

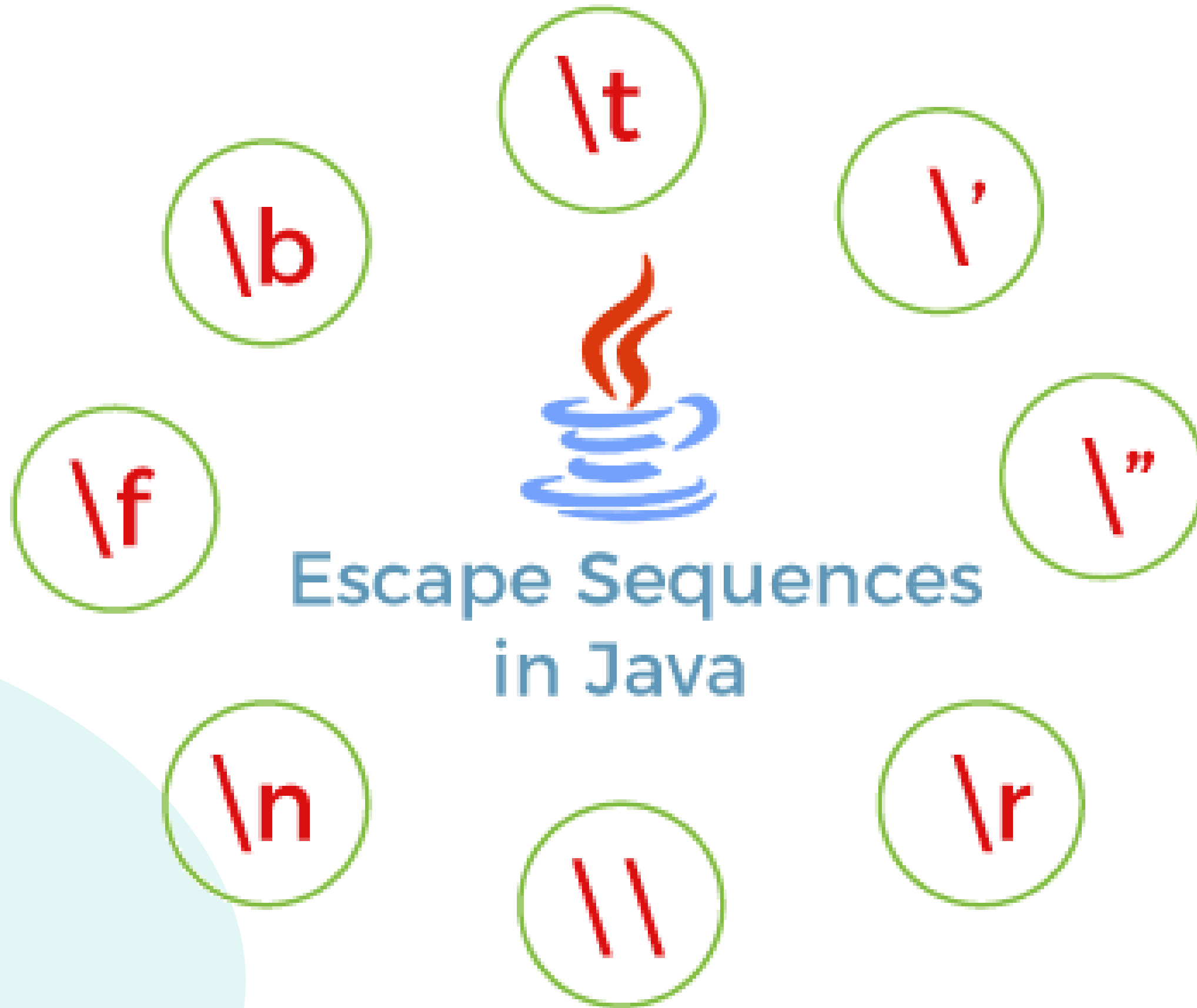
TECHNO **STUDY** +

MENTORİNG

ESCAPE CHARACTER, DEĞİŞKENLER, DATA TYPES

ESCAPE CHARACTER

TECHNO
STUDY +



ESCAPE CHARACTER

Kaçış karakterleri (escape characters), Java'da ters bölü (/) işaretiyle öncelenen karakterlerdir. Bu karakterler harfleri, rakamları, noktalama işaretlerini vb. içerebilir. Kaçış karakterleri çift tırnaklar ("") içinde olmalıdır. Bu geçerli karakter sabitleridir. Java derleyicisi, bu karakterleri derleyiciye belirli bir anlam katan tek bir karakter olarak yorumlar.

