

# TAS KAGIT MAKAS OYNU PROJESİ

Egemen Çakır

*Kocaeli Bilgisayar Mühendisliği*

*Email: cakiregemen0@gmail.com*

## I. GIRIS

Bu projenin adı Tas Kagit Makas oynu. Bu projenin verilme amacı öğrencinin nesneye yönelik programlamayı öğrenmesi veya daha iyi kullanabilir hale gelmesidir. Bu projenin önemli olmasının sebebi nesneye yönelik programaya daha aşina hale getirmesidir. OOP(object oriented programming) ile nesne oluşturma bir sınıf içerisinde oluşturulur ve böylelikle tüm projelerde kullanılabilir. Bu sayede sınıflar 1 kez oluşturulur ve uzun kodların tekrardan yazılmasına gerek olmaz ayrıca OOP, gerçek hayattaki nesnelerin bilgisayara ortamına aktarılmasına denir. Bu projede sınıflar(nesnelerin özelliklerini ve davranışlarını belirler. Özellikleri değişkenler ile davranışları metotlar ile tutar.), nesneler(bir sınıfın örneği denebilir) ve encapsulation(Davranış ve özellikler sınıfta soyutlanarak kapsülendir. Kapsülleme ile hangi özellik ve davranışın dışarıya sunulup sunulmayacağını belirleriz.), inheritance(Sınıflar birbirinden türeyebilir. Alt sınıf üst sınıfın özelliklerini almasıdır), polymorphism(Alt sınıflar üst sınıfın gösterdiği davranışları göstermek zorunda değildir. Alt sınıfların farklı davranışları göstermesine çok biçimlilik denir.) ve soyutlama(bir nesnenin mimarisi denebilir.) gibi OOP nin özelliklerini bu projede fazlasıyla kullandım.

## II. ADIMLAR

### Adım 1

Öncelikle projede kullanmam gereken bazı sınıfları actım. Bunlar Nesne sınıfı ve ondan miras alan Tas sınıfı, AğırTas sınıfını, Kagit sınıfı, ÖzelKagit sınıfını, Makas sınıfını ve UstaMakas sınıfını açtım. Daha sonra Nesne sınıfının özellikleri olan dayanıklılık ve seviyePuanı adındaki değişkenleri tanımlayıp Nesne sınıfının parametrelili ve parametresiz iki constructors larını(yapıcı metot olarak gecer) yazdım. Sonra

dayanıklılık ve seviyePuanı özelliklerinin getter ve setter metotlarını yazdım çünkü bu özellikler başka bir sınıfta erişilebilirler diye yazdım. Nesne sınıfı abstract(soyut) bir sınıf olduğu için içinde bulunacak olan etkiHesapla, durumGuncelle ve nesnePuanıHesapla metotlarını soyut bir şekilde yazdım. Diğer sınıflar için:

- Tas Sınıfı: Tas sınıfı Nesne sınıfını miras aldığı için Nesne sınıfını extends ettim, seviyepuanı ve dayanıklılık özelliğinin yanında katılık özelliği ekledim ve getter ve setter metotlarını yazdım. Daha sonra iki adet constructor yazdım ve Nesne sınıfının constructor ına ihtiyacım olduğunda super anahtar kelimesi ile ona erişebildim. En son olarak soyut metotları override etmem gerekiyordu ettim fakat içlerini projenin ileriki zamanlarında doldurmak için boş bıraktım.
- AğırTas Sınıfı: AğırTas sınıfı Tas sınıfını miras alır dolayısıyla Nesne sınıfının özelliklerine ve metotlarında sahip olur. Dayanıklılık, seviyePuanı ve katılıktan ayrı sıcaklık özelliği ekledim ve iki adet constructor yazıp soyut olan metotları override ettim fakat taş sınıfındaki gibi içleri suanlık boş bıraktım.
- Kagit Sınıfı: Kagit sınıfı Nesne sınıfını miras aldığı için Nesne sınıfını extends ettim, seviyepuanı ve dayanıklılık özelliğinin yanında nüfuz özelliği ekledim ve getter ve setter metotlarını yazdım. Daha sonra iki adet constructor yazdım. En son olarak soyut metotları override etmem gerekiyordu ettim fakat içlerini projenin ileriki zamanlarında doldurmak için boş bıraktım.
- ÖzelKagit Sınıfı: ÖzelKagit sınıfı Kagit sınıfını miras alır dolayısıyla Nesne sınıfının özelliklerine ve metotlarında sahip olur. Dayanıklılık, seviyePuanı ve nüfuz özelliğinden ayrı kalınlık özelliği ekledim ve

iki adet constructor yazıp soyut olan metotları override ettim fakat taş sınıfındaki gibi içleri suanlık boş bıraktım.

