**ARRAYS NOTES**

**Create an Array in Java:**

When we create an array, we:

**Declaration:** Bildirim. Dizinin tipini ve ismini belirttigimiz ilk adimdir.

int[] numbers;

**Instantiation**: Ornekleme. new anahtar kelimesi kullanilarak dizinin bellekte gercekten yer ayrildigi adimidir. Dizinin boyutu burada belirlenir.

numbers = new int[5];

**Initialization**: Baslatma. Dizinin elemanlarina ilk degerlerin atandigi adimdir. Elemanlara deger atamasi yapilir.

numbers[0] = 25;

Tek satirda bu 3 adim:

int []numbers = new int[]{1,2,3,4};

——————————-

int[] array = new int[3];

Heap’te alan oluşturulur: **JVM**, 3 elemanlık bir int dizisi için heap’te ardışık bir bellek bloğu ayırır.

Dizinin elemanları henüz bizim tarafımızdan belirlenmemiş olsa da, Java otomatik olarak her elemanı 0

ile başlatır. Yani array[0], array[1] ve array[2] başlangıçta 0 olur.

Stack’te referans tutulur: array değişkeni stack üzerinde tek bir referans olarak durur. Bu referans,

heap’te oluşturulan dizinin başlangıç adresine işaret eder. Stack’te dizinin elemanları için ayrı alan

açılmaz; sadece heap’teki diziyi gösteren bir işaretçi vardır.

Index mantığı: Dizinin elemanlarına array[0], array[1], array[2] şeklinde 0’dan başlayan index ile

erişilir. Bunun nedeni, ilk elemanın heap’teki başlangıç adresine direkt karşılık gelmesi ve böylece

hesaplamaların basit ve hızlı olmasıdır.

——————————————

**1. İlkel (Primitive) Veri Tipleri:** İlkel veri tipleri en temel yapı taşlarıdır. Değerleri doğrudan **stack** bellekte saklanır.

int, double, float, char, boolean, byte, short, long vb ilker veri tipleridir.

**2. Referans (Reference) Veri Tipleri:** Bu veri tipleri ise daha karmaşık verileri tutar. **Heap** bellekte saklanırlar. Değişkenin kendisi aslında bu verinin bellekteki adresini (referansını) tutar.

* Sınıflar(Classes): Bir referans veri tipidir. String’de bir sınıf olduğu için ilkel bir veri tipi değildir. new anahtar kelimesi ile oluşturulur, heap bellekte bir yer ayırılır, stack’te ise bu yere işaretçi (pointer) ile işaretlenir.
* Diziler (Arrays): Aynı türden birden fazla veriyi sıralı bir şekilde saklamak için kullanılır. İklel veya referans veri tiplerinden oluşabilir. Dizi ilk oluştuğunda heap’de dizinin elemanları sıralı bir şekilde tutulur. Stack’te ise heap’taki en baştaki elemanın pointer’ı tutulur. Bu şekilde dizilere çok hızlı bir şekilde erişmemiz sağlanır. Mesela dizi[5]’e gitmek için basit bir matematiksel boyut yeterlidir.

= baslangic adresi + (index x veri\_tipi\_boyutu) = int dizi[5 ]->100+ 5x4

* Arayüzler(Interfaces) ve Koleksiyonlar(Collections) da referans veri tipleridir.

Time and space complexity of 1D arrays:

