**Genel Bilgiler: Getter, Setter ve Constructor**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Class sayfasında (dikkat main de değil) Üç constructor alt alta ve hepsinin işlevi farklı:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

En alttaki constructor da email fieldı default atanmış. Her zaman [a@lol.com](mailto:a@lol.com) yazacak. Diğer field lar manuel girilecek.

Main sayfasında consturctorları çağırarak programı çalıştırıyoruz: Dikkat constructor lar farklı, ilki tüm fieldları getirebilen, diğeri, sadece iki field ın tanımlandığı, üçüncü fieldın ise class sayfasında default olarak girildiği constructor.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**8. HALI ÖRNEĞİ**

Üç tane class oluşturuyoruz. Zira halı satışı yapılacaksa bizim elimizde aşağıdaki hesaplamalar olmalı:

1. Taban (en ve boy bilgisi)
2. Hali (maliyet bilgisi olacak)
3. Hesaplayıcı (Taban ve hali bilgisini kullanacak)

Burada önemli olan halının alanını hesaplamak için bir metoda ihtiyacımız var. Bunun için getAlan metodu oluşturacağız.

En ve boy a dair 0 den küçükse 0 olarak atayın gibi bir koşulu, en ve boy fieldlarını oluşturup altına getirdiğimiz constructor bloğunun içine yazıyoruz.

Alanı burada hesaplayacağımız için getAlan (Getter)metodunu Taban sayfasında formüle ediyoruz. Aynı şekilde maliyeti de Halı da belirlediğimiz için getMaliyet (Getter)metodunu Hali da yazıyoruz. Metod da void yazmıyoruz, işlem yapıp alan değerini alacağımız için veri tipini belirteceğiz. Dolayısıyla return kullanıp, en ve boyu return de çarpmalıyız:

**Taban Class’ı**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Hali Class’ı**

**metin içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Hesaplayici Class’ı:**

**metin içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Main:**

**metin içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Dikkat:** Önce Taban ve Hali sınıflarını oluşturdurk. Şimdi bunlar üzerinden hesaplama yapabilecek bir sınıf oluşturuyoruz. Bu üçüncü sınıfta ilk iki sınıftan veri çekeceğiz. Bunun için Taban veri tipinde taban, Hali veri tipinde hali fieldları oluşturuyoruz:

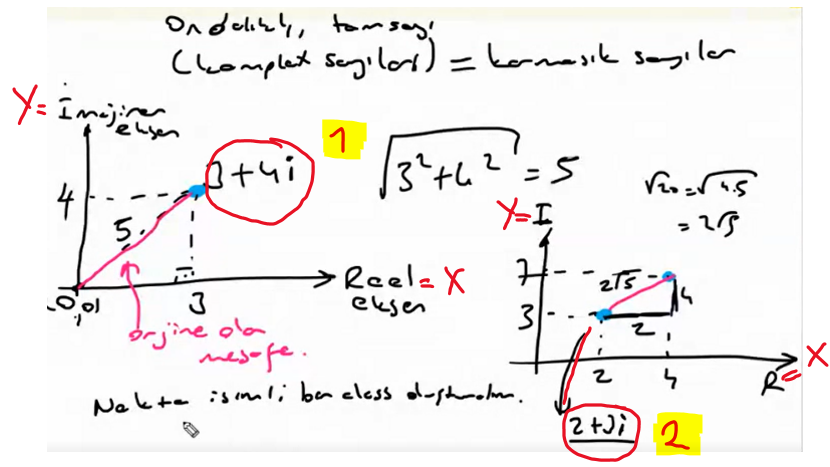
metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Sonrasında Hesaplayıcı constructor ını oluşturyoruz. Ve nihayet ToplamFiyatDon metodunu formüle ediyoruz.

**9. KOMPLEX (KARMAŞIK) SAYILAR:**

Komplex sayı veri tipi yok. Ondalık sayıları (double), tam sayıları(int) tutan veti türleri vardı ancak komplex sayılar için yok. Biz Class oluşturup veri tipi halinde kullanabiliyorduk. Öyleyse komplex sayılar için veri tipi oluşturalım.