- **Makas Sınıfı:** Makas sınıfı Nesne sınıfını miras aldığı için Nesne sınıfını extends ettim , seviyepuanı ve dayanıklılık özelliğinin yanında keskinlik özelliği ekledim ve getter ve setter metotlarını yazdım.Daha sonra iki adet constructor yazdım.En son olarak soyut metotları override etmem gerekiyordu ettim fakat içlerini projenin ileriki zamanlarında doldurmak için boş bıraktım.
- **UstaMakas Sınıfı:** UstaMakas sınıfı Makas sınıfını miras alır dolayısıyla Nesne sınıfının özelliklerine ve metotlarınada sahip olur.Dayanıklılık, seviyePuanı ve keskinlik özelliğinden ayrı hız özelliği ekledim ve iki adet constructor yazıp soyut olan metotları override ettim fakat taş sınıfındaki gibi içleri suanlık boş bıraktım.

## Adım 2

Sınıfların belli bir kısmını oluşturup düzenledikten sonra main() metodunun içinde bulunacağı Oyun sınıfını oluşturdum.Bu projede görsel arayüz için çerçeve görevi gören JFrame i extends ettim ve artık benim Oyun sınıfım bir JFrame oldu.Bundan sonra Oyun sınıfının constructor'ının içinde JFrame'in boyutunu , layout(sayfa düzenini) ayarladım ve JPanel panel = new JPanel() ile panellerimi oluşturdum ve JFrame'ime ekledim.Daha sonra JButton buton = new JButton() ile iki buton oluşturup alttaki panele ekledim.Üst panele ise JLabel label =new JLabel ile bir label oluşturup içine Oyun başlığını yazdım.En son olarak başlığımın altına üç tane JLabel koyup setIcon() metodu ile içlerine icon yerleştirip giriş penceresinin genel görünümünü verdim.

## Adım 3

Şimdiki aşamada eklediğim iki butona aksiyon verdim.Bunu yapmadan önce butona tıklandığında gerçekleşmesini istediğim olaylar için iki tane metod yazdım.Örneğin yap1 metodu kullanıcı-bilgisayar butonuna basınca giriş ekranını kapatması için setVisible(false) kodu ile giriş ekranı görünürlüğünü kapattım ayrıca daha sonra detaylı anlatacağım sahne2 isimli sınıfın bir nesnesini oluşturdum artık butona bastığımda yeni

bir pencere geliyo burada kullanıcı ve bilgisayar karşılıklı oyun oynayacaklar.Butona aksiyon ekleme işini aşağıdaki örnek koddaki gibi gerçekleştirdim.

```
JButton b1 =new JButton("Bilgisayar-Kullanıcı Oyna");
b1.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        yap1();
    }
});
```

Daha sonra aynı şekilde bilgisayar-bilgisayar butonu için yap2 metodunu yazdım ve içinde setVisible()'ı ve yeni penceremi oluşturacağım Sahne1 adında bir sınıftan nesne oluşturdum ve yukarıdaki görseldeki gibi aksiyon anında yapılması için metodu öğalırdım.

## Adım 4

Bu adımda kullanıcı-bilgisayar butonuna bastığımızda açılan pencereyi oluşturan Sahne2 sınıfını yazmaya başladım.Öncelikle JFrame'i miras aldım ve yazdığım tüm kodları Sahne2 nin constructor'ının içine yazdım.setsize() gibi JFrame'in temel ayarlarını yaptıktan sonra öncelikle beş panel oluşturup JFrame "CENTRE","WEST","EAST","NORTH" ve "SOUTH" olarak ekledim ve on buton oluşturdum.Bu butonlardan ikisi olan "Oyun oyna" ve "Nesne sec" butonlarını sol panele ekledim daha sonra üç buton olan "taş","kagıt" ve "makas" butonlarını alt panele ekledim daha sonra kalan beş butonu sag panele ekledim fakat bu butonlar hamle zamanında kullanılacaklar."Oyun oyna" ve "Nesne sec" butonlarını ekleme amacım kullanıcının nesne sec butonua tıkladığında alttaki üç butondan toplam beş nesnesini istediği gibi secmesini sağlamak ve bir kontrol ile toplam beş nesne sectiginde butonlar nesne ekleyemeyecek ve oyun oyna butonunu aktif hale gecip ona tıkladığında sag panelde yer alan hamle butonlarının aktif hale gelmesini amaçladım.Bunlara baslamadan önce nesneleri tutacak olan Oyuncu sınıfını miras alan Kullanıcı ve Bilgisayar sınıflarını yazdım.Bu üç sınıf:

- **Oyuncu Sınıfı:**Bu sınıfın özellikleri olan oyuncuID, oyuncuAdı, skor ve nesneListesi özelliklerini yazdım.Daha sonra bunlardan sadece nesneListesi dışında bulunan özelliklerin getter ve setter metotlarını yazdım.Daha sonra iki adet constructor yazdım ve soyut olarak yazdığım skorGoster ve nesneSec metotlarını tanıladım.

