

## HW\_2\_SONUC\_RAPORU

**MÜH.TĞM. MUHAMMET YURTSEVEN**

**Anlaşılmayan** : Soru 6, Soru 19,Soru 28, Soru 35, Soru 43, Soru 46 ,Soru 48

Soru 1- Cevap A : Java'da Aynı tip değişkenler kullanacaksak aralarına virgül koyarak tek satırda hepsini tanımlayabiliriz. Ama farklı Tip Değişkenler Aynı Satırda Tanımlanmasına java izin vermemektedir. Bu yüzden A seçeneği derlenmez.

Soru 2 Cevap D

Soru 3- Cevap B : String boş bırakıldığı zaman bir değere sahiptir o da NULL olarak atanır.

Soru 4 - Cevap B : Değişken isimleri sayı ile başlayamaz.

Soru 5 - Cevap B : Java'da , sınıf adları kurallara göre büyük harfle başlar. Devamında yeni kelimeler dışında küçük harf kullanılır. A seçeneği , değişken tanımlama şeklidir. C seçeneği , Sabit değeri değişmeyen değişkenleri için idealdir. D seçeneği hiçbir Java kuralına uymaz.

<https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-101/java-yazim-kurallari-ve-onerileri>

**Soru 6 –**

Soru 7 – Cevap C : Sayısal değişkenlerde iki basamak arasında alt çizgi konulabilir. Sayısal sabitlerin rakamları arasındaki herhangi bir yerde, istenen sayıda alt çizgi karakteri kullanılabilir.Bu özellik sayısal sabitlerin okunurluğunu kolaylaştırır. Değişmez değerin başında veya sonunda alt çizgilere konulmasına izin verilmez.

[https://www.bilgigunlugum.net/prog/java/2java\\_veri](https://www.bilgigunlugum.net/prog/java/2java_veri)

Soru 8 – Cevap C- Wrapper Classes(Sarmalanmış Sınıflar) olarak geçer. 2 tür değişkenimiz vardır. Obje ve İlkel değişkenleridir. İlkel değişkenler , sarmalanınca ,obje olmaktadır. Bu nedenle cevap A seçeneğidir. <https://malikmasis.blogspot.com/2014/12/wrapper-classes.html>

Primitive	Wrapper Class	Constructor Arguments
boolean	Boolean	boolean or String
byte	Byte	byte or String
char	Character	char
double	Double	double or String
float	Float	float, double, or String
int	Integer	int or String
long	Long	long or String
short	Short	short or String

Soru 9- Cevap C

Soru 10- Cevap C : Java'da yeni bir nesne oluşturmak için **new** anahtar kelimesini kullanırız.

Soru 11- Cevap D : Float çift duyarlı olmadığından dolayı 5.0 işlemlerde yanlış olabileceğinden dolayı seçenek D'dir.

Soru 12- Cevap A- Küçükten büyüğe sıralama A seçeneğinde doğrudur.

Soru 13- Cevap D- Constructor, instance variables, method names bunlar herhangi bir sırayla görülebilir.

Soru 14 Cevap B : Java farklı değişkenleri aynı satırda virgül ayırarak tanımlamaya izin vermez.

Soru 15- Cevap C- Satır 2-7 başlatıcıları gösterir. Satır 6 static bir başlatıcıdır. 3-5 ise constructordur.

Soru 16- Cevap A- defaultValue local bir değişken olduğundan dolayı otomatik olarak başlatılamaz. Bu herhangi biriyle derlenmeyeceği anlamına gelir. Eğer defaultValue bir 0 ve 0.0 gibi değeri olsaydı o zaman derlenirdi.

Soru 17- Cevap A- Finalize bir kez ve sadece bir kez, her nesne için çalışır.

[https://www.dijitalders.com/icerik/44/5159/java\\_class\\_yapisinda\\_finalize\\_metotunun\\_kullanimi.html](https://www.dijitalders.com/icerik/44/5159/java_class_yapisinda_finalize_metotunun_kullanimi.html)

Soru 18- Cevap D- String ilkel bir sınıftır.

**Soru 19- Cevap**

Soru 20- Cevap C- Çünkü byte ve short ondalıklı sayı içermez. Float seçeneğide doğrudur ama double çift duyarlı olduğundan dolayı C seçeneği daha uygundur.

Soru 21- Cevap B- k2 derlenmez. Çünkü böyle bir değişken tanımlamaya java izin vermez.

Soru 22- Cevap B- foo.bar gösterimi, hem örnek değişkenlerini okumak hem de yazmak için kullanılır. Local değişkenlere referans vermek için kullanılamaz.

Soru 23- Cevap C- Alt çizgi ve dolar işareti kullanılabilir. Sayılara izin verilir, ancak tanımlayıcının ilk karakteri olarak girilmez. Bu nedenle, Doğru seçenek C'dir.

