CSE 222 – HW7

HASAN MEN 131044009

https://github.com/hmenn/CSE222-HW7-2016-Simulation

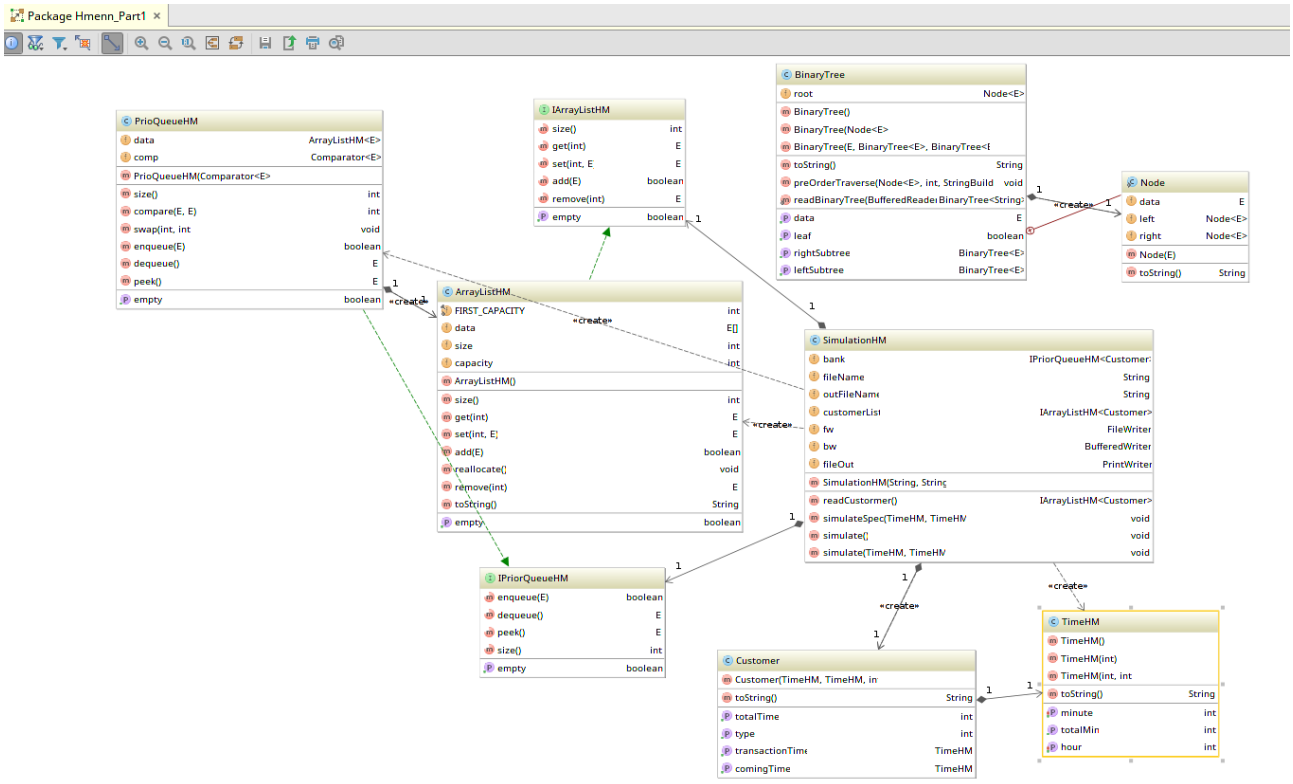
**2 Gün geç gönderim hakkımı bu ödev için kullanıyorum.

Part1:

Neler Yapılacak ?

Bir banka için küçük bir simulasyon hazırlanacak. Tek vezne üzerinden servis yapılmaktadır. Müşteri bilgileri input doyasından okunup sonuçlar ekrana yada output dosyasına ()basılacak. Sistem default olarak müşteriler bitene kadar çalışacaktır ek olarak sistem belirli aralıklarda çalıştırılabilir.