- Kullanıcı Sınıfı:Kullanıcı sınıfı Oyuncu sınıfını miras aldı ve onun tüm özelliklerini ve metotlarına sahip oldu fakat Oyuncu sınıfının metotları soyut olduğu için metotlar override edilmeli omları daha sonra doldurmak üzere override ettim.
- Bilgisayar Sınıfı:Kullanıcı sınıfı için ne yaptırıysam suanlık aynı şeyleri gerçekleştirdim.

### Adım 5

Oyuncu,Kullanıcı ve Bilgisayar sınıflarını yazdıktan sonra global olarak Kullanıcı ve bilgisayar sınıfından nesneler ürettim.Daha sonra "Nesne sec" butonuna JOptionPane'in static metodu ile ana ekrana mesaj yazdırdıktan sonra kart secimi için üç butonu aktif edecek boolean veri tipinde olan kartSecim değişkenini true deger atadım ve artık nesne secimi için en altta bulunan butonlara basışımda gerekli aksiyonlar gerceklesebilecek.kodu:

```
JButton b5 = new JButton("Nesne Secim");
Image img5 = new ImageIcon(this.getClass().getResource("/secim.png")).getImage();
b5.setIcon(new ImageIcon(img5));
b5.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        if (secim == true) {
            kartSecim = true;
            JOptionPane.showMessageDialog(pl, "Kart secmek için buton aktif edildi");
        } else {
            JOptionPane.showMessageDialog(pl, "Bu işlevi Kullanamazsınız");
        }
    }
});
```

Daha sonra örneğin Tas butonuna basışımda bir Tas sınıfından bir tas nesnesi ürettim ve kullanıcı nesnesinin özelliği olan nesneListesi ArrayListine nesneyi attım ve sag panelde yer alan butonlara eleman sayısı degerini parametre olarak alan ve nesneye göre butonu düzenleyen bir fonksiyon yazdım.Bu metot:

- ekle metodu:Parametre olarak bir tamsayı degeri alır ve birçok kosula göre belirli bir butonu düzenler.Örneğin parametre olarak bir gönderirsek aşağıdaki kodlar çalışacak:

```
switch (x) {
    case 1:
        String x1 = oyuncu1.nesneListesi.get(0).nesnePuaniGoter();
        String a1 = ad(oyuncu1.nesneListesi.get(0));
        b6.setText(a1);
        l3.setText(x1);

        if (a1.equals("Kagit")) {
            Image i = new ImageIcon(this.getClass().getResource("/kagit.png")).getImage();
            b6.setIcon(new ImageIcon(i));
        } else if (a1.equals("Makas")) {
            Image i = new ImageIcon(this.getClass().getResource("/makas.png")).getImage();
            b6.setIcon(new ImageIcon(i));
        } else {
            Image i = new ImageIcon(this.getClass().getResource("/tas.png")).getImage();
            b6.setIcon(new ImageIcon(i));
        }
        break;
}
```

daha sonra if kosulu ile kartSecim değişkenini false atadım.Bu kosulun gercekleme durumu kullanıcı.nesneListesi.size()=5 olma durumunda butonları kapatıp nesne secimini durdurmak için yaptım ve diger iki buton için sadece farklı nesneler üretildi ve eklendi.Örnek kod:

```
if (kartSecim == true) {
    if (oyuncu1.nesneListesi.size() < 5) {
        Makas m1 = new Makas();
        oyuncu1.nesneListesi.add(m1);
        int x=oyuncu1.nesneListesi.size();
        ekle(x);
    }
    if (oyuncu1.nesneListesi.size() == 5) {
        kartSecim = false;
        secim = false;
    }
}
```

Bu bes secim tamamlandıktan sonra Oyun başla butonum acıldı ve orada ilk önce JOptionPane'in yardımıyla kullanıcıdan hamle sayısını , adını , ID'sini aldıktan sonra bilgisayarın kartlarını secmesi için daha önce overloading ettiğim nesneSec metodunu çağırımdı ve rastgele nesnelerini seçtirdim.Daha sonra ekle metodu ile düzenlenen beş buton aktif hale geliyor.Overloading edilen nesneSec metodunun kodu aşağıda:

```
int sys = 0, deger = 0;
while (sys < 5) {
    deger = (int) (Math.random() * 3) + 1;
    if (deger == 1) {
        Tas t1 = new Tas();
        oyuncu2.nesneListesi.add(t1);
    } else if (deger == 2) {
        Kagit k1 = new Kagit();
        oyuncu2.nesneListesi.add(k1);
    } else {
        Makas m1 = new Makas();
        oyuncu2.nesneListesi.add(m1);
    }
    sys++;
}
```