Soru 24- Cevap D- Alt çizgi yalnızca iki basamak arasında görünebileceğinden, sayıda noktadan önce alt çizgi kullanılması hatalı olduğundan dolayı bu geçersizdir.

Soru 25- Cevap C- Local değişkenlerin varsayılan başlatma değeri yoktur. Bir değere ayarlanmadan önce başvurulursa, kod derlenmez. Bu nedenle, Seçenek C doğrudur. Değişken bir örnek değişkeni olsaydı, Seçenek B doğru olurdu. D seçeneği olmama sebebi ise Local değişken, herhangi bir yere başvurulmuyorsa veya başvurulmadan önce bir değere atanmışsa başlatma olmadan derlenebilir.

Soru 26- Cevap C- int,long,short da değer olarak o döndürür ve double o.o döndüreceği için cevap C'dir.

Soru 27- Cevap B- Bir ilkel ögeyi, yalnızca atayarak bir wrapper sınıf nesnesine dönüştürebilirsiniz. Ama bir wrapper nesnesini ilkele döndüremeyiz.

#### Soru 28- Cevap

Soru 29- Cevap D- Bir constructur çağırmak için yeni bir nesne üretmek gerekir bundan dolayı en uygun D seçeneği.

Soru 30- Cevap A-

I.String cat = "animal", dog = "animal"; //Doğru şekilde tanımlanmış

II.String cat = "animal"; dog = "animal"; // dog veri tipi belirtilmesi gerekir.

III.String cat, dog = "animal"; // cat bir değişken atılmamış.

IV.String cat, String dog = "animal"; // bir satırda virgülle ayrılıyorsa iki tane tip yazılmaz.

Soru 31- Cevap C- int > Integer ve char>Character olduğundan dolayı Cevap C'dir.

Soru 32- Cevap A- String null bir değer atanabilir.

Soru 33- Cevap A- İlkel türler küçük harfle başlar. Null değeri alamazlar. İlkel türler büyük harfle başlamaz. Kendimiz ilkel tür oluşturamayız.

Soru 34- Cevap D- Java'da bellek yönetimi arka planda JVM ve onun içinde yer alan Garbage Collector(Çöp Toplayıcısı) ile yapılır. Bu aslında Java'yı Java yapan özelliklerden biridir. Çünkü bir geliştirici olarak endişe duymadan yeni nesneler yaratırsınız ve GC sizin için memory allocation ve deallocation işlemlerini yapar. Yani JVM bunu sizin için yönetir. C/C++ gibi dillerde bunun kontrolü sizin elinizdedir. Bu durum yazılımda hata çıkma ihtimalini artırır ve geliştirme sürecini uzatabilir. Bu durumdan dolayı Cevap D.

<https://medium.com/@tugrulbayrak/jvm-garbage-collector-nedir-96e76b6f6239>

#### Soru 35- Cevap

Soru 36- Cevap B : Yeni tanımlanan sınıf nesnesi büyük harfle başlar bu nedenle Cevap B'dir.

Soru 37- Cevap B

Soru 38- Cevap C : int bir ilkedir, bu yüzden null değeri ile derlenmez. Integer ve String her ikisi de nesnedir ve bu nedenle null atanabilir.

Soru 39- Cevap C : Instance yalnızca sınıftaki veriler ulaşabilir. Static herhangi bir yerden ulaşılabilir.

Soru 40- Cevap B : Sayısal değeri iki basamak arasında alt çizgilere izin verilir. Ondalık noktaya bitişik alt çizgilere izin verilmez.

Soru 41- Cevap A : byte < short < int < long

Soru 42- Cevap A- Çağırma işlemi nokta konularak yapılır. Burada name değişkeninde cat.name şeklinde çağırılması gerekmektedir.

**Soru 43- Cevap**

Soru 44- Cevap A- Çağırma işlemi nokta konularak yapılır. Bu p.beakLength = b; bu şekilde atama işlemi gerçekleştirilir.

Soru 45- Cevap B- Integer.**valueOf()** ile Integer.**parseInt()** metotları arasında temel fark **valueOf()** metodu Integer tipinde bir nesne, **parseInt()** metodu ise int tipinde bir veri tipi döndürür. **parseInt()** metodu temel veri tipi döndürdüğü için daha hızlı çalışır.

**Soru 46- Cevap**

Soru 47- Cevap C- A ve B seçenekleri constructor'den ziyade bir static yöntemlerdir. D seçeneği aynı yöntemdir. Bu nedenle C seçeneği uygun olanıdır.

[https://www.dijitalders.com/icerik/44/1997/java\\_class\\_yapisinda\\_constructor\\_ve\\_overloading.html](https://www.dijitalders.com/icerik/44/1997/java_class_yapisinda_constructor_ve_overloading.html)

**Soru 48- Cevap**

Soru 49- Cevap B : Her wrapper sınıfının eş değeri olan bir kurucusu vardır. A, C ve D'de belirtilen yöntemler mevcut değildir.

Soru 50- Cevap C