

---

## BİL-211 Bilgisayar Programlama - I

### Ödev - 2

Veriliş Tarihi: 02.02.2018

Teslim Tarihi: 13.02.2018 (Saat : 23:59)

Teslim Şekli: Piazza'dan duyurulacak.

**Kurallar:** Geç gönderilen ve teslim şekline uymayan ödevler kabul edilmez. Kopya kesinlikle yasaktır, kopya veren ve alan öğrenciler bütün ödevlerden 0 alırlar ve ayrıca üniversite disiplin yönetmeliği kuralları bu öğrencilere uygulanır.

---

Bu ödevde iki oyuncunun birbirleri ile bir çeşit sadeleştirilmiş satranç oynayabilmelerini sağlayan bir program yazacaksınız. Satrancın nasıl oynandığını bilmiyorsanız internette bunun için bolca kaynak mevcut.<sup>1</sup> Yazacağınız programda aşağıda belirtilenler dışında satrancın bütün kuralları/hamleleri geçerli olacak:

- En passant (Geçerken Alma): Piyonların geçerken alma hamlesi sadeleştirilmiş satrancımızda olmayacak.
- Castling (Rok): Şah ile kalenin özel bir hamlesi olan rok hamlesi de geçerli değil.
- Saat : Hamlenin yapılmasında bir zaman sınırlaması yok.

Satranç tahtası bildiğiniz gibi 8x8 lik bir griddir ve aşağıdaki şekilde numaralandırılır:

	a	b	c	d	e	f	g	h
8	.	.	.	.	.	.	.	.
7	.	.	.	.	.	.	.	.
6	.	.	.	.	.	.	.	.
5	.	.	.	.	.	.	.	.
4	.	.	.	.	.	.	.	.
3	.	.	.	.	.	.	.	.
2	.	.	.	.	.	.	.	.
1	.	.	.	.	.	.	.	.

Taşların olduğu pozisyonlar da bu numaralandırmaya göre isimlendirilir; a4, b2, c7 gibi.

### Nasıl yazacaksınız?

Yazacağınız kod aşağıda verilen kurallara uygun yazılmalıdır.

1. Öncelikle aşağıdaki metotlara ve veri üyelerine sahip Tas sınıfını yazınız.

```
private char color; // tasın rengi
private String location; // tasın o an tahtadaki konumu : "a3", "c4" vs.
public void setColor(char c)
public char getColor()
public void setLocation(String loc)
public String getLocation()
public boolean checkMove(String loc) // tas verilen loc konumuna satranc kurallari icinde hareket ettirilebilir mi?
public String[] getMoves() // tasın o an kurallara uygun olarak gidebilecegi butun noktaları bir dizide doner.
```

2. Her bir taş için ayrı bir sınıf tanımlayınız. Bu sınıfların hepsi yukarıdaki Tas sınıfından türetilmelidir.
3. *Chess* isimli çalıştırılabilir bir sınıf yazınız. Sınıf tahtayı aşağıdaki gibi tutmalıdır.

```
Tas[][] board = new Tas[8][8];
```

*Chess* sınıfı aşağıda anlatıldığı şekilde iki kullanıcının satranç oynayabilmesini sağlamalıdır.

Oyun, şu adımlardan oluşur:

1. sıradaki oyuncudan yeni hamleyi al,
2. hamlenin doğru olup olmadığını kontrol et,

---

<sup>1</sup>Örneğin: <http://satranc.tr.net/kurallar.html>

3. doğru ise hamleyi gerçekleştir, değilse bir hata mesajı verip aynı kullanıcıdan tekrar hamle iste.

Hamle tabi ki karşı tarafın taşını yeme ile de sonuçlanabilir. Bu durumda oyunun bitip bitmediği (yani yenen taşın şah olup olmadığı) kontrol edilmelidir. Eğer oyun bittiyse oyunu hangi tarafın kaç hamlede kazandığı ekrana basılıp çıkılmalıdır.

#### Kullanıcıdan hamleyi alma

Oyuncu, hamlesini, genel kullanımdan biraz farklı biçimde, şu şekilde girmelidir:

b3 e6

Bu girdi, “b3’deki taşı e6’ya hareket ettir” anlamına geliyor. b3’te hangi taşın olduğu, e6’ya hareket edip edemediği vs. gibi soruları programınızın çözmesi gerekiyor.