### Adım 6

Öncelikle bu hamle yapacağımız butonların aksiyon kodları aynı işlevleri yapıyo o yüzden birtanesi üzerinden gidicem.Kullanıcı'nın nestelist özelliğindeki ilk nesneyi hamle olarak kullanmamız için oluşturdugum butonun aksiyonu bir if koşulunun içimde bu koşul bizim hamle secmeden aktif olmayan "Oyun oyna" butonu içinde "true" deger alan bir değişke ile kontrol ediliyor.Onun altında bu butonun ilk nesneyi temsil elttigini belirten bir "int i=1" değişken atamamız var ve bu atama beş buton içinde farklı sayılarda atandı.Daha sonra bütün nesneleri secmeden aynı nesneyi secemeyecegimiz için her butonun boolean degerinde durum değişkenleri var bu değişkenleri global olarak tanımladım ve basta hepsi true."i" atamasından sonra bu durum değişkenini kontrol eden bir if yapısı var bu if yapısının kosulu eger "şimdiye kadar secilen kart sayısı j hamleSayısı" ve "syc(her kart seciminde bir artar) mod syc7(suanlık degeri 5 çünkü beş eleman var fakat bir eleman silinecegi zaman bir azaltılır aynı zamanda syc da azltılır.) == 0" ise kosula girer,5 butonun durum değişkenlerini true yapar ve "syc" değişkenini sıfırlar.Bunların amacı her beş kartta ya da ne kadar kartımız varsa hepsini kullanmadan aynı kartı kullanmamamız için yaptım.Bu if kosulu bittikten sonra bu buton için gecerli durum1'in true olma kosulunu kontrol eden bir if var ve asıl işleri bu kısımda yapıyoruz. Giriş kısmın kodu:

```
if(hamle == true){
    int i=1;
    int j=0;
    if(syc%syc7==0 && syc3<hamleSayısı)
    {
        durum1=true;
        durum2=true;
        durum3=true;
        durum4=true;
        durum5=true;
        syc=0;
    }
    if(durum1 == true)
    {
```

### Adım 7

Bu durum değişkenini kontrol eden if koşulunun içini yazmadan burada kullanacağım ve daha önce içini boş bıraktığım metotları ve bu sınıfta yazdığım metotları başta açılalım:

- nesneSec metodu : Bu metot Kullanıcı sınıfında ve Bilgisayar sınıflarında bulunuyor.Bu metotun kullanıcı için dönüş degeri "Nesne" ve parametre olarak bir "int" deger ve Nesne tutan bir ArrayList alıyor.İçinde gelen int degerine göre Arraylistteki elemanları döndürüyor. Bilgisayar içinse bu metodu overloading yaparak hem beşta nesne seçiminde hemde hamle zamanında nesne seçerken kullandım.
- skorGoster metodu:Bu metot Kullanıcı sınıfında ve Bilgisayar sınıflarında bulunuyor , bir tamsayı döndürür ve parametresizdir.Ayrıca kodu sadece "return this.getSkor()" şeklindedir.
- nesnePuaniGoster metodu:Bu metot Nesne sınıfını miras alan tüm sınıflarda vardır ve hepsinde override edilip aynı kodlar yazılmıştır.Bu metot "String" bir deger dondurur.Bu metotun amacı hamle yaptığım butonların içinde butona ait nesnenin dayanıklılık ve seviye puanini göstermek için kullandım bu nedenle String bir deger donduruyor.
- etiketHesapla metodu:Bu metot Nesne sınıfını miras alan tüm sınıflarda vardır ve hepsinde override edilip farklı kodlar yazılmıştır.Öncelikle bu metot "double" bir deger dondurur ve parametre olarak int bir deger alır.Bu metotun int bir deger almasının sebebi karşı nesnelere verdiğim sayısal degerler içindir.Örneğin bir taş nesnesinin bu metodunu çağırdım ve karşı rakip kağıt nesnesini attı benim programımda kağıt nesnesinin sayısal degeri 4 ve bu gere göre bu metot taşın kağıt üzerindeki etkisini hesaplıyor tabi bu hesaplamalar projede verilen formuller ile gerçekleşiyor.
- durumGuncelle metodu:Bu metot Nesne sınıfını miras alan tüm sınıflarda vardır ve hepsinde override edilip aynı kodlar yazılı.Bu metodu overloading yapıp iki fonksiyon adında yazdım.Ilkl durumGuncelle metodu double bir deger alıyor bu double deger rakipten gelen hasar degeri ve bu metot nesnenin dayanıklılığını güncelliyor.Ikinci durumGuncelle metodu ise int deger alıyor ve seviye puanı güncelliyor.
- secenek metodu:Bu metot int bir deger döndürür , parametre olarak Nesne tipinde bir obje alır ve aldığı nesneyi "instanceof" operatörü ile hangi sınıfa aitse bir int deger döndürür.Örneğin bu metodun dönüş degerini

"etkiHesapla" metodunun parametresi olarak kullandım.