`\n`: Yeni satır karakterini temsil eder. Bu karakter kullanıldığında, bir dizede yeni bir satır başlar.
`\t`: Sekme karakterini temsil eder. Bu karakter kullanıldığında, bir dizede bir sekme boşluğu oluşur.
`\"`: Çift tırnak işaretini temsil eder. Bu karakter kullanıldığında, bir dizede çift tırnak işareti görüntülenir.
`\'`: Tek tırnak işaretini temsil eder. Bu karakter kullanıldığında, bir dizede tek tırnak işareti görüntülenir.
`\\`: Ters eğik çizgiyi temsil eder. Bu karakter kullanıldığında, bir dizede ters eğik çizgi görüntülenir.
`\b`: Geri alma karakterini temsil eder. Bu karakter kullanıldığında, bir dizede bir karakter geriye alınır.
`\r`: Satır başına dönme karakterini temsil eder. Bu karakter kullanıldığında, bir dizede imleç bir satırın başına döner.
`\uXXXX`: Unicode karakterlerini temsil eder. Burada XXXX, dört haneli bir Unicode karakter kodu yer alır.

Bu karakter kullanıldığında, ilgili Unicode karakteri dizede görüntülenir. Bu kaçış dizileri veya karakterleri, Java programında özel karakterleri belirtmek veya düzgün bir şekilde görüntülemek için kullanılır. Örneğin, bir dizede çift tırnak işareti görüntülemek istediğinizde `\"` kullanabilirsiniz veya bir dizede yeni bir satır başlatmak için `\n` kullanabilirsiniz.

DEĞİŞKENLER

Değişkenler

Değişkenler yani “variable”lar, bilgisayarın hafızasında geçici olarak değerler tutmamıza yarar. Bir değişken şu şekilde tanımlanır:

```
int secretNumber = 5;
```

Bu tanımlamada int tam sayıları tutmak için kullandığımız data tipi, secretNumber değişkenin ismi, 5 ise değişkenin atandığı değerdir.

