

1) TowerOfHanoi

Belirtilen problemin çözümünün iterative olması istendiği için algoritma kurma konusunda oldukça zorlandım yaptığım deneyimler, hesaplamalar ve araştırmalar sonucunda oyunda yapılması gereken toplam hamle sayısının $2^n - 1$ olması gerektiği sonucuna vardım. Ve bu hamlelerin belirli bir düzene uygun biçimde sırasıyla tekrarlandığını saptadım. Geriye kalan implementasyon aşamasında 3 Kule için 3 farklı stack yapısı kullandım. Ve her hamlede spesifik olarak 2 kule arasında işlemleri özelleştirdim. Algoritma kurma konusunda internetteki kaynaklardan faydalandım. Kaynak olarak belirtebileceğim siteler aşağıdadır.

https://en.wikipedia.org/wiki/Tower_of_Hanoi#Iterative_solution

<http://zylla.wipos.p.lodz.pl/games/hanoi-ex.html>

2) private boolean remove(Node < E > head, Node < E > pred, E outData)

Yukarıda belirttiğim metod için öncelikle tek parametre alan wrapper metod yazdım. Bu wrapper metod ise yukarıdaki metodu kendi içinde çağırılmaktadır. Bu işlem sayesinde information hiding prensibine daha uygun bir kod yazmış oldum.

Silme işlemi için 2 koşul tasarladım bu koşullar sırasıyla aranan elemanın listenin ilk elemanı olması ve aranan elemanın listenin herhangi bir yerinde olmasıdır. Eğer aradığım eleman ilk elemansa listenin head değişkenini güncelleyerek silme işlemi gerçekleştiriyorum. Eğer aradığım eleman listenin ortasında veya sonunda ise pred değişkeni ile silme işlemi gerçekleştiriyorum. Metodun basecase koşulu ise head değişkeninin null olması.

3) public List intersectionOfLists();

```
public List unionOfLists();  
public boolean isSubset();
```

Verilen fonksiyon prototiplerini wrapper olarak kullandım ve arka planda bir integer parametre alan aynı isimde fonksiyonlar kullandım. Eğer parametre olarak gelen integer listelerin size değerlerine eşitse basecase koşulu sağlanmaktadır. Ve istenen özellikler fonksiyonlar tarafından sağlanmaktadır.

Bu ödev sonucunda Recursive düşünme ve algoritma oluşturma konusunda kendimi geliştirdiğimi düşünüyorum.