- duzenle ve duzenle2 metotları:Bu metotlar int bir deger alırlar ve bu int degere göre "CENTRE" a ekledigim panele ekledigim JLabel'lara icon ekler.Bu int degeri "secenek" metodundan dönen degerdir ve icon ekleme işlemleri if koşulları ile gerçekleşir.
- end metodu:Bu metot iki tane Nesne sınıfından obje tutan ArrayList alır.Bu ArrayListler kullanıcı ve Bilgisayarın Arraylistleridir.Bu metot hamle sayısı kadar hamle yaptığımızda çağrılır ve kullanıcının ve bilgisayarın elindeki kartların toplam dayanıklılıklarına göre karşılaştırıp galibi ekrana JOptionPane ile gösterir.
- random metodu:Bu metot int bir tamsayı döndürür ve parametre almaz.Bu metod başta bir sıfır ile dört arası bir sayı oluşturur ve sınıfta global olarak oluşturduğum int deger tutan bir Arrayliste atamak için kontroller yapar.Bunun amacı bir kartı diğerleri kullanmadan tekrar kullanmamak için.Bu kontroller öncelikle ArrayList boşsa Arrayliste ekler ve return olarak ürettiği sayısı döndürür.Fakat ArrayList boş değilse bir while döngüsü içinde ArrayList elemanları ile karşılaştırıp aynı eleman içermiyorsa ekler ve sayıyı donduru fakat sayıyı içeriyorsa iki ihtimal vardır ya bütün kartlar kullanılmış ya da başka kart atabilir onun için o kontrolleri yapar ya sayıyı tekrar üretir baştan bu kontrolleri yapar ya da diğer kartlar kullanıldıysa o koşulun içinde Arraylistin içindeki elemalar silinir daha sonra üretilen sayi eklenir ve sayi döndürülür.

## Adım 8

Adım 7 de durum değişkenini kontrol eden if koşulunun içinde kullandığım metotları ve Nesne sınıfındaki abstract metotların override edilmiş hallerini verdim.İlk önce int s=random() diyerek random metodundan dönen sayıyı aldım ve bu sayıyı Nesne o1=bilgisayar.nesneSec(s,bilgisayar.nesneListesi) içine gönderdim ki nesne sec hangi nesneyi dönderceğine karar verebilsin.Daha sonra butonun basında tanımladığımız "i" değişkenini kullanıcı nesneSec metodunda kullanarak bir Nesne objesi elde ediyoruz.Daha sonra hem bilgisayardan

hem kullanıcıdan elde ettiğimiz nesne objelerini secenek metoduna gönderip nesneye karşılık gelen tamsayıları alıyoruz ve bunları kullanıcı.nesneSec(i,kullanıcı.nesneListesi).durumGuncelle(bilgisayar.nesneSec(s2,kullanıcı.nesneListesi).etkiHesapla(s)) diyerek kullanıcının hamle sırasında kullandığı nesnein dayanıklılık degerini güncelliyoruz aynı şekilde bilgisayar olanında yapıyoruz.Daha sonra bu duzenle ve duzenle2 ye secenek metotlarından gelen int sayıları parametre olarak atıyoruz ve bu iki metot ekranda secilen nesnelerin görsellerini getiriyor.Daha sonra bazı durumları kontrol ettim.Bunlar:

- 1.durum:Eger kullanıcının kullandığı nesnenin dayanıklılığı güncellendikten sonra sıfırdan büyük ve bilgisayarın kullandığı nesnenin dayanıklılığı sıfırdan küçük eşit ise kullanıcının nesnesinin seviyePuanını durumGuncelle metodu ile yirmi arttırdım.Bunun devamında bir if blogu ile seviye puanının otuzu gecip gecmediğini kontrol ettim eger gectiyse if blogunun içinde yükseltilecek nesnenin sayısal degerini secenek metodu ile aldım ve karşılaştırmalar ile hangi nesne bir üst seviyeye çıkacaksa onun nesnesini oluşturup nesneList.set() ile nesnenin yerine atadım.Eger seviye puanı otuzu gecmediyse ya da yükseltme yapıp if blogundan çıktıktan sonraki işlem ise bilgisayarın nesnesini silmek.Cünkü dayanıklılığı sıfırın altında ya da eşit bu yüzden o nesneyi siliyoruz ve random içinde üretilen sayıları tutan Arraylistin nesnelerini siliyoruz ve sayı üretimini sağlayan syc5,mod işleminde kullanılan syc8'i birer azaltıp her bilgisayar nesnenin sectiginde artan syc4'ü sıfırlıyoruz.
- 2.durum:1.durumdaki olayın bu sefer bilgisayar nesnesinin seviye puanının yükseltilmesi eger otuzu gectiyse yükseltme işlemleri en sondada kullanıcının nesnesini sectiği butonu setEnabled(false) ile tıklama özelligini yok ediyoruz bu sayede o nesneyi bir daha secemeyecek.
- 3.durum:Bu durumda durumGuncelle() metodundan sonra her iki nesnenin dayanıklılıkları sıfır ya da sıfırdan küçük olabilir bu durumda bilgisayar nesnesini silme ve ardından yapılan işlemler ile kullanıcının nesnesini sectiği butonu kapatma işlemleri ile iki nesnede kullanılmaz hale getirilir.