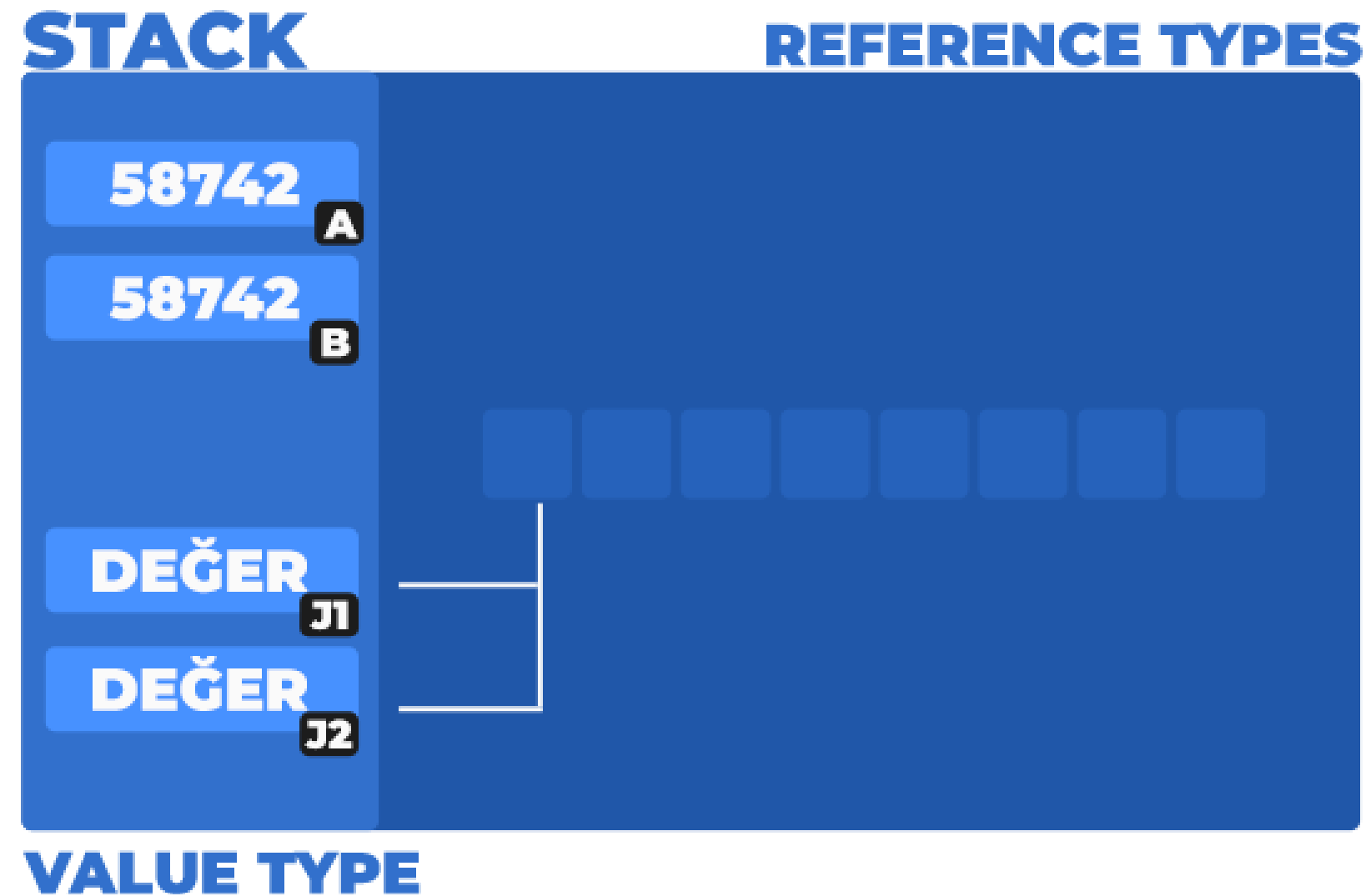
TECHNO
STUDY +

```
int a=58742
```

```
int b=a
```

```
Java j1= new Java();
```

```
Java j2= j1
```