UML Diyagramı :

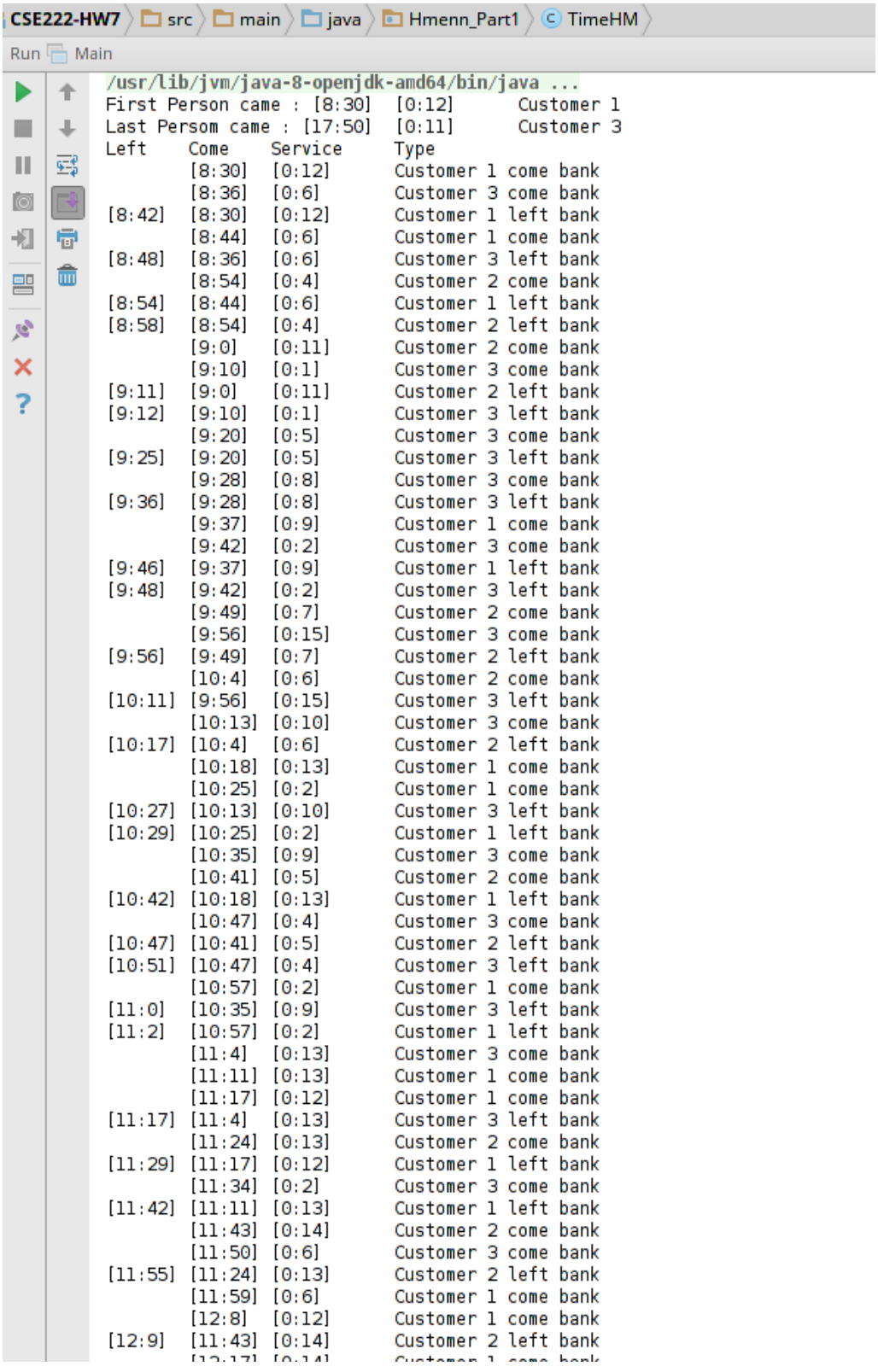


Nasıl Çalışır?

1. Simulasyon objemize input dosyanin path ini veririz.
2. Dosyadan müşterileri okur ve Customer Listine kaydeder.
3. Simulasyonu baslatırız.
4.1. Default olarak ilk elen ve son gelen musterinin saatlerini belirleyerek o aralıklarda sistem dakika arttırarak devam eder.
4.2 Ya da verilen süre aralığında artış yaparak simulasyona devam eder.

5. Müşteri gelince queue boş ise direk alınır.
6. İşlemi devam eden varsa queue ye eklenir.
7. Sistem müşterinin bitiş zamanı geldiğinde onu çıkarır ve yenisini alır.
8. Tüm müşteriler bitene kadar yada 4.2 deki gibi verilen süre bitene kadar devam eder.
9. Eger kuyrukta musteri kaldıysa hepsi sırayla isleme alınır ve biter.

Output1 : Default çalışma (data1.txt)



Output2 Belirli zaman aralıklarında calısma (data2.txt)

```
StartTime : [0:0]
 EndTime : [20:0]
Left     Come     Service      Type
         [0:0]    [0:13]       Customer 3 come bank
         [0:5]    [0:11]       Customer 3 come bank
         [0:13]   [0:4]        Customer 3 come bank
[0:13]   [0:0]    [0:13]       Customer 3 left bank
[0:17]   [0:13]   [0:4]        Customer 3 left bank
         [0:20]   [0:2]        Customer 1 come bank
         [0:26]   [0:15]       Customer 1 come bank
[0:28]   [0:5]    [0:11]       Customer 3 left bank
         [0:34]   [0:6]        Customer 3 come bank
[0:43]   [0:26]   [0:15]       Customer 1 left bank
         [0:44]   [0:10]       Customer 3 come bank
[0:45]   [0:20]   [0:2]        Customer 1 left bank
         [0:49]   [0:14]       Customer 1 come bank
[0:55]   [0:44]   [0:10]       Customer 3 left bank
         [0:57]   [0:11]       Customer 2 come bank
         [1:4]    [0:6]        Customer 2 come bank
[1:9]    [0:49]   [0:14]       Customer 1 left bank
         [1:11]   [0:15]       Customer 1 come bank
[1:15]   [1:4]    [0:6]        Customer 2 left bank
         [1:21]   [0:13]       Customer 2 come bank
         [1:28]   [0:8]        Customer 2 come bank
[1:30]   [1:11]   [0:15]       Customer 1 left bank
         [1:34]   [0:13]       Customer 1 come bank
[1:38]   [1:28]   [0:8]        Customer 2 left bank
         [1:41]   [0:14]       Customer 3 come bank
         [1:47]   [0:11]       Customer 2 come bank
[1:51]   [1:34]   [0:13]       Customer 1 left bank
         [1:53]   [0:5]        Customer 3 come bank
         [2:1]    [0:5]        Customer 3 come bank
[2:2]    [1:47]   [0:11]       Customer 2 left bank
         [2:11]   [0:5]        Customer 3 come bank
[2:15]   [1:21]   [0:13]       Customer 2 left bank
         [2:21]   [0:14]       Customer 1 come bank
[2:26]   [0:57]   [0:11]       Customer 2 left bank
         [2:30]   [0:15]       Customer 3 come bank
         [2:40]   [0:10]       Customer 3 come bank
[2:40]   [2:21]   [0:14]       Customer 1 left bank
         [2:48]   [0:1]        Customer 3 come bank
[2:50]   [2:40]   [0:10]       Customer 3 left bank
[2:51]   [2:48]   [0:1]        Customer 3 left bank
         [2:55]   [0:14]       Customer 3 come bank
         [3:1]    [0:11]       Customer 2 come bank
[3:6]    [2:30]   [0:15]       Customer 3 left bank
         [3:9]    [0:5]        Customer 2 come bank
[3:17]   [3:1]    [0:11]       Customer 2 left bank
         [3:18]   [0:3]        Customer 2 come bank
[3:22]   [3:9]    [0:5]        Customer 2 left bank
         [3:24]   [0:5]        Customer 2 come bank
[3:25]   [3:18]   [0:3]        Customer 2 left bank
[3:30]   [3:24]   [0:5]        Customer 2 left bank
         [3:34]   [0:15]       Customer 3 come bank
         [3:43]   [0:4]        Customer 1 come bank
[3:44]   [2:55]   [0:14]       Customer 3 left bank
[3:48]   [3:43]   [0:4]        Customer 1 left bank
         [3:50]   [0:4]        Customer 1 come bank
```