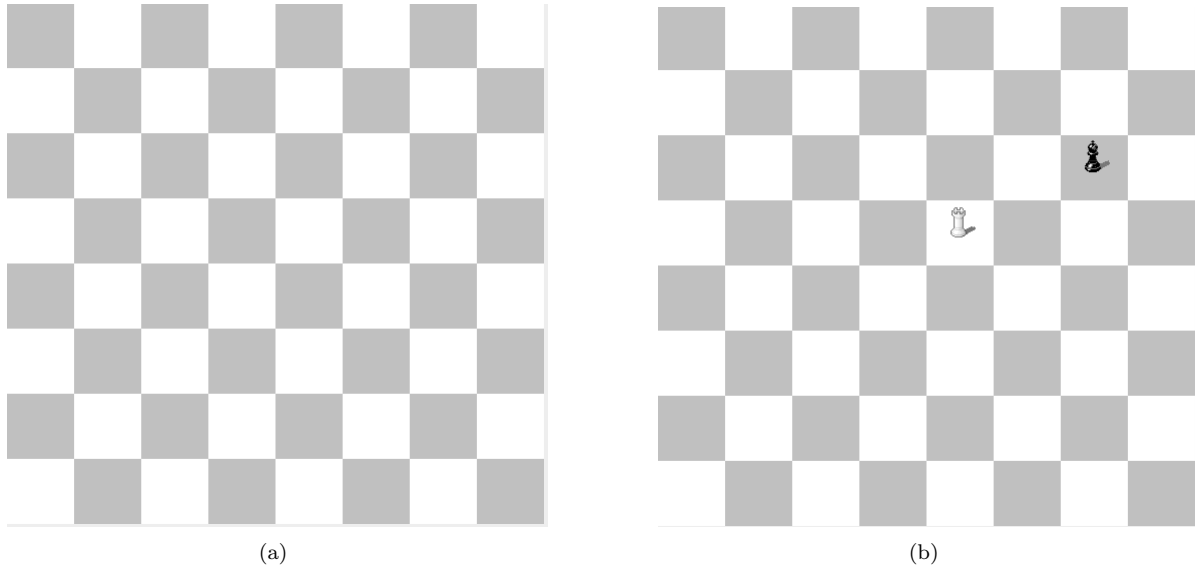
Hareket hamleleri dışında piyon karşı kenara ulaştığında gerçekleşen bir terfi (promotion) hamlesi var biliyorsunuz. Bunun için de girilmesi gereken hamlenin formatı aşağıdaki gibi olmalıdır:

d7 d8 vezir

Bu girdi de, “d7’de bulunan piyonu d8’e hareket ettir ve vezire terfi et” anlamına geliyor. Vezire çevirmek zorunda değilsiniz elbette, şah dışında herhangi bir taşa terfi edebilirsiniz. Terfilerde bir sınır yok; yani 8 piyonunuzu da vezire çevirip zaten elinizde bulunan vezirle beraber 9 vezire ulaşabilirsiniz.

#### Tahtayı görüntüleme

Çalıştırılan her adımdan sonra tahtanın son durumunu ekranda görüntüleyeceksiniz. Tahtayı görüntülemek için kullanacağınız ChessBoard adındaki sınıfı biz sağlayacağız. Bu sınıf tipinde bir nesne oluşturduğunuzda Şekil 1(a)’dakine benzer bir pencere açılacak ve programınız çalıştığı sürece ekranda kalacak.



Şekil 1: ChessBoard sınıfının çıktısı

Bu sınıfın, taşları görüntülemek ve yerini değiştirmek için kullanacağınız iki adet metodu var:

```
public void putIcon(int x,int y,String s,char c)
public void removeIcon(int x,int y,String s,char c)
```

putIcon metodu s dizgisi içinde tipini verdiğiniz, rengi c olan taşı (x,y) koordinatına çizecek. Yani, örneğin,

```
putIcon(3,4,"kale",'b');
putIcon(2,6,"fil",'s');
```

satırları, tahtayı Şekil 1(b)'deki duruma çevirir. (Buradaki koordinatlarla satranç tahtası numaralandırması arasındaki farka dikkat edin)

removeIcon ise tam tersini yapıp verilen noktadaki taşı tahtadan silecektir. Siyah bir kale koymak içinse c parametresini 's' olarak çağırmanız gerekiyor.

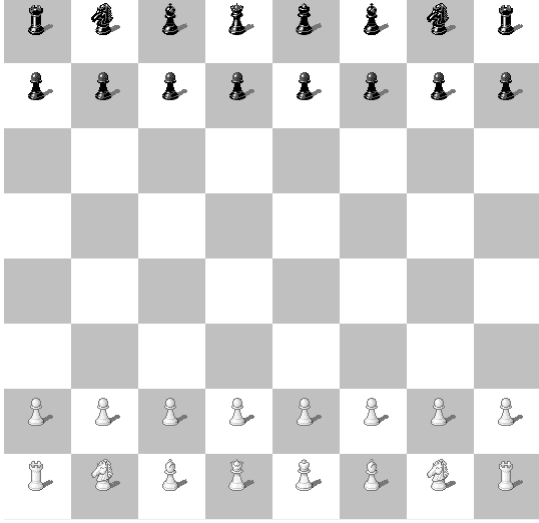
Bu metodlarda kullanabileceğiniz taşların isimleri: piyon, kale, at, fil, vezir ve şah (s ile).

ChessBoard sınıfının düzgün çalışabilmesi için sağlayacağımız resim dosyalarını, "eclipse working directory" altındaki images klasörüne kopyalamanız gerekiyor.

