

1. Soru) Aşağıdaki Java ifadelerinin her birinin tip ve değerini yazınız. Eğer bir ifade derlenemezse, ilgili ifadenin tip kısmına **Hatalı** yazınız ve değer kısmına da bir **X** koyunuz. Her bir kutucuğu doldurmanız gerekmektedir. Boş bırakılan hücreler yanlış olarak değerlendirilecektir (7 puan).

<i>İfade</i>	<i>Tip</i>	<i>Değer</i>
<code>(7 / 2) * 2.0</code>	double	6.0
<code>(7 / 2.0) * 2</code>	double	7.0
<code>!(!!false && !!true)</code>	boolean	true
<code>"1.1" + "2.2"</code>	String	"1.12.2"
<code>(1 < 2 < 3)</code>	Hatalı	X
<code>1 + 2.2 + "1.1" + "a"</code>	String	"3.21.1a"
<code>Integer.parseInt("123")</code>	int	123

2. Soru) Dizi içerisinde arama yapan `indisAra` adında bir metod yazınız. Bu metodun parametresi, tamsayı değerleri tutan bir dizidir. Bu dizi içerisinde 100 sayısı birkaç indiste kayıtlıdır. `indisAra` metodu, dizideki en son kayıtlı 100 sayısının indisini geri döndürmektedir. (7 Puan)

Cevap:

```
public int indisAra(int[] dizi){  
    int index=-1;  
    for (int i = dizi.length-1; i >=0 ; i--) {  
        if (dizi[i]==100) {  
            index= i;  
            break;  
        }  
    }  
  
    return index;  
}
```

3 Soru) 2 boyutlu düzlem için aşağıdaki kurallara göre Nokta sınıfını yazınız.

- Nokta sınıfının, dışarıdan erişilemeyen int tipindeki global değişkenleri x ve y dir. Sınıfın dışından bu değişkenlere ulaşmak için gerekli metotları yazınız (setter ve getter) (2 Puan).
- 2 tane yapıcı metot yazınız. Birinci yapıcı metodun parametresi yoktur ve bu metot x ve y değişkenlerini sıfıra eşitlemektedir. Yani noktayı $(0, 0)$ 'a yerleştirir (2 Puan).
- İkinci yapıcı metodun ise iki tane parametresi vardır. İlk parametrenin adı x ve ikinci parametrenin adı y dir. Bu metot, parametre değerlerini ilgili global değişkenlere atamaktadır (3 Puan).
- `sagdami` adında static bir metot yazınız. Bu metodun erişim belirtecini sadece aynı pakettekiler erişebilecek şekilde ayarlayınız. Bu metodun Nokta tipinde 2 tane parametresi vardır. Bu metot, birinci parametredeki Nokta nesnesinin düzlemde ikinci parametredeki Nokta nesnesinin sağında olup olmadığını kontrol eder. Eğer sağında ise `true`, diğer durumlar için `false` değerini döndürür (5 Puan).
- `uzaklik` adında bir metot yazınız. Bu metodun erişim belirtecini her yerden erişebilecek şekilde ayarlayınız. Bu metodun Nokta tipinde 2 tane parametresi vardır. Bu metot, verilen iki Nokta arasındaki uzaklığı bulmaktadır ve sonucu geri döndürmektedir (5 Puan).

Cevap:

<pre>public class Nokta { private int x; private int y; public Nokta(){ x=y=0; } public Nokta(int x, int y){ this.x=x; this.y=y; } public int getX() { return x; } public void setX(int x) { this.x = x; } }</pre>	<pre>public int getY() { return y; } public void setY(int y) { this.y = y; } public double uzaklik(Nokta n1, Nokta n2){ int x_ = n1.x - n2.x; int y_ = n1.y - n2.y; return Math.sqrt((x_*x_) + (y_*y_)); } static boolean sagdami(Nokta n1, Nokta n2){ if (n1.x > n2.x) { return true; } return false; } }</pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

4 Soru)

- İptal etme (override) nedir? İptal etmeye basit bir örnek veriniz. (7 Puan)
- İptal etme işleminde dikkat edilmesi gereken hususlar, kurallar nelerdir? (5 Puan)
- Bir metodun iptal edilmesini engellemek istiyorsak ne yapmamız gerekir? (2 Puan)

Cevap:

- Üst sınıf içerisinde tanımlı bir metodun, bu sınıftan türetilen alt sınıf içerisinde tekrardan aynı isim ve yapı ile tanımlanmasına iptal etme (override) denir.

```
class Kedi {  
    public void yakalaAv() {  
        System.out.println("Kedi sinifi Av yakaladi");  
    }  
}  
  
public class Kaplan extends Kedi {  
    public void yakalaAv() {  
        System.out.println("Kaplan sinifi Av yakaladi");  
    }  
}
```

- Üst sınıfa ait private erişim belirleyicisine sahip olan alanlara ve metotlara, türeyen alt sınıf tarafından kesinlikle erişilemez.
 - Aynı şekilde türeyen alt sınıf, türetildiği ana sınıf ile aynı paket içerisinde değilse, ana sınıfa ait friendly erişim belirleyicisine sahip olan alanlara ve metotlara erişemez, sadece protected erişim belirleyicisine sahip olan alanlara ve metotlara erişebilir.
 - İptal eden metot, iptal edilen metodun erişim belirleyicisi ile aynı veya daha erişilebilir bir erişim belirleyicisine sahip olması gerekir.
- Bir metodun iptal edilmesini engellemek istiyorsak metodu final yapmalıyız.