Part2

Ihtiyaçlar :

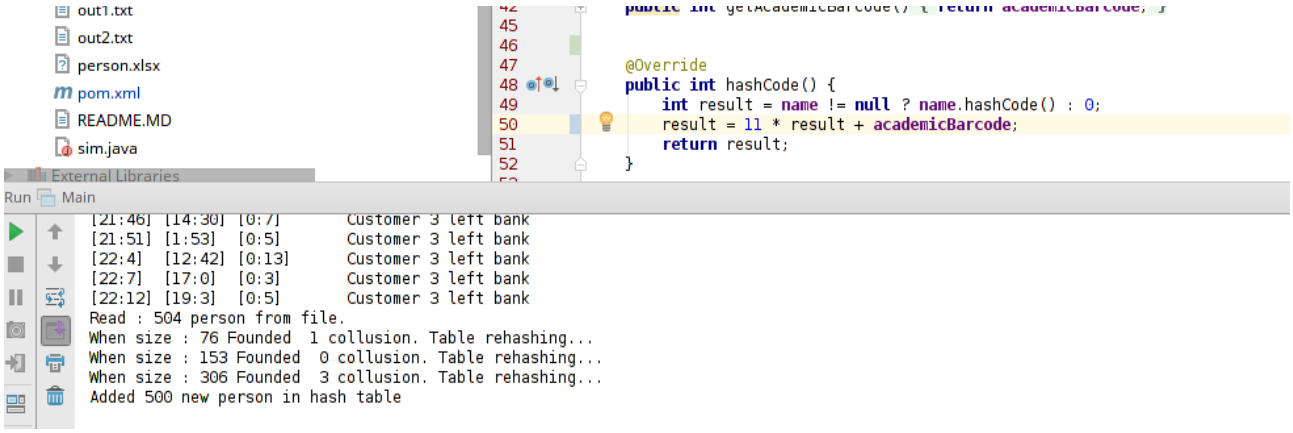
GTU de asistan hocaların hem öğrenci hem academic kartları vardır. 2Li kart kullanımını önlemek ve verimlilik için bu kartlar combine edilecek. Aynı isimli kişiler olması durumu göz önüne alınacak. Aynı isimli asistan ve academic calısanlar olabilir.

Nasıl Yapılacak-Nasıl Çalışır ?

