

Java Kod Yapısı ve Yazım Standartları

Dilin yazım biçimine Syntax yani söz dizimi denir. Bu sizin yaklaşımdan çok kod yazma yapınızı belirler. Bu anlamda her dilde olduğu gibi Java' da da bazı kurallar vardır ve bu kurallara uymamak derleme hatasına neden olur. Daha açıklayıcı olması açısından aşağıda maddeler şeklinde belirttim.

C/C++ türevi bir dildir

Java daha öncede bahsettiğimiz üzere C/C++ türevi bir dildir. Bu yüzden C, C++, Java Script, C# veya benzeri dil yapısına sahip uygulama geliştiricilerin kod yapısı olarak Java' ya alışmaları çok daha kolay olmaktadır.

Sınıflardan oluşur

Java da yazdığınız uygulamalar sınıflardan oluşur ve sınıflar içerisinde olmak zorundadır. Bunlar haricinde interface, enum, annotation gibi tanımlar olsada asıl iş yapan sınıflardır.

Satır sonları ; ile biter

Yine tüm C/C++ türevi diller gibi Java' da da satır sonları noktalı virgüle ";" ile biter. Eğer Basit türevi bir dil kullansaydık burada her bir satır geçişi yeni satırı ifade ederdi. Örnek olarak aşağıdaki iki kod bloğu arasında görüntü olarak fark olsada çalışma şekli olarak bir fark bulunmamaktadır.

```
public class Ornek {  
    public static void main(String[] args)  
    {  
    }  
}
```

```
public class Ornek {  
    public  
    static  
    void  
    main(String[] args) {  
    }  
}
```

Büyük küçük harf duyarlıdır (case sensitive)

Yine tüm C/C++ türevi diller gibi Java' da case sensitive yani büyük küçük harf duyarlıdır.

Örneğin Java anahtar kelimeleri büyük küçük harfe dikkat ederek yazmalısınız. Eğer sınıf tanımlarını class anahtar kelimesi yerine Class şeklinde yaparsanız bu bir derleme hatasına neden olur. Bir başka örnekte int x diye tanımladığınız bir değişkeni X olarak çağıramazsınız. Bu tanımlar Java tarafında farklı olarak görünür.

Değişkenler tip belirtilmeleri ile birlikte yapılmalıdır

Java içerisinde kullanacağınız değişkenler öncelikle tanımlanmalıdırlar. Bu tanımlar dinamik olmamakla birlikte tip belirtilmelerini taşımalıdırlar.

Örneğin eğer x adında bir değişken kullanmak istiyorsanız bu değişken **int x** şeklinde tanımlı olmalıdır.

Aynı kapsamda aynı isim iki değişken tanımlamaz

Java içerisinde tanımladığınız bir değişken reserve edilmiş sayılır ve bu isimde bir başka değişken tanımlanamaz.

Örnek vermek gerekirse main metodu içerisinde int x isminde bir değişken tanımlanmışsa bu x değişkeninin geçerli olduğu alanda herhangi bir tipde x değişkeni tanımlanamaz.

Rezerve kelimeler değişken tanımı olarak verilemez

Java' ya özgü rezerve kelimeler değişken tanımlarında kullanılamaz.

Örneğin if kelimesi koşulu ifade eder ve Java içerisinde bir anlam ifade ettiğinden bir değişken tanımı olarak verilemez.

Java Rezerve Kelime Listesi

abstract	continue	for	new	switch
assert	default	goto	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while

İsimlendirme kullanımları

Java içerisinde Sınıf, değişken ve nesne gibi tanımlar için olmazsa olmaz kurallar vardır. Bunlar aşağıdaki gibi listelenmiştir.

- Hiç bir tanım sayı ile başlayamaz.
- Tanımlar içerisinde " ? # , ; gibi özel karakterler kullanılamaz.
- Revize kelimeler kullanılamaz

Sınıf isimleri dosya ismi ile aynı olmak zorundadır

Java' oluşturduğunuz sınıflar dosya isimleri ile aynı olmak zorundadır. Örneğin Ornek.java isminde bir Java dosyası varsa bunun sınıfı büyük küçük harf duyarlı olmak üzere Ornek olmalıdır.

Kod blokları

Java içerisinde kod blokları { yani süslü parantez şeklinde belirtilir. Alan belirtimi yapıyorsanız bu alanların kapsamları } karakterleri içerisinde alınır.

Örneğin if koşulunun kapsadığı alanı belirtmek istiyorsak aşağıdaki gibi bir tanım yapılmalıdır.

```
if(true){  
    ...  
}
```

Açıklama satırları

Tüm dillerde olduğu gibi Java’ da da açıklama satırları bulunmaktadır. Bunlar tek satırlık ve çok satırlık açıklamalar olabilir.

Aşağıda tek satırlık bir açıklama bulunuyor. Burada tek bir satırın açıklaması // yani çift slash işareti ile yapılıyor.

```
public class Ornek {  
    public static void main(String[] args) {  
        //Bu bir açıklama satırıdır...  
    }  
}
```

Aşağıdaki örnekte ise birden fazla satırın açıklaması bulunuyor. Bu ise /* ile başlayan ve */ ile biten alanları açıklama satırı haline getiriyor.

```
public class Ornek {  
    public static void main(String[] args) {  
        /*Bu  
        bir  
        açıklama  
        satırıdır...  
        */  
    }  
}
```

Kod yazım kuralları

Java uygulama geliştiricileri için zorunlu olmayan ancak bir standart haline gelmiş bazı kurallar vardır. Bu kurallar uymak zorunlu olmamasına karşın doğru ve okunabilir kod yazımını sağlarlar.

- Camel-Case denen deve hörgücü şeklinde bir isimlendirme şekli kullanılır. Buna göre ikinci kelimeler her zaman büyük harf ile başlar.
 - BuDenemeAmacliBirSinifTanimi
 - buDenemeAmacliBirMetodTanimi
 - buDenemeAmacliBirDegiskenTanimi
- Sınıf isimleri büyük harf ile başlar
- Metod isimleri küçük harf ile başlamalıdır.
- Değişken isimleri küçük harf ile başlamalıdır.
- Sabitler yani static ve final olan değişkenler tamamen büyük harflidir ve kelimeler kendi aralarında _ yani alt çizgi ile ayrılırlar.

Not: Geliştirme boyunca bu standartlara uymanız sizin daha okunabilir ve doğru kod yazdığınızı gösterir.