- 4.durum:Yukarıdaki durumlar sonunda herhangi bir oyuncunun elindeki nesneler bitmiş ise o oyunu kaybeder bu sonuçta ekrana JOptionPane ile gösterilir.

Bu ard arda gelen if koşullarında sonra eger eger oyun bitmeyip devam ediyorsa butonun kontrolünü sağlayan durum değişkeni "false" yapılır ve hamle sayısı sayan syc3 değişkeni bir arttırılıp hamle sayısına eşit olup olmadıgı kontrol edilir eger hamle sayısına eşit olmuşsa sonucu gösterecek olan end metodu çağrılır ve Kullanıcı-Bilgisayar oynu ve sahne2 sınıfını bitirmiş olduk.

#### Adım 9

Şimdi sıra Sahne1 sınıfını yamak kaldı.Bilgisayar-Bilgisayar oyna butonuna tıklayınca bu sınıfın nesnesi oluşuyoe ve yeni bir cerceve olusup bilgisayar bilgisayara karşı oyun oynuyor.Öncelikle metotlar ve global değişkenler hariç her şey constructor içine yazdım.Ilk önce Sahne1'i JFrame'den miras aldım ve setsize(),setLayout() gibi temel ayarları yaptıktan sonra Sahne2'deki gibi beş panel oluşturup "CENTER","SOUTH","WEST","EAST"ve"NORTH" olarak JFrame ekledim bundan sonra on bir tane buton oluşturdum bunların beşini üst panele diğerlerini alt panele ekledim altaki bir buton haric diğerleri iki bilgisayarın hangi nesneleri sectigini gosterecek fakat bu on butondan herhangi bir işlem gerçekleştiremeyeceğiz aynı zamanda alt panele eklenen extra bir buton ise bir aksiyona sahip olacak ve işlemlerimizi orda gerçekleştireceğiz.

#### Adım 10

Bilgisayarların random bir şekilde nesnelerini secmesi,random hamle yapmaları ve benzeri şeyler için metotlar yazdım.Bunlar:

- secenek metodu:Bu metot int bir deger döndürür ve parametre olarak Nesne sınıfından bir obje alır bu objenin hangi sınıftan olduguna göre birden altıya kadar sayı döndürür.
- ekle1 ve ekle2 metotları:Bu metotlar aynı işi yaparlar fakat birisi bilgisayar1 diğeri bilgisayar2 için.Bu metotlar hiçbir şey döndürmezler ve parametre olarak Nesne sınıfından obje tutan bir ArrayList alır.İçinde her bir eleman için secenek metodu ile degerler alınır ve o degere göre butonlara icon ekleme işlemi yapar.

- duzenle ve duzenle2 metotları:Bu metotlar aynı işi yapar fakat biri bilgisayar1 diğeri bilgisayar2 için yapar.Hiçbir şey döndürmez ve parametre olarak bir tamsayı deger ve Nesne sınıfından obje alır.Gelen tamsayı degeri secenek metodundan gelir yani hangi nesne gönderilmişse onun sayısını döndürür ve bu sayıya göre merkezdeki JLabel'lara nesnenin iconunu getirir ayrıca yanına nesnenin dayanıklılık ve seviye puanını yazar.Bu metotlar nesneler karşılaşacağı zaman kullanılır.
- random1 ve random2 metotları: Bu metotlar aynı Sahne2 deki random metodu ile aynıdır fakat sadece biri bilgisayar1 için diğeri bilgisayar2 için çalışır.Bu metotların içerigini random2 metodu üzerinden açılacakım.Öncelikle sınıfın başında bu metot için global iki tane tamsayı değişken tutan ArrayList tanımladık daha sonra random2 başta bir sayı üretilir eger ArrayList boşsa üretilen sayı ArrayListe atılır,kart secim sayısını tutan syc2 değişkeni arttırılır ve sayı döndürülür.Eger boş değilse while içinde ArrayListteki diğer elemalar ile karşılaştırılır ve while içinden bir sayı gelir bu sayı ArrayListin eleman sayısına eşitse Arrayliste ekleme yapılır,syc2 arttırılır ve sayı döndürülür ya da syc2 mod syc3(bilgisayar2 deki elema sayısına yani beşe eşittir) == 0 ise ArrayListin içindeki elemanlar silinir bu demek oluyorki beş eleman eklenmiş ve seçilen bir nesne tekrar seçilebilir ve üretilen sayı Arrayliste eklenir ve syc2 arttırılır eger buda gerçekleşmezse yeniden bir sayı üretilir ve baştan bu kontrollerden bir sayı döndürünceye kadar devam eder.
- end metodu:Bu metot hiçbir şey döndürmez ve parametre olarak Nesne objesi tutan iki ArrayList alır.Bu metod hamle sayısına eşit sayıda hamle yapıldığında çağrılır ve gelen ArrayListler bilgisayar1 ve bilgisayar2 nin nesneList özellikleridir.Bu metot bu ArrayListleri kullanarak hangi tarafın dayanıklılık toplamı fazla ise JOptionPane ile sonucu ekranda gösteriyor.