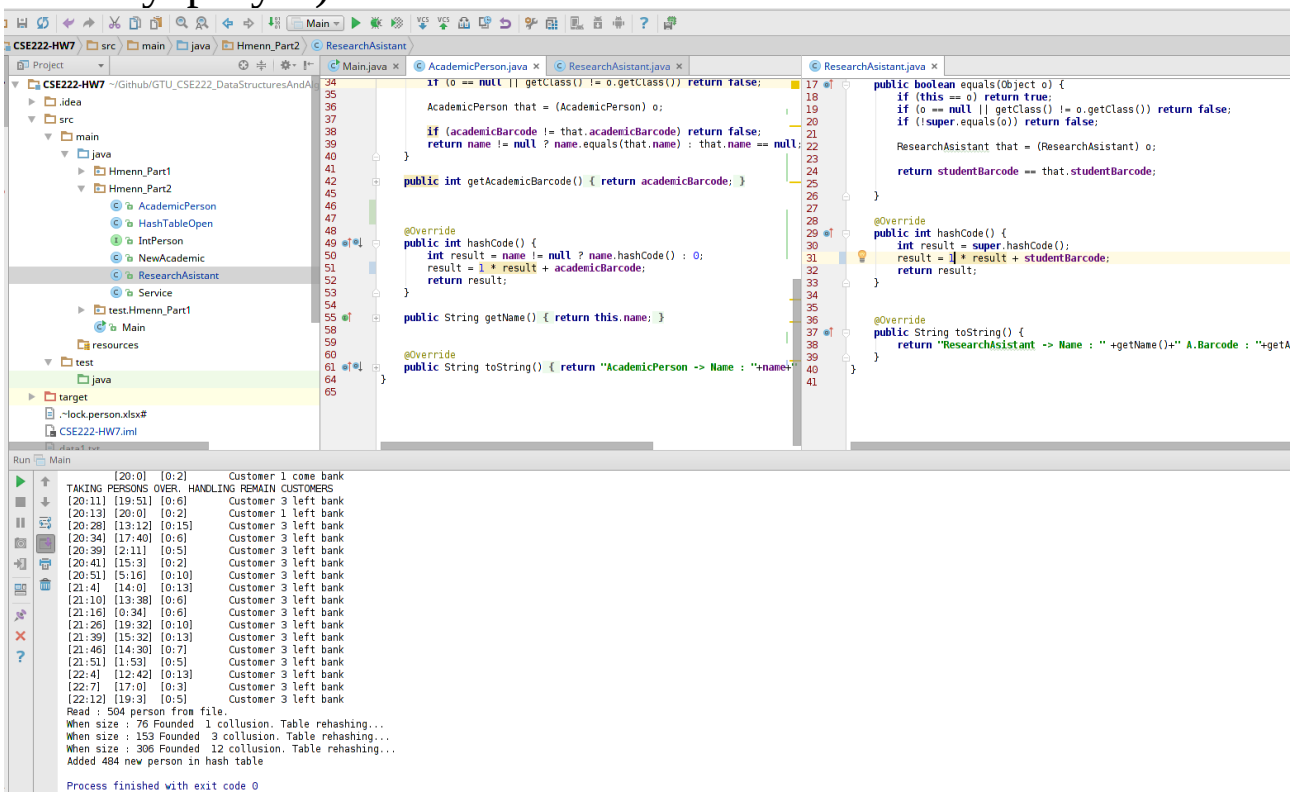
1. Input dosyasından kişiler okunacak ve listeye kaydedilecekler.
2. Tüm kişiler tekrardan hashlenecek.
3. HastTablosunda keyler barkod numaraları valueler kişi bilgileri olacak.
4. Barcode numaralar her kisiler icin olan barkod numalarrı ve isimlerinin hashlenmesi ile olusacak.
5. Stringler icin standart java hashcodu, integer degerler icin 17ile carpilarak hashleme yapildi.(* farklı hashler kullanılınca ortaya cıkan collisionlar asagida verilmiştir. 17 yada 31 iyi bir deger)
6. Olusan yeni kartlar tekrardan aynı yontemle hashlenerek bir 101 lik HashTablosu üzerine oturtulacaklar.
7. Tablo %75 dolum oranına gelince rehash edilmiştir.

Hash Katsayısı seçme :

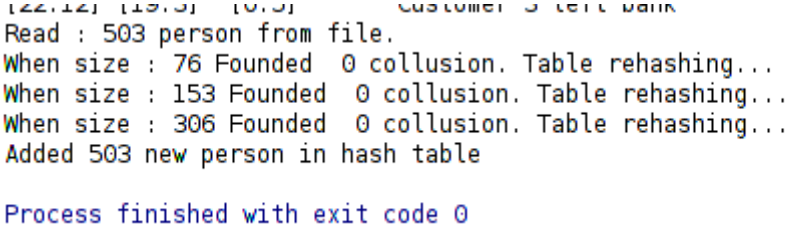
(11 iken son hashleme esnasında 3 collision var)



