Merhabalar,

Bu anlatımda Java nın temelleri olan veri tiplerini ve özelliklerini kod örnekleri ile anlatacağım.Java ile masaüstü uygulaması ,Android Uygulaması , Web Uygulaması , Web Servisi ,Java EE Uygulamları , Java ME ne isterseniz yazın temellerini bilmeden iyi öğrenmeden bahsettiğim konulara geçemezsiniz.Veri türlerini hemen hemen her yerde kullanacağız.Peki nedir bu veri türünün anlamı ? Programlama yaptığımız cihazımızın üzerinde hafızada tutulmasını istediğimiz , geçici verileri tutma araçları diyebilriiz.Nasıl yani derseniz gerçek hayattan örnekleyelim … diyelim ki buzdolabınıza koymak istediğiniz meyveleriniz var ancak bu meyvelerin her birini ayrı kaplara koyarak psikopattça düzen oluşturmak , gerektiğinde yemek istiyorsunuz.Elma için elmaKabi armut için armutKabi diyelim.(isimlendirmelere dikkat , Örnekler Çoğaltılabilir).Gerektiğinde bu kabı açıp yersiniz veya çürüdüğünde yani değiştiğinde çöpe atarsınız veya elektrikleri uzun süreli kesilir çöpe atmak durumunda kalırsınız ancak bir sonraki kullanım için kablarınızı yıkar yine dolaba koyarsınız..Veri türleri de buradaki elma armut kabına benzetebiliriz verilerimizi gecici süre bu kablarda tutabiliriz.Veri türlerimizi isimlendirirken ilk harfi küçük var ise ikinci kelime ve sonraki kelimelerin ilk harfleri büyük ,bitişik , türkçe karakter (ğ,ş,İ,ı,ç,ö,ü) ve noktalama işaretleri olmadan isimlendiririz. İsimlendirmenin ilk harfi rakam içeremez 2elmaKabi gibi ancak elmaKabi2 şeklinde isimlendirebiliriz.Hadi şimdi veri tiplerini ve özelliklerini görelim.

Veri tiplerini isimlendirmekten bahsetmiştik.Şimdi veri tiplerinin Java da nasıl yazıldığını görelim ;

```
veriTipi değişkenAdi = değişkenDeğeri;
```

Gördüğünüz gibi biraz sonra açıklayacağım veri tipi türü arada bir boşluk ve değişken isimlendirmemiz gerekli ise değişlen atamasını = işareti ile yapıp ; ile değişken tanımlama kod parçamızı sonlandırdık.Bunu tanımları öğrenirken örnekleyeceğiz …

Java dili içerisinde ana olarak veri tipleri iki bölüme ayrılır ;

İlkel(Primitif) teri tipleri
Referans tipler

İLKEL(Primitif-Temel) VERİ TİPLERİ

Java dilinde ki ilkel veri tipleri aşagıda ki gibidir ;

Tamsayılar
Ondalıklı Sayılar
Karakterler
Mantıksal tipler

İlkel Veri Tipi	Uzunluk	Minumum Deger	Maksimum Deger	Gömüldügü Sınıf	Varsayılan Deger
char	16 bit	unicode 0	unicode 2^{16}-1	Character	'\u0000'
long	64 bit	-2^{63}	2^{63}-1	Long	0l
int	32 bit	-2^{31}	2^{31}-1	Integer	0
short	16 bit	-2^{15}	2^{15}-1	Short	0
byte	8 bit	-2^{7}	2^{7}-1	Byte	0
double	64 bit	$-3.4*10^{38}$	$3.4*10^{38}$	Double	0.0d
float	32 bit	$-1.7*10^{38}$	$1.7*10^{38}$	Float	0.0f
boolean	1 bit			Boolean	false

İlkel veri tipleri arasında sadece **Tamsayıları** içeren verilerimizi tutan 3 çeşit veri tipimiz vardır.Bunlar ;

byte
short
int

byte : byte en küçük tam sayı tipidir. Büyüklüğü 8 bit tir ve *-128 ile +127* arası değer alır. Kod geliştirirken ***byte*** anahtar kelimesi ile tanımlama yapılır.

Örnek Kod

```
byte byteVeriTipi = 127;
byte byteVeriTipi = -128;
```

Buradaki -128 ve 127 fazlası ve/veya eksiğini girdiğimizde kullandığımız IDE miz hata verecektir.Tek veri tanımlamasında yani tek elmamızın elma kabına sığmayıp taşması kaba sığmaması gibi düşünebiliriz.Bu örneği diğer anlatacağım tiplere de uyarlayabilriiz.

short : Büyüklüğü 16 bit olan short veri tipi -32768 ile +32767 arasında bir değer alabilir. Kod geliştirirken short anahtar kelimesi ile tanımlama yapılır.

Örnek Kod

```
short shortVeriTipi = 85;
```

int : En çok kullanılan veri tipidir. 32 bitlik veri tipi, -2.147.483.648 ile 2.147.483.647 arasında bir değer alabilir. Kod geliştirirken int anahtar kelimesi ile tanımlama yapılır.

