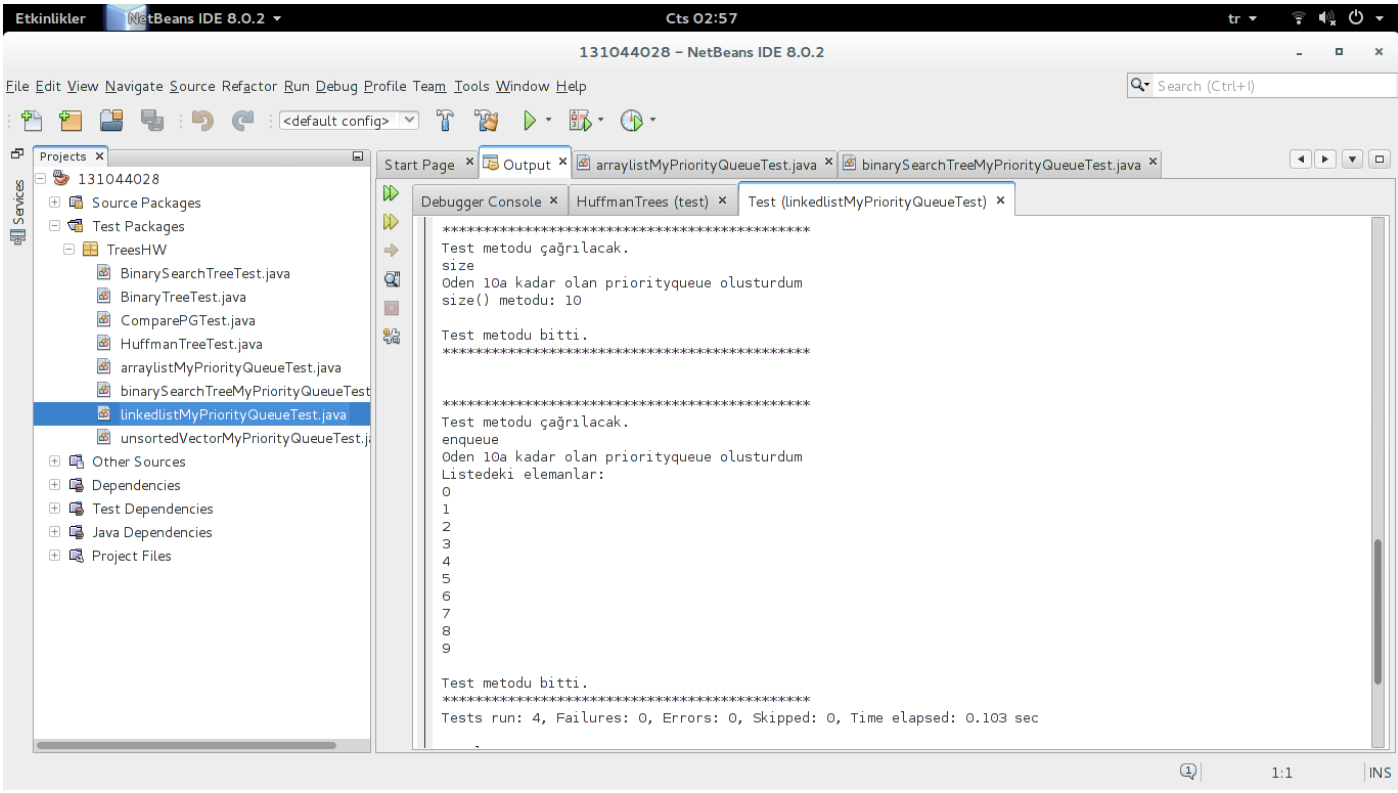




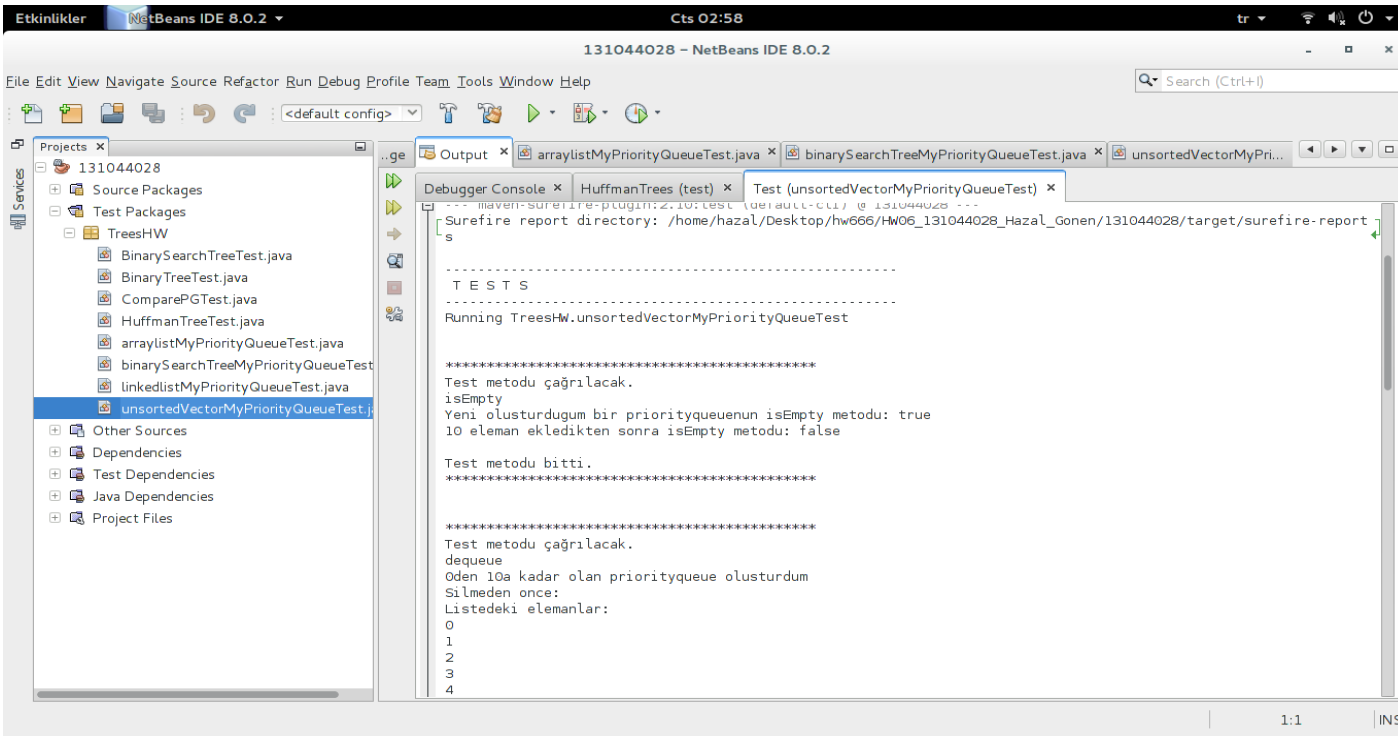
LinkedList için

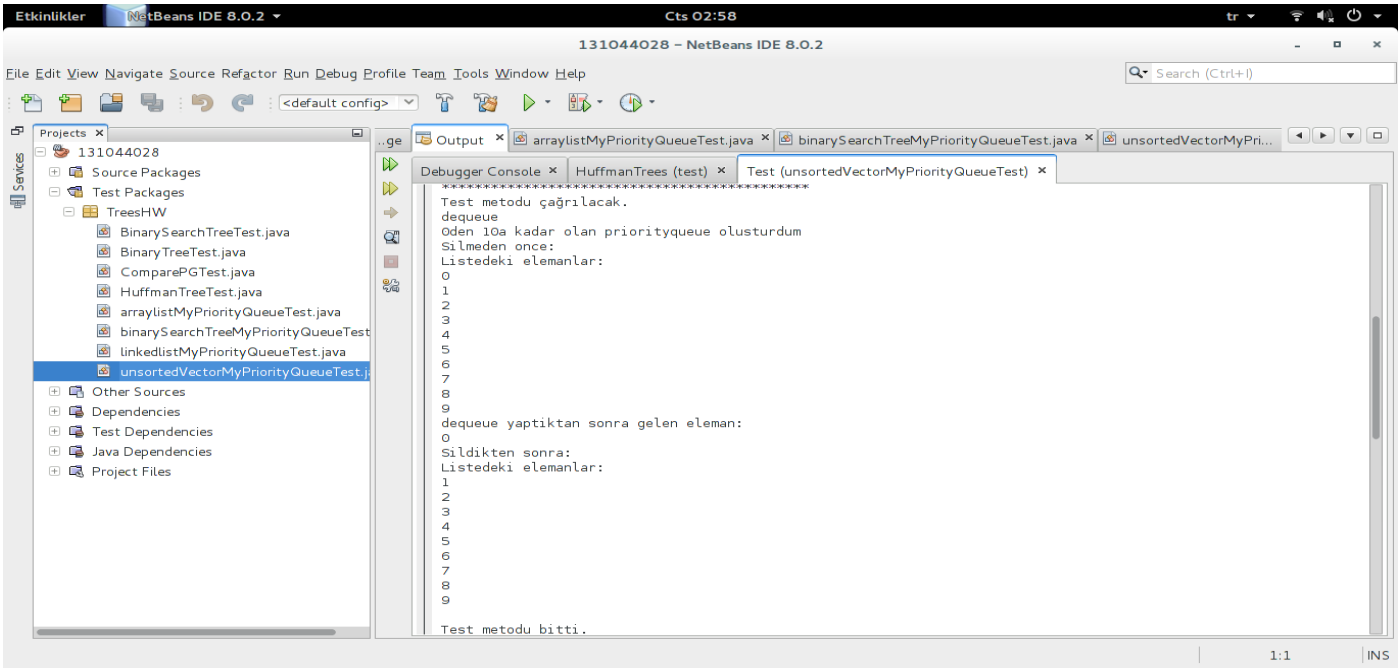


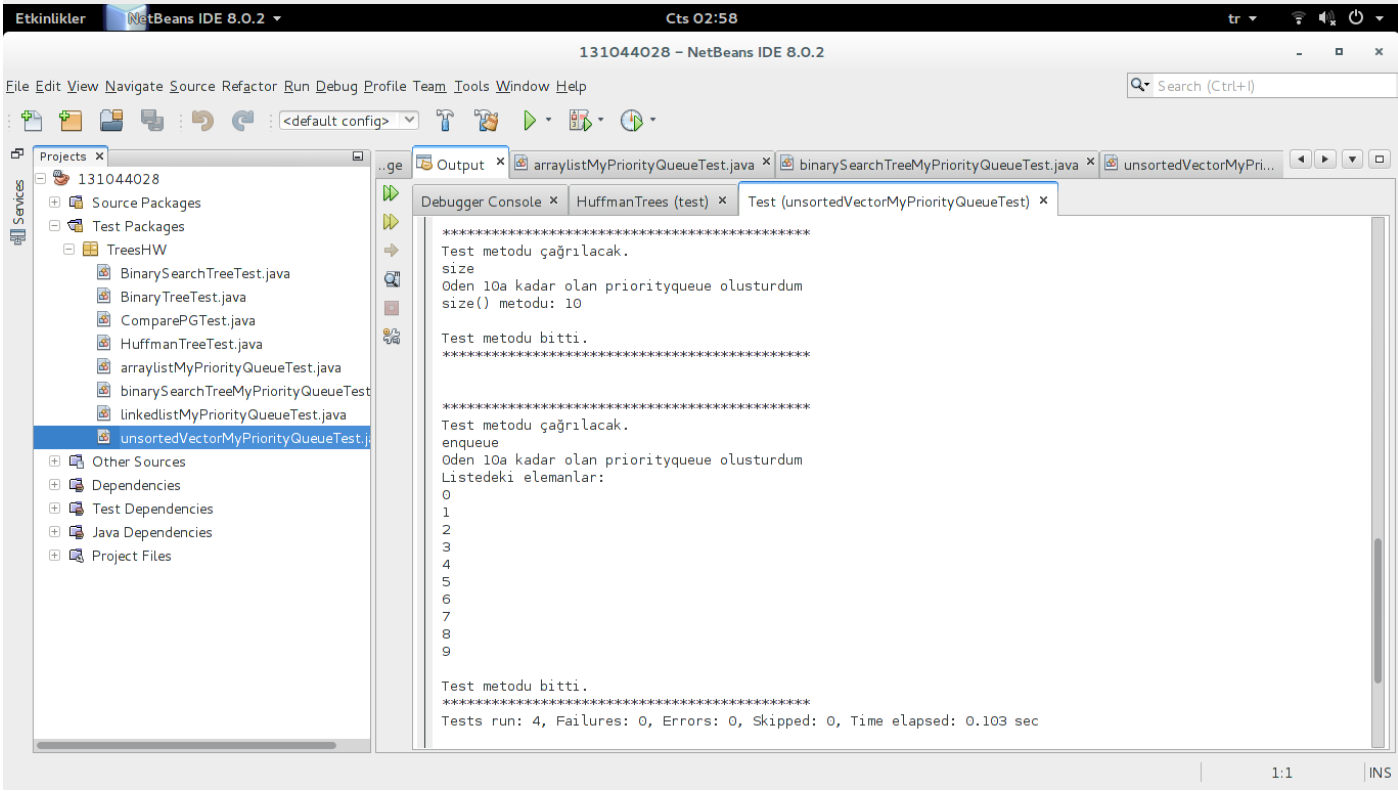




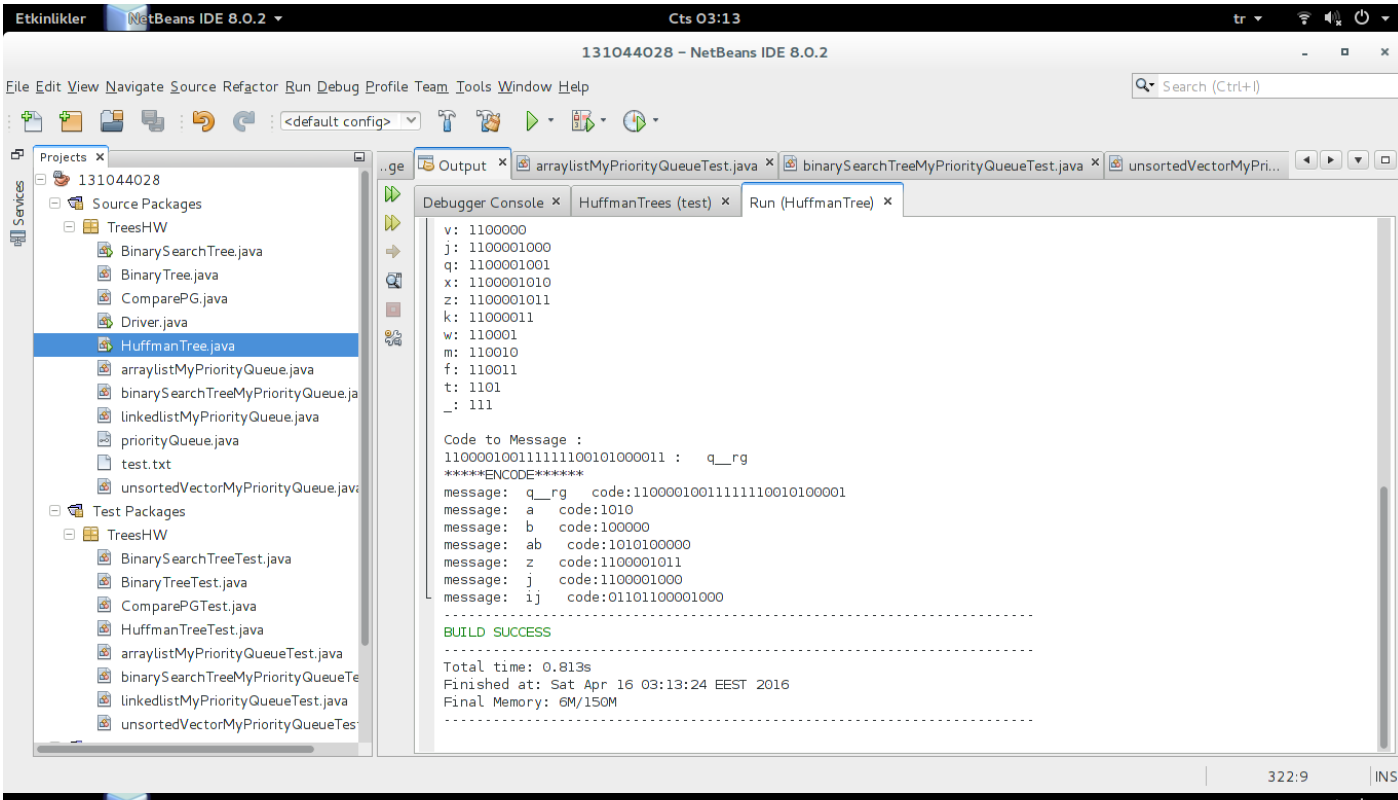
UnsortedVector için:

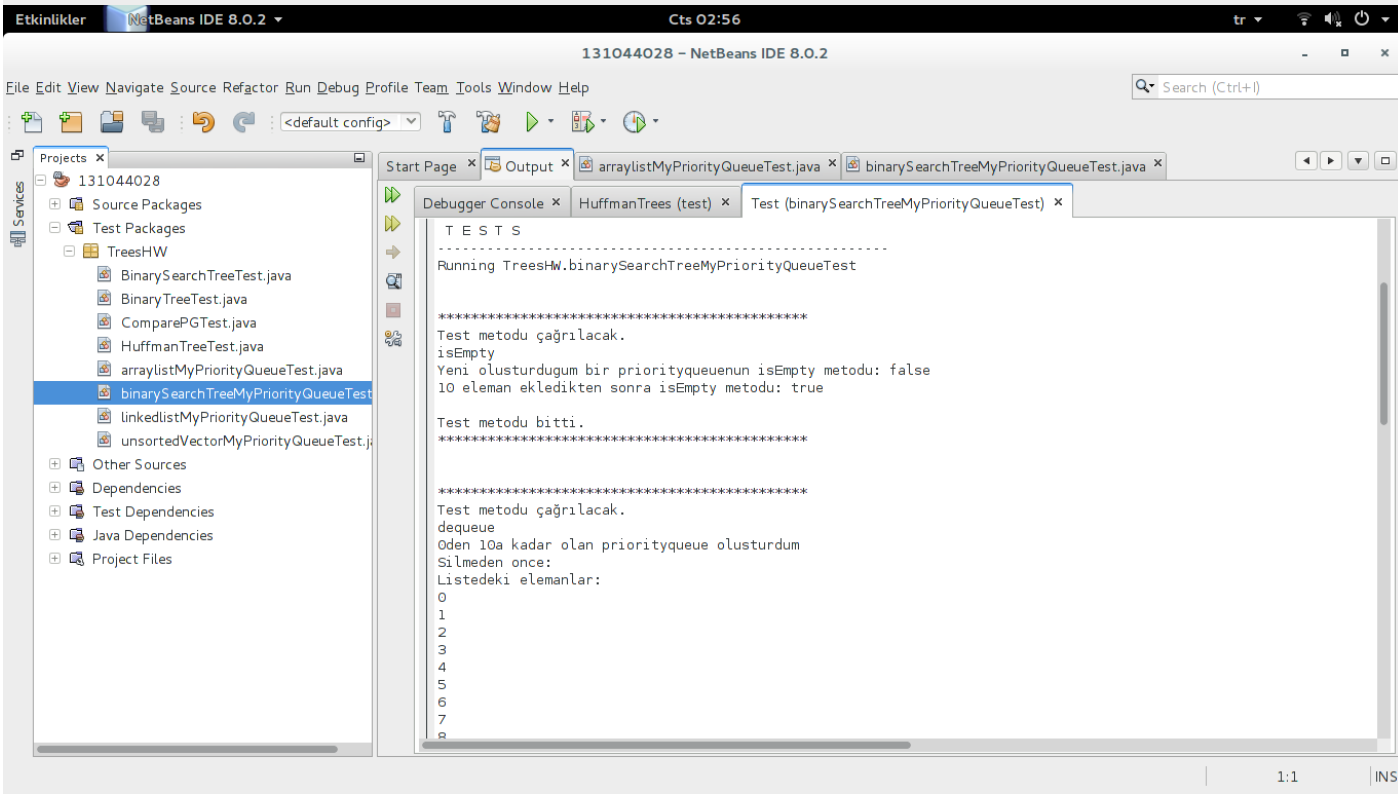


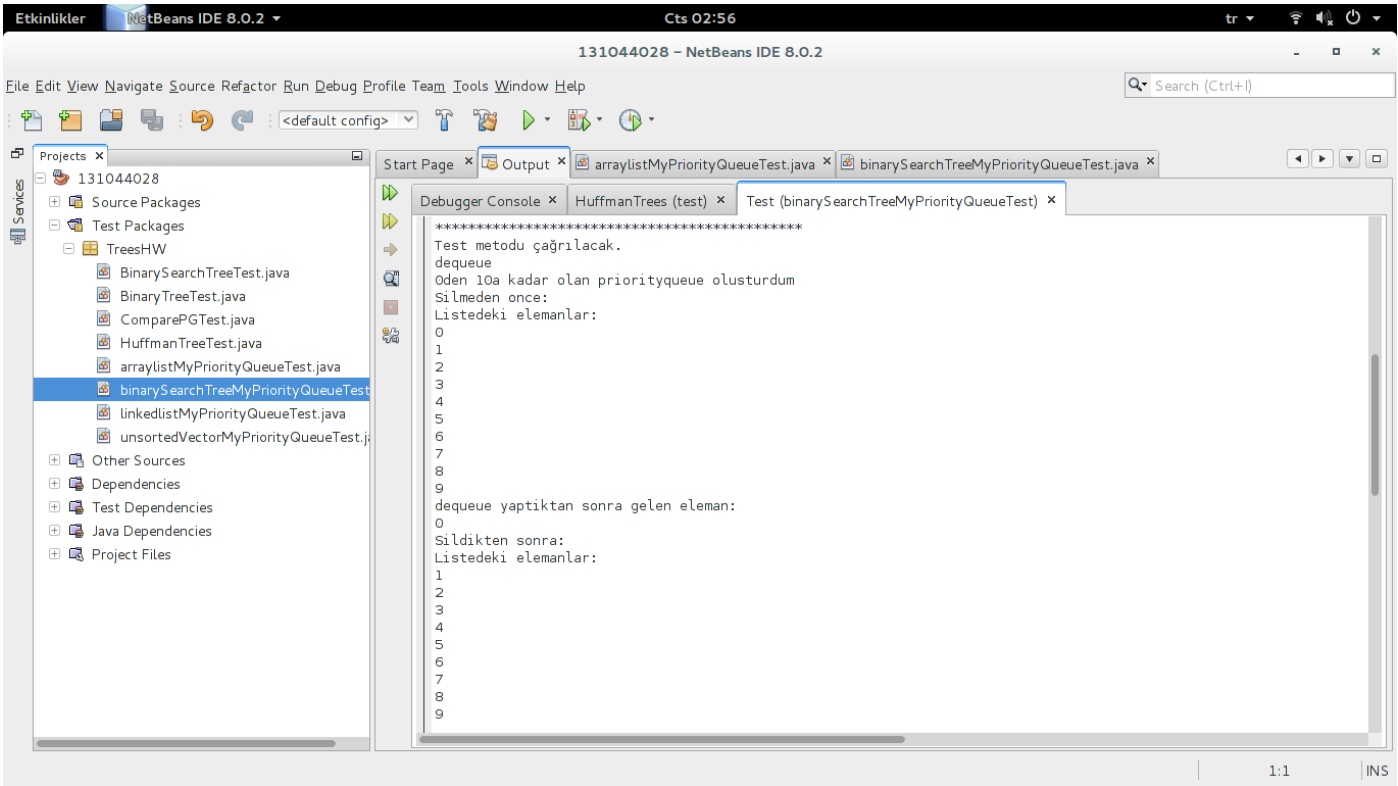

Etkinlikler
NetBeans IDE 8.0.2
Cts 02:58
tr
131044028 – NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Projects
Services
131044028
Source Packages
Test Packages
TreesHW
BinarySearchTreeTest.java
BinaryTreeTest.java
ComparePGTest.java
HuffmanTreeTest.java
arraylistMyPriorityQueueTest.java
binarySearchTreeMyPriorityQueueTest
linkedlistMyPriorityQueueTest.java
unsortedVectorMyPriorityQueueTest.j
Other Sources
Dependencies
Test Dependencies
Java Dependencies
Project Files
Output
arraylistMyPriorityQueueTest.java
binarySearchTreeMyPriorityQueueTest.java
unsortedVectorMyPri...
Debugger Console
HuffmanTrees (test)
Test (unsortedVectorMyPriorityQueueTest)
****
Test metodu çağrılacak.
dequeue
Oden 10a kadar olan priorityqueue olusturdum
Silmeden once:
Listedeki elemanlar:
0
1
2
3
4
5
6
7
8
9
dequeue yaptiktan sonra gelen eleman:
0
Sildikten sonra:
Listedeki elemanlar:
1
2
3
4
5
6
7
8
9
Test metodu bitti.
1:1
INS

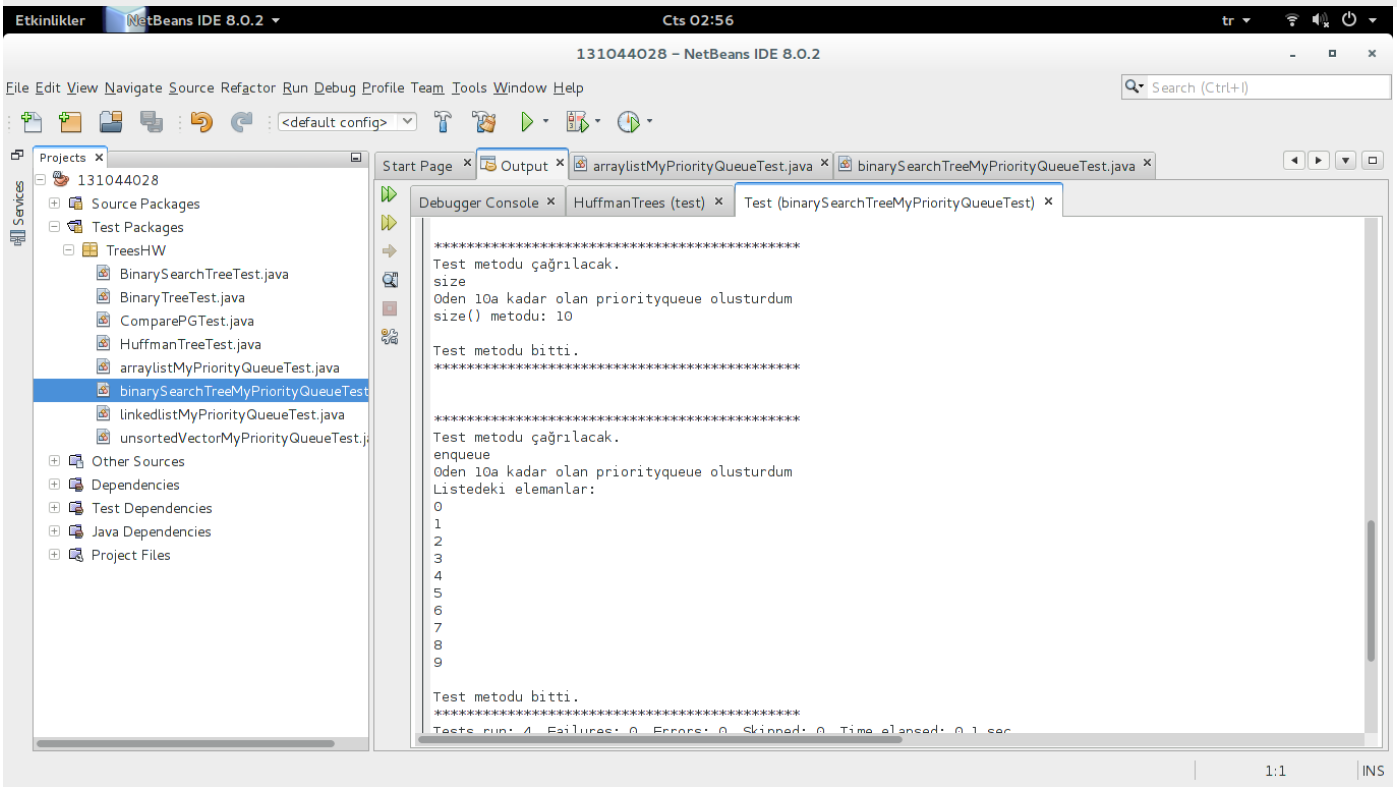

Etkinlikler
NetBeans IDE 8.0.2
Cts 02:58
tr
131044028 – NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Projects
Services
131044028
Source Packages
Test Packages
TreesHW
BinarySearchTreeTest.java
BinaryTreeTest.java
ComparePGTest.java
HuffmanTreeTest.java
arraylistMyPriorityQueueTest.java
binarySearchTreeMyPriorityQueueTest
linkedlistMyPriorityQueueTest.java
unsortedVectorMyPriorityQueueTest.j
Other Sources
Dependencies
Test Dependencies
Java Dependencies
Project Files
Output
arraylistMyPriorityQueueTest.java
binarySearchTreeMyPriorityQueueTest.java
unsortedVectorMyPri...
Debugger Console
HuffmanTrees (test)
Test (unsortedVectorMyPriorityQueueTest)
****
Test metodu çağrılacak.
size
Oden 10a kadar olan priorityqueue olusturdum
size() metodu: 10
Test metodu bitti.
****
****
Test metodu çağrılacak.
enqueue
Oden 10a kadar olan priorityqueue olusturdum
Listedeki elemanlar:
0
1
2
3
4
5
6
7
8
9
Test metodu bitti.
****
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.103 sec
1:1
INS

BinarySearchTree için:



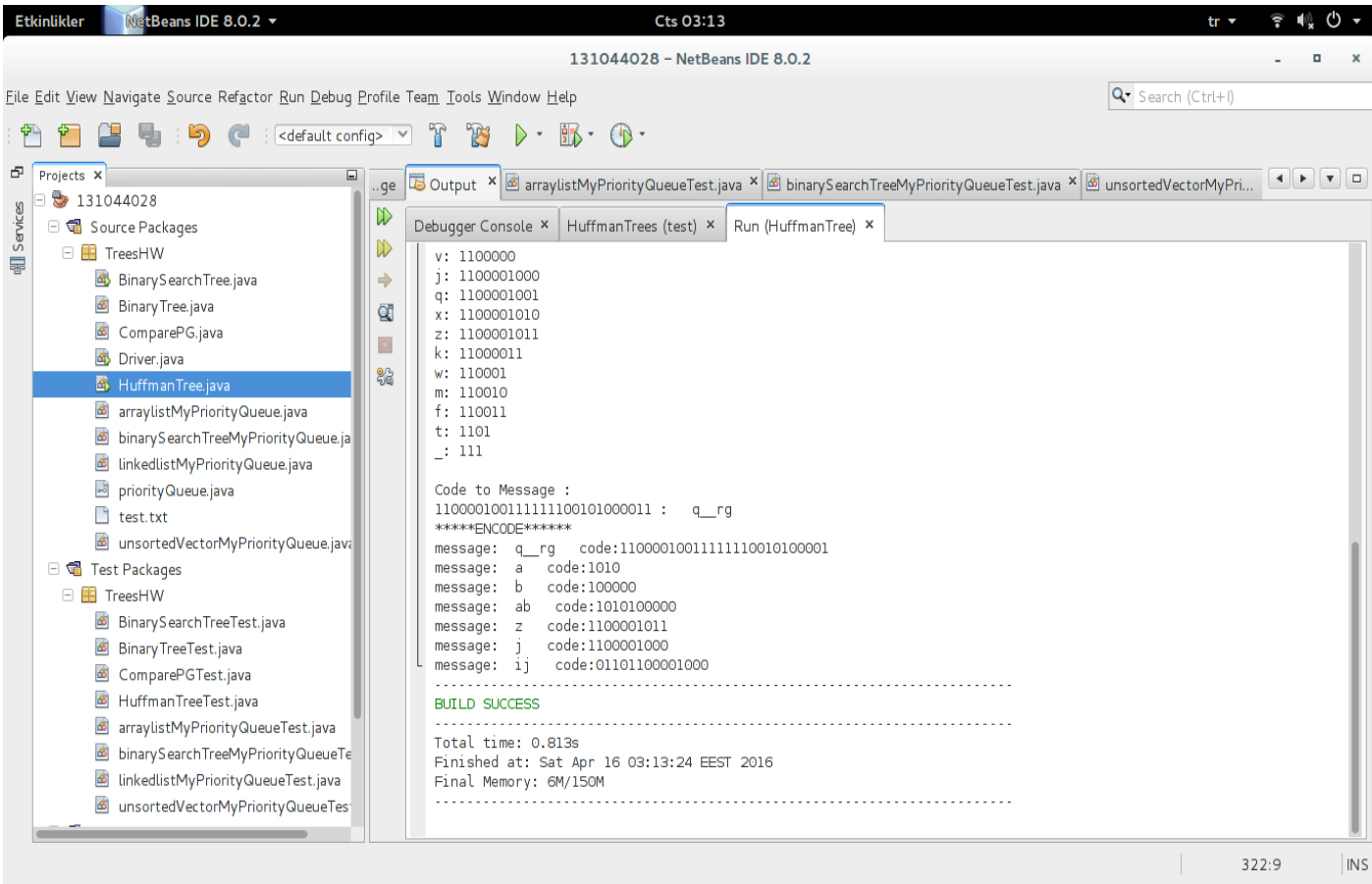



Etkinlikler
NetBeans IDE 8.0.2
Cts 02:56
tr
131044028 – NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Projects
Services
131044028
Source Packages
Test Packages
TreesHW
BinarySearchTreeTest.java
BinaryTreeTest.java
ComparePGTest.java
HuffmanTreeTest.java
arraylistMyPriorityQueueTest.java
binarySearchTreeMyPriorityQueueTest
linkedlistMyPriorityQueueTest.java
unsortedVectorMyPriorityQueueTest.j
Other Sources
Dependencies
Test Dependencies
Java Dependencies
Project Files
Start Page
Output
arraylistMyPriorityQueueTest.java
binarySearchTreeMyPriorityQueueTest.java
Debugger Console
HuffmanTrees (test)
Test (binarySearchTreeMyPriorityQueueTest)
****
Test metodu çağrılacak.
dequeue
0den 10a kadar olan priorityqueue olusturdum
Silmeden once:
Listedeki elemanlar:
0
1
2
3
4
5
6
7
8
9
dequeue yaptiktan sonra gelen eleman:
0
Sildikten sonra:
Listedeki elemanlar:
1
2
3
4
5
6
7
8
9
1:1
INS