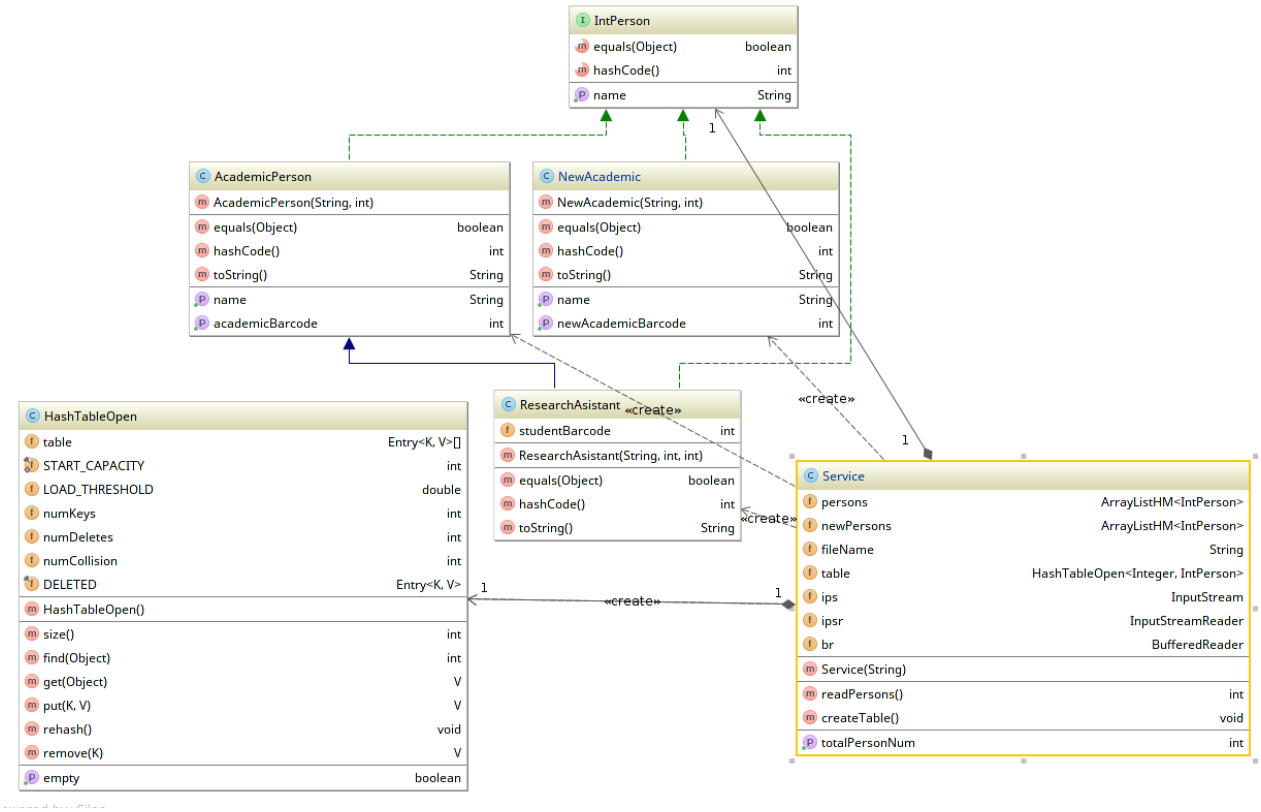
(1 olarak alininca1-3-12 tane collision olmustur her asamada-rehash yapılıyor)



(17 olarak katsayi alininca)



UML Diyagramı:



*Her asistan aynı zamanda bir academic personel oldugu icin türetilmiştir.

TESTLER – JUNIT – OUTPUT DOSYALARI

1. Part1 için Customer ve PriorityQueue sınıflarının tesleri yazılmıştır.
2. Part1 için daha detaylı inceleme için ayrı ayrı outpur dosyaları yazıldı.