

3- JVM( Java Virtual Machine) sayesinde platform bağımsız bir çalışma ortamı sağlar. JVM Java Development Kit (JDK) içerisinde yer alır. Çalışma mantığı çok basittir. Yazdığımız .java uzantılı kodlarımızı sanal makine içerisinde çalıştırır ve derler. Bu derleme sonucunda .byte uzantılı yeni bir dosya oluşur.Bu dosya uzantısı geliştirme ortamının işletim sisteminden bağımsız bir şekilde ortak olarak her türlü çalışma ortamında çalışabilir.

4-Heap ve Stack arasındaki en belirgin farklardan ilki birinin dinamik verileri saklaması, diğerinin ise statik verileri saklamasıdır. Dinamik veri hafıza içerisinde bulunan değişkenlerimizin sabit olması gerekmez ancak statik veriler program çalıştıktan sonra değiştirilemez.

Heap dinamik verileri saklar, Stack ise statik verileri saklar.

Stack ve Heap arasındaki diğer bir fark ise Stack içerisindeki veriler işlem bittikten sonra otomatik temizlenir ancak Heap içerisindeki verileri hafıza temizlik yöntemlerinden biri kullanılarak manuel temizlenmesi gerek.

5-String Sınıfının immutable olmasını sağlayan bir kaç tane farklı sebep var. String sınıfının içerisindeki değişkenlerin access modifierları Private olarak tanımlanır. Ayrıca Set metodları bulunmaz bu şekilde değişkenlere Get metodları ile erişilebilir ancak değiştirilemez.

Set metodunun olmaması dışında değişkenleri tanımlarken final olarak tanımlandığı için veri bütünlüğü sağlanabiliyor. Bu değişkenlerin ilk atamaları yapılırken ise constructor ile yapılır örneğin "String str = new String("KODLUYORUZ");".

Bunlardan emin olabilmek için String Sınıfının kaynak kodlarına girdiğimizde değişkenler ile ilgili tüm methodların erişim belirleyicileri private olarak belirlenmiştir.

6-Java'da çoklu kalıtım normal sınıflar aracılığı ile yapılamaz. Bunun sebebi karmaşaya engel olabilmektir. Normal sınıflarda bulunan methodlar soyut olmadığı için method karışıklığına sebep olur. Ancak çoklu kalıtım kullanılması gerektiği durumda interface kullanılarak yapılabilir. Interface kullanılarak yapılabilmesinin sebebi interface'lerin içerisinde bulunan tüm methodlar soyuttur.

7-Build Tool, bir projenin birçok farklı konuda yapılması gereken işlerin kolaylaştırılması ve düzenli hale gelmesi için kullanılan araçlara verilen addır. Farklı farklı ihtiyaçlar sebebiyle java ekosisteminde birçok build tool bulunmaktadır. Bunlardan birkaçı: Apache Maven, Apache Ant, Gradle, CMake, Bazel, JFrog gibi bir çok araç bulunmaktadır.

8- Collection Framework temel olarak 3 bölümden oluşur. List, Queue ve Set. Bunların dışında Map adında ayrı bir bölüm daha vardır.

-Queue, kelime anlamı olarak kuyruk anlamına gelir. Basit olarak gerçek hayatta bir yerde sıra beklerken oluşan kuyruk ile aynı mantığa sahiptir. Sıra ilk giren, sıradan ilk çıkandır. Bu duruma FIFO(First in First out) denir.

-List, normal bir liste gibi düşünebiliriz. Normal bir liste hazırlarken sırayla listede bulunmasını istediğimiz şeyleri listeye ekleriz ve listede aynı elemandan 1 adetten fazla bulunabilir.

-Set, çalışma mantığı List gibidir ancak aralarındaki en büyük fark, aynı elemandan sadece 1 adet bulundurulabilir.

-Map, key ve value eşleştirme üzerine kurulu bir interface'dir. Davranışsal olarak Set ile benzer bir şekildedir. Aynı değerden 1 adetten fazla bulunamaz.

List Interface'inin içerisinde bulunan sınıflar

-Vector, ArrayList, LinkedList, Stack

Set Interface'inin içerisinde bulunan sınıflar

-HashSet, TreeSet

Queue Interface'inin içerisinde bulunan sınıflar

-PriorityQueue, LinkedList

Map Interface'inin içerisinde bulunan sınıflar

-HashMap, Hashtable, LinkedHashMap, TreeMap

Collection Framework içerisinde bulunan önemli metotlar

<b>Metot</b>	<b>İşlevi</b>
add()	Oluşturduğumuz Collection'a nesne ekler
isEmpty()	Collection'ın boş olup olmadığını kontrol eder. Boolean olarak dönüş sağlar.
clear()	Collection içerisinde bulunan tüm nesneleri temizler.
remove()	Seçtiğimiz nesneyi siler.
size()	Collection içerisindeki elemanların sayısını döndürür.
stream()	Collection içerisindeki elemanları sıralı olarak döndürür.
max()	Collection içerisinde bulunan en yüksek değeri döndürür.
contains()	Collection içerisinde sorgulanan eleman bulunuyorsa true döndürür.
equals()	Collection içerisinde bulunan elemanlarla nesneyi karşılaştırır.