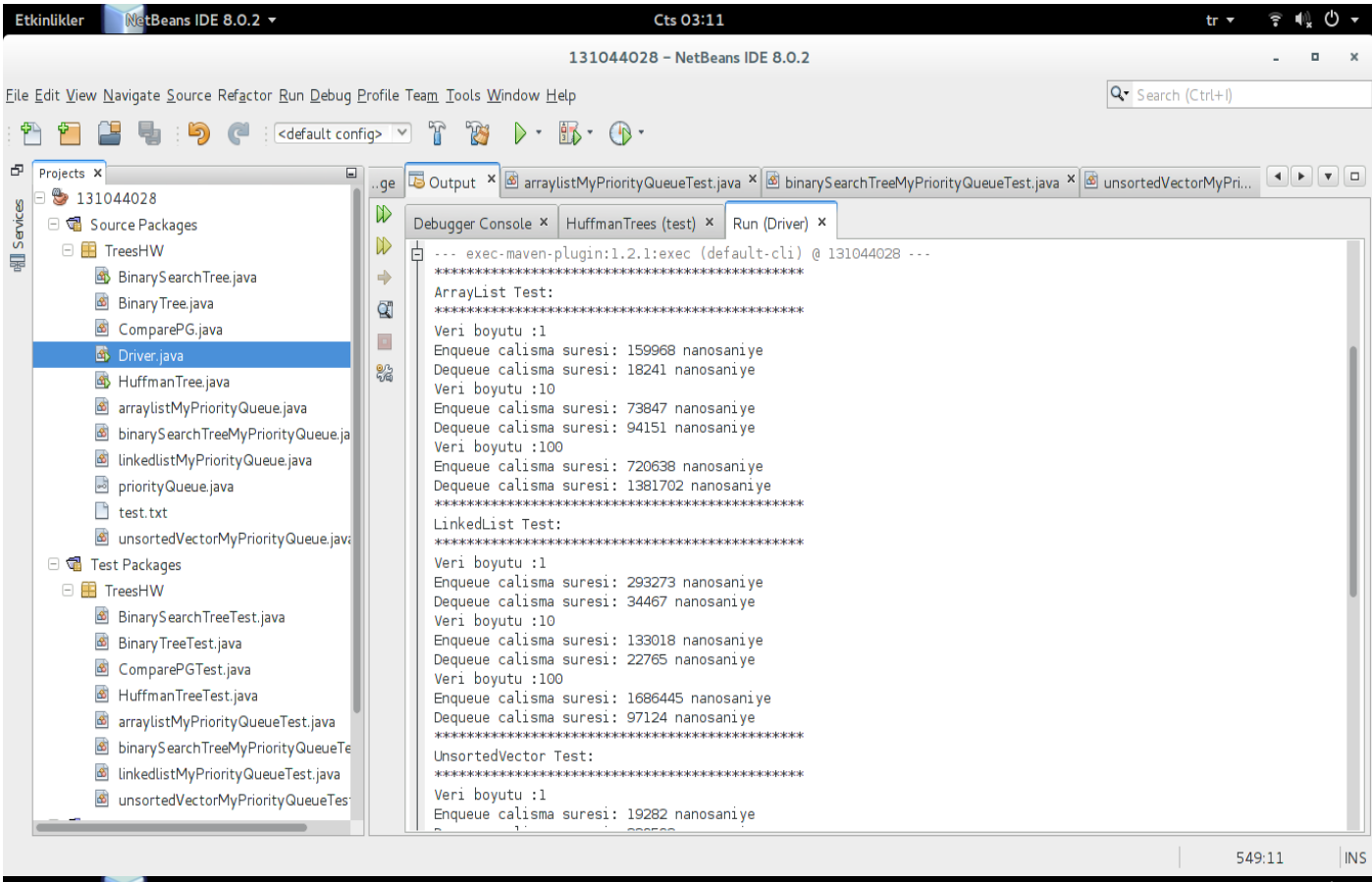

Etkinlikler
NetBeans IDE 8.0.2
Cts 02:56
tr
131044028 – NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Projects
Services
131044028
Source Packages
Test Packages
TreesHW
BinarySearchTreeTest.java
BinaryTreeTest.java
ComparePGTest.java
HuffmanTreeTest.java
arraylistMyPriorityQueueTest.java
binarySearchTreeMyPriorityQueueTest
linkedlistMyPriorityQueueTest.java
unsortedVectorMyPriorityQueueTest.j
Other Sources
Dependencies
Test Dependencies
Java Dependencies
Project Files
Start Page
Output
arraylistMyPriorityQueueTest.java
binarySearchTreeMyPriorityQueueTest.java
Debugger Console
HuffmanTrees (test)
Test (binarySearchTreeMyPriorityQueueTest)
****
Test metodu çağrılacak.
size
0den 10a kadar olan priorityqueue olusturdum
size() metodu: 10
Test metodu bitti.
****
****
Test metodu çağrılacak.
enqueue
0den 10a kadar olan priorityqueue olusturdum
Listedeki elemanlar:
0
1
2
3
4
5
6
7
8
9
Test metodu bitti.
****
1:1
INS

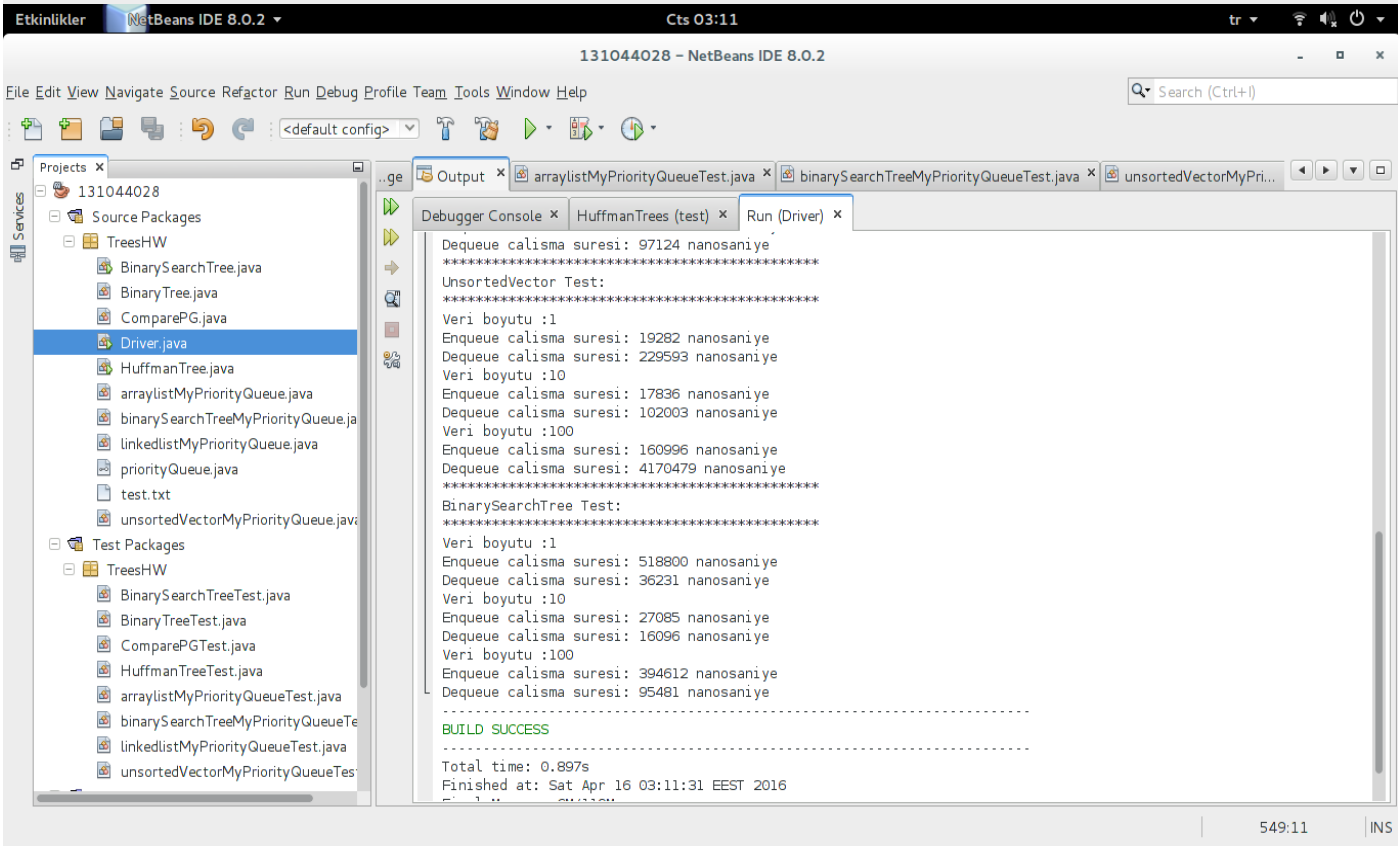
MAINDE YAZILAN
TEST
SENARYOLARI:

HuffmanTree main



Part3 main (Driver):


Etkinlikler
NetBeans IDE 8.0.2
Cts 03:11
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>
Projects
Services
131044028
Source Packages
TreesHW
BinarySearchTree.java
BinaryTree.java
ComparePG.java
Driver.java
HuffmanTree.java
arraylistMyPriorityQueue.java
binarySearchTreeMyPriorityQueue.ja
linkedlistMyPriorityQueue.java
priorityQueue.java
test.txt
unsortedVectorMyPriorityQueue.java
Test Packages
TreesHW
BinarySearchTreeTest.java
BinaryTreeTest.java
ComparePGTest.java
HuffmanTreeTest.java
arraylistMyPriorityQueueTest.java
binarySearchTreeMyPriorityQueueTe
linkedlistMyPriorityQueueTest.java
unsortedVectorMyPriorityQueueTes
..ge
Output
arraylistMyPriorityQueueTest.java
binarySearchTreeMyPriorityQueueTest.java
unsortedVectorMyPri...
Debugger Console
HuffmanTrees (test)
Run (Driver)
--- exec-maven-plugin:1.2.1:exec (default-cli) @ 131044028 ---
****
ArrayList Test:
****
Veri boyutu :1
Enqueue calisma suresi: 159968 nanosaniye
Dequeue calisma suresi: 18241 nanosaniye
Veri boyutu :10
Enqueue calisma suresi: 73847 nanosaniye
Dequeue calisma suresi: 94151 nanosaniye
Veri boyutu :100
Enqueue calisma suresi: 720638 nanosaniye
Dequeue calisma suresi: 1381702 nanosaniye
****
LinkedList Test:
****
Veri boyutu :1
Enqueue calisma suresi: 293273 nanosaniye
Dequeue calisma suresi: 34467 nanosaniye
Veri boyutu :10
Enqueue calisma suresi: 133018 nanosaniye
Dequeue calisma suresi: 22765 nanosaniye
Veri boyutu :100
Enqueue calisma suresi: 1686445 nanosaniye
Dequeue calisma suresi: 97124 nanosaniye
****
UnsortedVector Test:
****
Veri boyutu :1
Enqueue calisma suresi: 19282 nanosaniye
549:11
INS


Etkinlikler
NetBeans IDE 8.0.2
Cts 03:11
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>
Projects
Services
131044028
Source Packages
TreesHW
BinarySearchTree.java
BinaryTree.java
ComparePG.java
Driver.java
HuffmanTree.java
arraylistMyPriorityQueue.java
binarySearchTreeMyPriorityQueue.ja
linkedlistMyPriorityQueue.java
priorityQueue.java
test.txt
unsortedVectorMyPriorityQueue.java
Test Packages
TreesHW
BinarySearchTreeTest.java
BinaryTreeTest.java
ComparePGTest.java
HuffmanTreeTest.java
arraylistMyPriorityQueueTest.java
binarySearchTreeMyPriorityQueueTe
linkedlistMyPriorityQueueTest.java
unsortedVectorMyPriorityQueueTes
..ge
Output
arraylistMyPriorityQueueTest.java
binarySearchTreeMyPriorityQueueTest.java
unsortedVectorMyPri...
Debugger Console
HuffmanTrees (test)
Run (Driver)
Dequeue calisma suresi: 97124 nanosaniye
****
UnsortedVector Test:
****
Veri boyutu :1
Enqueue calisma suresi: 19282 nanosaniye
Dequeue calisma suresi: 229593 nanosaniye
Veri boyutu :10
Enqueue calisma suresi: 17836 nanosaniye
Dequeue calisma suresi: 102003 nanosaniye
Veri boyutu :100
Enqueue calisma suresi: 160996 nanosaniye
Dequeue calisma suresi: 4170479 nanosaniye
****
BinarySearchTree Test:
****
Veri boyutu :1
Enqueue calisma suresi: 518800 nanosaniye
Dequeue calisma suresi: 36231 nanosaniye
Veri boyutu :10
Enqueue calisma suresi: 27085 nanosaniye
Dequeue calisma suresi: 16096 nanosaniye
Veri boyutu :100
Enqueue calisma suresi: 394612 nanosaniye
Dequeue calisma suresi: 95481 nanosaniye
BUILD SUCCESS
Total time: 0.897s
Finished at: Sat Apr 16 03:11:31 EEST 2016
549:11
INS

JUNIT TEST SONUCU

```
Butun testler başarılı
```