#### Adım 11

Butonlarımı,JLabellarımı ekledikten sonra nesneSec metodunu kullanarak her iki bilgisayarın nesnelerini rastgele sectirdim daha sonra ekle1 ve



ekle2 metotlarına bilgisayar1 ve bilgisayar2'nin nesneListlerini parametre olarak gönderip üst ve alt panellerde yer alan toplam 10 butona nesnelerin iconlarını ekledim. Daha sonra alt kısımda yer alan hamle butonuna tıklandığında gerçekleşecek olayları yazmak kaldı. Butonun aksiyon kısmında ilk önce bir kontrol yapısı yazdım bu kontrol yapısı `syc==0` şeklinde bu `syc` bizim hamle sayımızı tutuyor yani daha hamle sayımız belli değil bu if blogunun içinde `JOptionPane`'in static metodu olan `showInputDialog` ile kullanıcıdan bir hamle sayısı aldım ve oynum bu hamle sayısına göre oynanacak. Daha sonra bu if koşulundan çıktıktan sonra bir if koşulu yazdım fakat bütün işlemler bu kısımda gerçekleşiyor ve bu koşulu yazma amaçım oyun bittikten sonra tekrar hamle yapılamaması. Bu koşul içinde ilk önce `random1` ve `random2` ile int değerleri alıp bilgisayar1 ve bilgisayar2'nin `nesneSec` metotlarına `nesneList`leri ile gönderdip bunları bir Nesne objesine eşitledim. Burada elde ettiğim `nesne1` ve `nesne2`'yi `secenek` metoduna gönderip nesnelerin sayısal değerlerini elde ettim. Bu sayısal değerleri `etkiHesapla` metotlarında kullanıp hangi nesneye karşı etki hesaplanacağını anlıyoruz. `Etkier` hesaplanıp `durumGuncelle` metoduna parametre olarak gider ve nesnelerin dayanıklılıkları güncellenir. Bu adımda en son olarak `duzenle` ve `duzenle2` yi çağırıp merkez paneldeki iconlar ve yazılar düzenleniyor.

#### Adım 12

Dayanıklılıklar güncellendikten sonra 4 koşulu kontrol ettim bu koşullar:

- 1.durum:Eğer bilgisayar1'in kullandığı nesnenin dayanıklılığı güncellendikten sonra sıfırdan büyük ve bilgisayar2'nin kullandığı nesnenin dayanıklılığı sıfırdan küçük eşit ise bilgisayar1'in nesnesinin `seviyePuanını` `durumGuncelle` metodu ile yirmi arttırdım. Bunun devamında bir if blogu ile `seviye` puanının otuzu geçip geçmediğini kontrol ettim eğer geçtiyse if blogunun içinde yükseltilecek nesnenin sayısal değerini `secenek` metodu ile aldım ve karşılaştırmalar ile hangi nesne bir üst seviyeye çıkacaksa onun nesnesini oluşturup `nesneList.set()` ile nesnenin yerine atadım ve yeni oluşturulan nesnenin iconunu yükselmeden önceki nesnenin butonuna ekledim. Eğer `seviye` puanı otuzu geçmediyse

ya da yükseltme yapıp if blogundan çıktıktan sonraki işlem ise bilgisayar2'nin nesnesini silmek. Çünkü dayanıklılığı sıfırın altında ya da eşit bu yüzden o nesneyi siliyoruz ve random içinde üretilen sayıları tutan `Arraylist`in nesnelerini siliyoruz ve sayı üretimini sağlayan `syc6,mod` işleminde kullanılan `syc3`'i birer azaltıp her bilgisayar nesnenin seçtiğinde artan `syc2`'yi sıfırlıyoruz.