Örnek Kod

```
int integerVeriTipi = 1500;
```

ONDALIKLI SAYILAR

Burada ki veri tiplerini ondalıklı sayılar için kullanırız ve bellekte tuttukları alana göre 2 çeşit ondalıklı sayılar için kullanabilecegimiz ilkel veri tipi vardır.Bunlar ;

float
double
NOT : Java ondalıklı sayıları bilgisayar ortamında depolayabilmek için java dili standartlarından IEEE 754 olanını benimser.

float : float veri tipi 32 bitlik büyüklüğe sahiptir ve 1.4×10^-45 ile 3.4×10^38 aralığında bir değer tanımlanabilir. float olarak belirlenmiş olan veri tipine integer bir değer atandığı takdirde java tarafından direk olarak 1.0 şeklinde algılanır. Geliştirme aşamasında hata oluşmaz. Fakat float veri tanımlarken (.) noktadan sonra değişken değerinin sonuna “f” veya “F” koyulmadığı takdirde geliştirme esnasında hata alınacaktır. Sebebi ise javanın bu

değişkeni double olarak algılayacak olmasıdır. Aşağıdaki örneklerle konuya biraz daha açıklık getirmeye çalışalım.

Örnek Kod

```
float floatVeriTipi= 10.655555555555555f;
float floatVeriTipi2 = (float)10.5;
```

Bu örnekte iki kod parçacığı görüyoruz.İkinci örnekte f koymadan casting işlemi ile tanımlama yaptık ileride bu konuya detaylı değineğim.

double : double veri tipi 64 bitlik büyüklüğe sahiptir ve 4.9×10^-324 ile 1.8×10^308 arasında bir değer tanımlanabilir.

Örnek Kod

```
double doubleVeriTipi= 5.545;
```

KARAKTERLER

1.)char : Java da karakterler char veri tipi içinde saklanır. C/C++ gibi yazılım dillerinde char veri tipi 8 bitlik büyüklüğe sahipken, Java'da 16 bitlik büyüklüğe sahiptir. Bunun sebebi Java'nın Unicode karakter setini kullanıyor olmasıdır. Ayrıca Java ASCII kod yapısını da desteklemektedir. Kod geliştirirken char anahtar kelimesi ile tanımlama yapılır. Char ilkel veri tipinde bir degişken tanımlayıp buna da değer atarken değeri tek tırnak(') içinde atamamız gerekir

NOT : char veri tipine sahip degişkene maxsimum bir karakter verebiliriz. Örnegin. ;

Örnek Kod

```
char charVeriTipi= 'S';         /Hata Vermez
char charVeriTipi= 'Serkan';    /Hata Verir !
```

NOT :String ilkel(primitif) veri tipi **degildir**, String karakter kümelerini(katarlarını) tutmak için tasarlanmış bir sınıfdır.

MANTIKSAL TİPLER

boolean : Mantıksal işlemlerin sonuçlarını tutmak için kullanılan ve sadece true veya false diye iki deger alabilen veri tipidir.Bellekte 1 bit yer kaplarlar.Durumun doğru ya da yanlış olduğuna göre yapılacak işlemlerde kullanılır.

Örnek Kod

```
boolean booleanVeriTipi=true;
```

REFERANS TİPLERİ

Nesneye yönelimli programlama dilinin temeli sınıf(class) mantıgına dayanır ve java da veri tipi olarak sınıflarımız vardır.Bu veri tiplerinden bahsedecek olursak bizim ilkel(primitive) veri tiplerimizin birer tane gömüldügü sınıf vardır.Bunlar Byte,Short,Integer,Double,Character,Float,Double,Boolean sınıflarıdır peki diyebiliriz ki ilkel veri tipleri varsa neden sınıf olan veri tiplerine ihtiyacımız var ya da sınıf veri tiplerimiz varsa neden ilkel veri tiplerine ihtiyacımız var bunların farkı nelerdir,İlkel veri tipleri ile sınıf veri tiplerinin farkını int ilkel tipi ile Integer veri tipi arasında ki farkı göstererek anlatmaya çalışım;

İnt bir primitif veri tipdir **Integer** ise bir nesnedir
Integer bir degişken oldugu için **null** degeri verilebilir ama int'e verilmez.
Integer tipinde bir degişken java.lang.**Integer** sınıfında ki metotları kullanabilir, **int** ise kullanamaz.
Integer tipinde ki degişkenleri bir vektör ya da koleksiyon içerisinde tutabiliriz ama **int** ile bunu yapamayız.
Primitif veri tipleri(**byte,short,int,double,float,double,char,boolean**) bir nesne olmadıgı için serialize(serileştirme) işlemine tutamayız.

Integer gibi sınıflar immutable (değişmez) özelliğe sahiptirler, sıradan nesneler gibi davranmazlar.
Aynı benzer durumlar **byte-Byte,short-Short,int-Integer,float-Float,double-Double,char-Character,boolean-Boolean** için de geçerlidir.

KARAKTER KÜMELERİ(Katarları)

String : Karakter kümelerini (Kelimeleri)yani birden fazla karakterin bir araya gelerek oluşturdugu degişkenler için kullanılan veri tipidir.String veri tipi primitif veri tipi olmadıgını söylemiştik String karakter kümelerini tutmak için tasarlanmış özelleşmiş bir sınıftır.

NOT : String veri tipinde bir degişken tanımlayıp buna da deger atarken degeri çift tırnak(“) içinde atamamız gerekir.

Örnek Kod

```
String stringVeriTipi= "Serkan Kaya";
```

Tanımlamalarımızı yaptık bunların hepsini ezberlemenize gerek yok.Önemli olan gerekli yerde gerekli veri tipine karar verebilmektir.Ne gibi diyecek olursanız örneğin ödeme ile ilgili veri tutuyor iseniz ondalıklı sayıları kullanmanız gereklidir.Sıfırdan sonrası bir süre sonra dağ olabilir

Sonraki Yazım İçin Tıklayın(IDE nedir ? IDE açılımı Nedir ? JDK Nedir ? JRE Nedir ? JVM Nedir ? SDK Nedir ?)

Umarım faydalı olmuştur

Serkan Kaya
Full Stack Java Developer