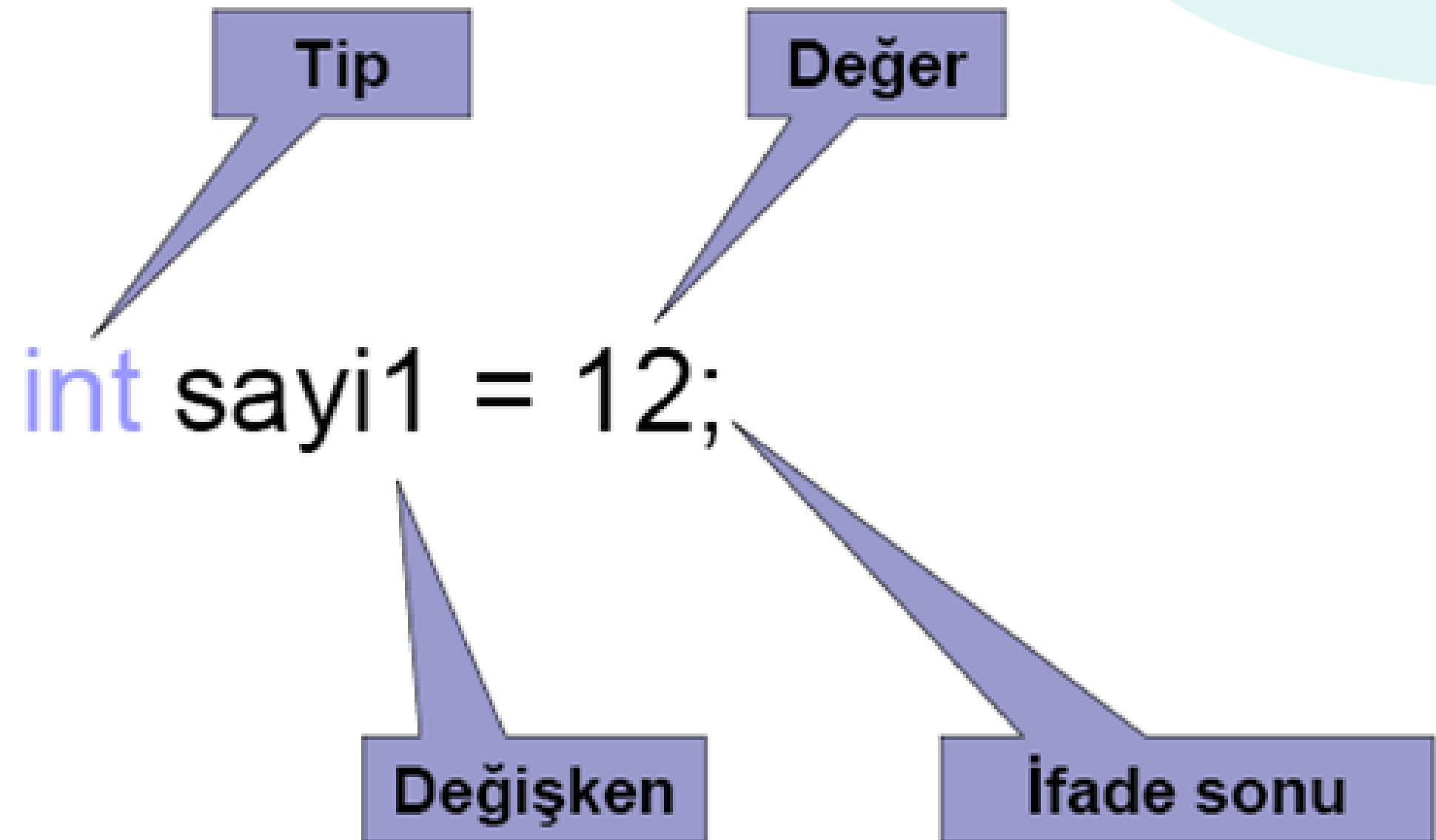


DEĞİŞKENLER

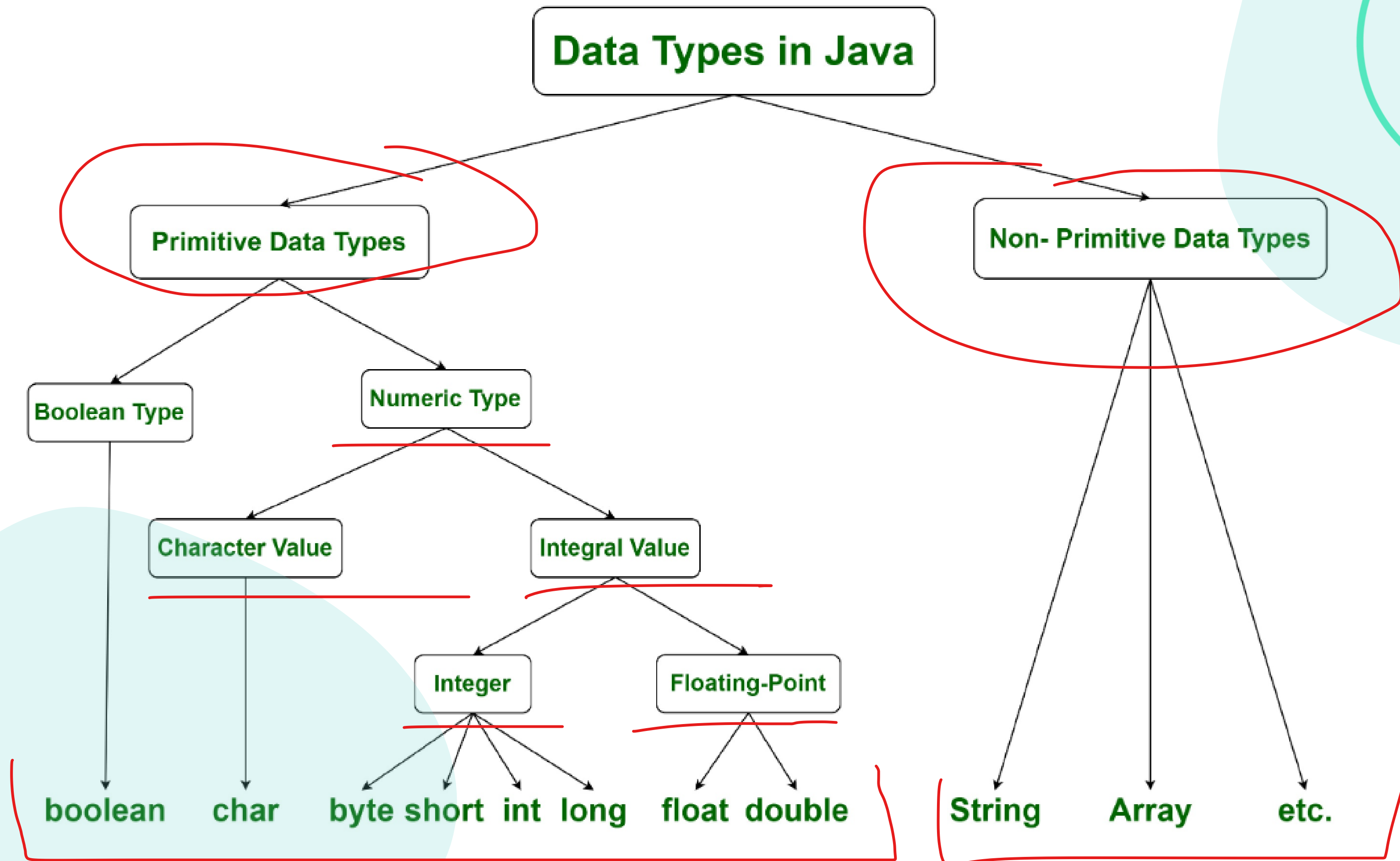
İsimlendirme

Java'da isimlendirme yaparken aşağıdaki noktalara dikkat etmekte yarar var,

1. İsim biraz uzasa bile anlamlı isim vermek.
2. İlk kelimenin ilk harfini küçük, sonrasındaki her kelimenin ilk harfini büyük yapmak. Buna camelCase deniyor.
3. Method isimlerinin camelCase olması.
4. Class isimlerinde her kelimenin ilk harfinin büyük olması. (Ör: NumberGame)



DATA TYPES



DATA TYPES

Data Tipleri

Java'da data tipleri primitive ve reference olarak iki kategoriye ayrılır:

Primitive Type

Sayılar gibi basit değerleri tutmak için kullanılır.

Java'da 8 tane primitive bulunur, bunlar aşağıdaki gibidir:

Data Tipi	Boyut	Açıklama
byte	1 byte	-128 ile 127 arasındaki tam sayıları tutar
short	2 byte	-32.768 ile 32.767 arasındaki tam sayıları tutar
int	4 byte	-2.147.483.648 to 2.147.483.647 arasındaki tam sayıları tutar
long	8 byte	-9.223.372.036.854.775.808 to 9.223.372.036.854.775.807 arasındaki tam sayıları tutar
float	4 byte	Küsürlü sayıları tutar. Noktadan sonra 6 veya 7 basamak alır
double	8 byte	Küsürlü sayıları tutar. Noktadan sonra 15 basamak alır
boolean	1 bit	True (doğru) ve false (yanlış) değerlerini tutar
char	2 byte	Tek bir karakter, harf veya ASCII değeri tutar

DATA TYPES

Non-primitive veri tipleri, nesnelere başvuran (referans) tipler olarak adlandırılır.

Primitive ve non-primitive veri tipleri arasındaki temel farklar şunlardır:

Primitive tipler Java'da önceden tanımlanmıştır. Non-primitive tipler ise programcı tarafından oluşturulur ve Java tarafından tanımlanmaz (String hariç).

Non-primitive tipler, belirli işlemleri gerçekleştirmek için yöntemleri çağırmak için kullanılabilirken, primitive tipler kullanılamaz.

Primitive tiplerin her zaman bir değeri vardır, non-primitive tipler ise null olabilir.

Primitive tipler küçük harfle başlar, non-primitive tipler ise büyük harfle başlar.

Bir primitive tipin boyutu veri tipine bağlıdır, non-primitive tiplerin ise tümü aynı boyuttadır.

Non-primitive tiplere örnek olarak String'ler, diziler (Arrays), sınıflar (Classes), arayüzler (Interfaces) vb. verilebilir.