- 2.durum:1.durumdaki olayın bu sefer bilgisayar2 nesnesinin `seviye` puanının yükseltilmesi eğer otuzu geçtiyse yükseltme işlemleri en sondada bilgisayar1'in kullandığı nesneyi silip sayıları tuttuğu `Arraylist`in elemanlarını sıfırlıyoruz ve bazı değişkenlerin değerlerini azaltıp sıfırlıyoruz.
- 3.durum:Bu durumda `durumGuncelle()` metodundan sonra her iki nesnenin dayanıklılıkları sıfır ya da sıfırdan küçük olabilir bu durumda bilgisayar1 ve bilgisayar2'nin nesnelerini silme ve ardından yapılan işlemleri yapılır.
- 4.durum:Yukarıdaki durumlar sonunda herhangi bir oyuncunun elindeki nesneler bitmiş ise o oyunu kaybeder bu sonuçta ekrana `JOptionPane` ile gösterilir.

#### Adım 13

Bu ard arda gelen if koşullarından sonra hamle sayısını tutan `syc` değişkeni bir artırılır ve hamle sayısı ekranda güncellendikten sonra `syc`'ın kullanıcının başta girdiği hamle sayısına eşit olup olmadığını kontrol edilir eğer eşitse bu işlemler gerçekleştirdiğimiz if koşulunu kontrole eden boolean değerindeki durum değişkeni false yapılır ki hamle sayısı bittiğinde bir daha hamle yapmayalım diye ve `end` metodu çağırılıp galip belirlenir. Bu adımda birlikte proje bitmiştir.

### III. ÖZET

Projedeki amacım nesneye yönelik programlamayı öğrenmek veya daha iyi şekilde kullanarak Tas Kağıt Makas oyunu projesini yapmaktır. Bu proje boyunca java'yı daha fazla hakim oldum, nesneye yönelik programlamada kullanılan birçok özelliği sebepleriyle öğrendim ve görsel bir proje olduğu için swing'e de hakim oldum. Proje bittiğinde artık ana ekranda kullanıcı-bilgisayar butonuna basıp kendi nesnelerimi

secip bilgisayara karşı oyun oynayabiliyordum ya da bilgisayar-bilgisayar butonuna basıp rastgele neslerini secip karşılıklı oyun oynatabiliyordum ve en sonunda galibin kim olduğunu ekranda gösterebildim.

#### IV. SONUC

Proje bittikten sonra yapılan testler ve yapım aşamasında yapılan testler sonrasında sorunlar giderildikten sonra artık ana ekranda iki butona basıp ya bilgisayara karşı kendimiz oynuyoruz ya da bilgisayar bilgisayara karşı oynatabiliyoruz. Burada karşılaştığım bazı sorunlardan bahsedeyim. Bunlar:

- 1. Sorun: Bilgisayarın nesnesini sildikten sonra örneğin 4. nesne 3. nesne olacak bu yüzden bilgisayar kart seçiminde sıkıntı çıktı. Bu sorunu bilgisayarın kartlarını seçmek için ürettiği sayıları tutan ArrayListin elemanlarını sıfırlayarak bir çözüm buldum.
- 2. Sorun: Kullanıcı bilgisayar bölümünde kullanıcının hamle yapması için koyduğum bes butonun birine bastıktan sonra diğerlerini beklemeden yine kullanılabir olma sorunu vardı. Bu sorunda her butona boolean durum değişkenleri verip kullanıldıktan sonra false yapıp hamle sayısına ve nesne sayısına göre koşullar sonucunda true ya döndürerek bu sorunu çözdüm.

#### V. KAYNAKÇA

- <https://www.javatpoint.com/java-tutorial>
- <https://www.w3schools.com/java/default.asp>
- <https://www.tutorialspoint.com/java/index.htm>

TABLE I  
OYNANIŞ TABLOSU

Adım Sayısı	Bilgisayar ya da Kullanıcı	Bilgisayar
1	Giriş ekranında seçim yapılır	
2	nesneler secir	nesneler secir
	Tas1	Makas1
	Tas2	Kagit
	Makas	Makas2
	Kagit	Tas1
	Tas3	Tas2
3	Hamle sayı ve bilgiler girilir	
4	Hamleler yapılır	
	Makas	Tas1
	Tas1	Tas2
	Kagit	Makas1
	Tas3	Kagit
	Tas2	Makas2
	Tas2	Tas2
	Kagit	Makas2
	Tas1	Kagit
	Makas	Tas1
	Tas3	Makas1
	.	.
	.	.
	.	.
5	Hamle sayısına gelindiğinde ya da bir rakipte nesne kalmadığında oyun biter	