Terminalde çalışıyor iseniz çalıştığınız klasörün altındaki bir images klasörüne koymanız yeterli.

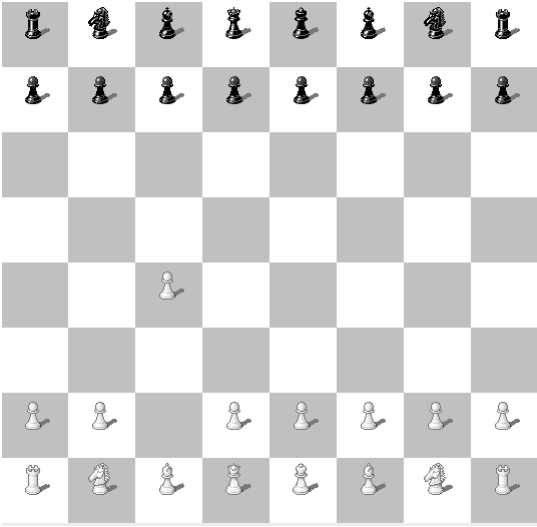
### Örnek çalışma

Aşağıda en başından itibaren oyunun ilk 4 hamlesini gösteren çalışma örneği var. Burada beyaz oyuncu 2, siyah oyuncu da yine 2 hamle yapmış oluyor.

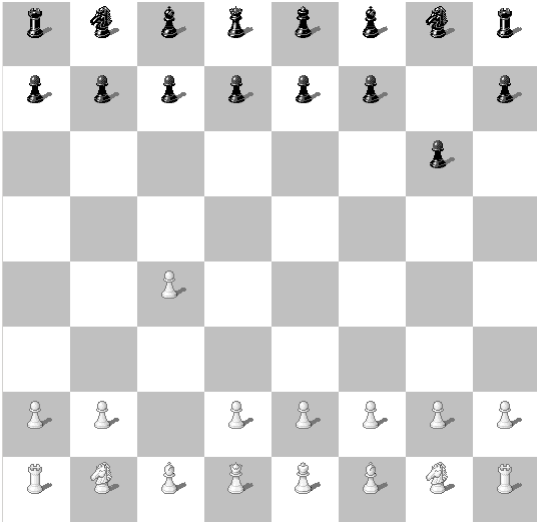


Oyuna beyaz başlayacak.

Beyazın hamlesi : c2 c4



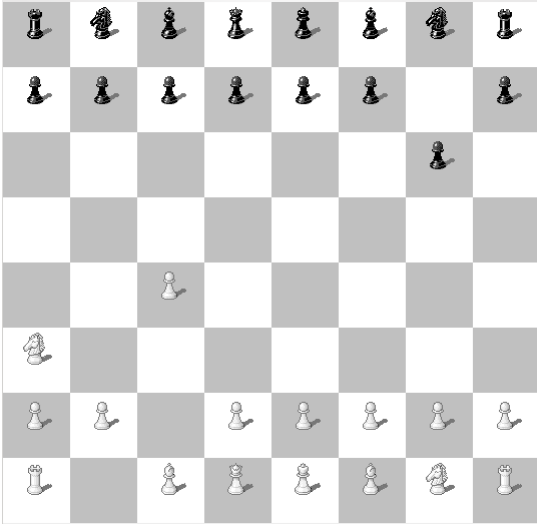
Siyahın hamlesi : g7 g6



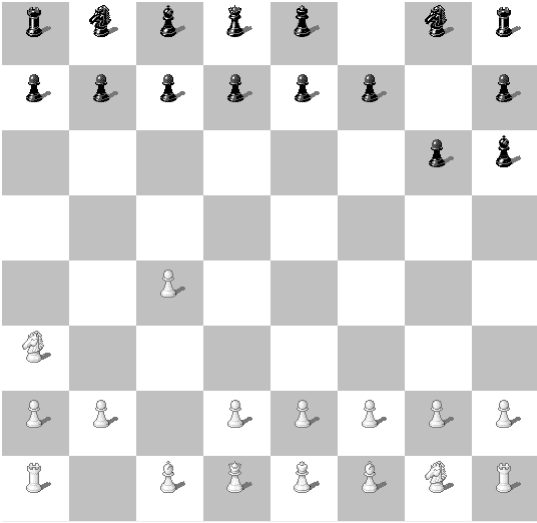
Beyazın hamlesi : c1 e3

Geçersiz hamle.

Beyazın hamlesi : b1 a3



Siyahın hamlesi : c6 c4  
Geçersiz hamle  
Siyahın hamlesi : a8 a5  
Geçersiz hamle  
Siyahın hamlesi : f8 h6



Bir oyuncu şahını kaybettiğinde aşağıdaki çıktıyı verip programı kapatabilirsiniz. Aşağıda beyaz oyuncunun kazandığı oyunda yaptığı hamle sayısı görüntüleniyor.

Beyaz 57 hamlede kazandı.