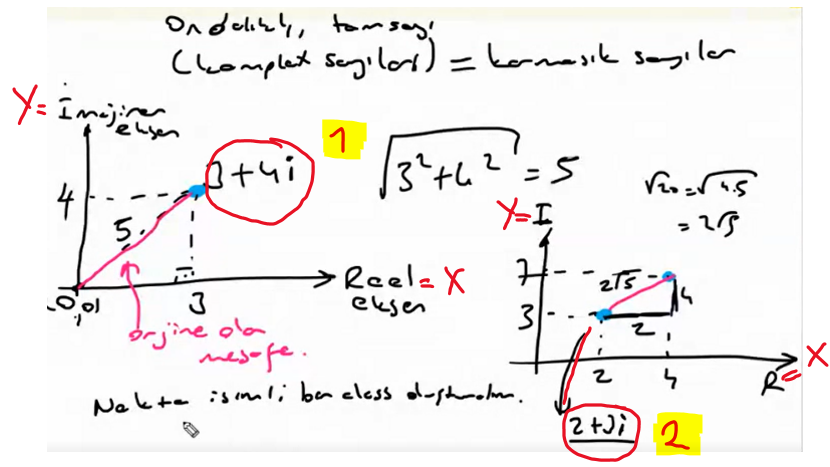


1. Nokta isimli bir Class oluşturacağız.
2. Bunun 2 adet double parametresi olacak (x,y)
3. Bir constuctor oluşturalım, bütün field ları construct etsin.
4. Mesafeyi ölçeceğiz, değer üretecek bir işlem olacak. Dolayısıyla bir metoda ihtiyacımız var. Adı mesafe olsun(mesafe(x,y)). Hiçbir parametre almasın. Mesafe isimli Metodumuz karmaşık sayının orijine olan mesafesini bulsun. Field larda tanımlanmış değerleri kullanarak ilgili karmaşık sayının orijine olan mesafesini bulsun.
5. Bir metod daha yazalım o da iki karmaşık sayı arasındaki mesafeyi bulsun. Mesafe(x,y).
6. Üçüncü mesafe adlı metodumuzun parametresi de Nokta tipinde nokta olsun. Bu nokta zaten bir karmaşık sayıyı ifade ediyordu. Bir de constructor oluştururken bir karmaşık sayı tanımlamış olacağız. Bu son metodumuz bu ikisi arasındaki mesafeyi bize verecek. Mesafe(nokta, nokta).

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

İlk metodu oluşturduk buraya kadar problem yok ancak en altta görünen ikinci mesafe metodumuzu şu mantıkla oluşturmamız lazım:



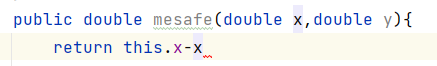
İlk karmaşık sayıda(3+4i sayısı) x ve y eksenleri belli. İkinci karmaşık sayıda (2+3i sayısı),

İkisinin x ve y eksenindeki izdüşümlerini birbirinden çıkar; (4 den 2 yi; 7’den 3’ü çıkar, elde 2 ile 4 kalıyor.)

Eldeki verilerin karesini al. Bunları topla ve karaköke al:

Başlıyoruz;

1. önce x eksenlerini çıkaralım: Dikkat birinci x, sayfanın başındaki parametremiz, ikinci x ise metodun içine yazdığımız x parametresi.



1. karesini almak için çarpalım (Math.pow la da yapabilirdik):

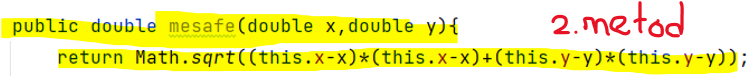
metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. şimdi de y eksenini çıkarıp karesini alalım ve elde edilen verileri toplayalım:



1. Son olarak tüm işlemin karakökünü alalım:



Şimdi de üçüncü metodumuzu yazalım: Nokta Class ından oluşturduğumuz nokta nesnesinden devam edeceğiz. Dikkat: bu noktanın da artık bir x değeri var, nokta.x onu karşılıyor: Yukarıdaki aynı işlemi yapacağız tek farkı:

* ikinci x ve y eksen değerlerimiz artık nokta.x ve nokta.y değerleri:



Şimde Main sayfasına gelelim ve ks1 diye bir nesne oluşturalım ki bu nesne Nokta Class ından özelliklerini alacak:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

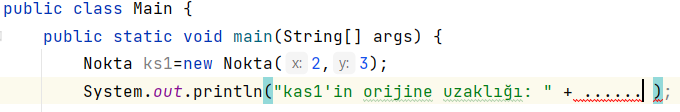
Burada parametreleri constructor ın içine yazarak construct edeceğiz:

Örnekteki gibi 2 ve 3 olsun:

metin, portakal içeren bir resim

Açıklama otomatik olarak oluşturuldu

Yazdıralım şimdi, sout derken metodu getirmeliyiz. ks1 2+3i sonucunu dönmeli.



Nokta yazdığım yere ne getirmeliyiz?

Ks1 nesnesinin mesafe metodunu getireceğiz: mesafe metodu 3 taneydi. Bunlardan ilki olan parametre almayan (içi boş olan) mesafe metodunu getireceğiz zira şu an ks1 in orijine uzaklığını ölçüyoruz.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şimdi, bir diğer karmaşık sayı oluşturup bu ikinci karmaşık sayı ile uzaklığını ölçeceğiz. Bunu İkinci metodda şöyle formüle etmiştik:



Şimdi onu Main de çağırırken iki parametreyi sout içinde çağırdığımız metodun içine yazarak çağırıyoruz. Örneğin 6 ve 11 değerleri olsun.

metin içeren bir resim

Açıklama otomatik olarak oluşturulduŞimdi doğrudan bizim oluşturduğumuz nesneyi parametre olarak alan metodu kullanalım: Üçüncü metot:

Dolayısıyla bir karmaşık sayı oluşturmamız gerekiyor, üçüncü karmaşık sayımız (ks3): Sonrasında da metodu yazarken şunu seçeceğiz:



Şimdi son hali:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Sonuç olarak; karmaşık sayı kategorisinde bir veri tipi yoktu, bu çalışmada onu oluşturduk.