Ismet Mertkan YILMAZ

Payten Java Spring Bootcamp HW1.1

1-Pass by value, pass by reference kavramları nedir ? Java' ile ilişkili olarak açıklayınız.

Pass-by-reference olan dillerde değişkenler direkt olarak memorydeki adresi gösterirler ve üzerlerinde yapılan işlemler direkt olarak bu memory hanesinde yapılır.
Pass-by-value olan dillerde ise değişkenler direkt olarak bir memory hanesi üzerinde işlem yapmazlar bunun yerine aslında bir adres tutarlar ve bu adres memory hanesinin adresidir. O sebeple o adreste bulunan değer aslında değişkenin değeri gibi print ettiğimizde vesaire bize gelir. Aslında değişken adresi tutuyordur ve print edildiğinde bu adrese gidilip bu hanedeki değer print ettiriliyordur. Pass-by-referencede ise direkt olarak hanedeki değer print edilir.
Java ise pass-by-value bir dildir ve değişkenler aslında bir adres tutarlar.

2-Immutability nedir, neden önemlidir ? Bir Java sınıfı nasıl immutable yapılır ?

Immutability yani değiştirilemezlik bir yapının adından da anlaşılacağı üzere değiştirilememsidir. String, Integer, Boolean yapıları aslında immutabledır. Akla şu soru gelebilir, değişken ismi = ... yaparak biz aslında değiştirebiliriz. Fakat burada java aslında memorydeki değeri değiştirmez, değişkenimizin bir önceki soruda da gösterdiği adresi değiştirir ve hafızada yeni bir haneye işaret etmiş olur. Ama değerin değişmesinden önceki hafıza hanesinde eski değer korunur. Faydası nedir?
1) Java direkt olarak hafızadaki değer yerine, değişkenin tuttuğu adresten hafıza gittiği için aynı değeri tutan başka değişkenler de bu memory hanesini gösteriyor olabilir. Ve immutable sayesinde eğer 1.değişkenin değeri değişse bile yeni bir hafıza hanesini göstereceği için eski değişkenin değeri 2. değişken için korunur. Çünkü hafızadaki değer değişmedi.
2) Bu sayede bir işlem eğer multithread bir işlemse ve bu değişkenin değeri bu threadler işlem yaparken arada değişirse bir güvenlik açığı veya yanlış bir işlem yapılmasına sebep olabilir. Bu durumları immutablity ile engelleriz.
Class nasıl immutable edilirin cevabı ise 5 maddelik bir reçetedir. 1)Değişkenlerin hepsi private yapılmalıdır. 2)Setter bulundurmamalı. 3)Değişkenler final yapılmalıdır. 4)Değişkenlerin ilk değerleri constructor ile verilmelidir. 5)Sınıf final yapılmalıdır.

3-Framework ve library arasındaki fark nedir ?

Libraryler bir işi yapmayı kolaylaştıran, basitleştiren ve temiz kod yazmamızı sağlayan kodlardır. Libraryleri kodumuza dahil ederek gerektiğinde bunları direkt olarak kodumuzda biz çağırarak bir işlem yaptırırız.
Frameworklerde ise olay akışı framework tarafından kontrol edilir. Libraryde olduğu gibi biz çağırmayız, framework bu olay örgüsü içinde kodun akışına göre bizim yazdığımız kodu çağıracaktır.

4- Java'da Garbage Collector' un görevi nedir?

Garbage collector JVM'in aslında bir nevi kullanılmayan memory alanı temizlikçisidir. Programda objeler yaratıldıklarından sonra kullanılmaya devam edenleri ve kullanılmayanlar olarak ikiye ayırırsak, kullanılmayanların memoryden silinmesi, hafıza yönetimi açısından gereksinimdir. Bu

ihtiyacı javada Garbage collector giderir ve bu programda belli bir süre sonra kullanılmayan hafızada oluşturulan alanları temizler. Javada bu otomatik olarak JVM tarafından üstlenilir ve programcıya bırakılmazken, bazı dillerde de bu programcı tarafından yapılmalıdır. Garbage collection işlemi belli aralıklarla gerçekleştirilir ve memory birkaç farklı alana bölünmüştür. Bazı nesneler kullanımda olma süresine göre hafızada daha yaşlı olarak adlandırılan bölümlere taşınabilir veya kullanılmama durumuna göre temizlenirler.

5- Memory leak nedir ? Java'da memory leak oluşması mümkün müdür ?

Memory Leak denen olay heap'te temizlenmesi gereken objelerin temizlenmemesi sonucu oluşan hafıza dolmasıdır. Garbage collector sorusunda anlatıldığı üzere JVM hala kullanılmayan bir objeyi daha doğrusu referans edilmeyen objeleri memory'den boşaltır. Bu temizleme hafızada yer açılmasına ve gereksiz objelerin temizlenmesini sağlar. Bu temizlenmenin olmadığı duruma "memory leak" denir. Bu olay gerçekleştiğinde Java OutOfMemory hatası verir.
Javada memory leak olabilir. Buna fazla static değişken kullanımı, collection, stream yapısının kapatılmaması, uygun hashCode & equals yazılmaması bazı başlıca nedenlerdir.

6-Yeni Java sürümleri ne sıklıkla çıkmaktadır ?

Oracle her 6 ayda bir yeni bir java sürümü çıkarmaktadır. Bu 6 ayda 1 den kastedilen yeni sürümler non-LTS yani "non-Lon term support" sürümlerdir. Bunlar ufak değişiklikler içeren sürümlerdir. LTS olan sürümleri ise 3 senede bir çıkaracağını açıklamıştır.

7-Stack ve Heap nedir ? Java'da hangi yapılar stack ile, hangi yapılar heap ile ilişkilidir ?

Stack yapısı memoryde primitive değişkenleri ve obje referanslarını barındıran sabit alana sahip hafıza alanıdır. Stack işleyişinde "son giren ilk çıkar" LIFO sistemi vardır. Zaten türkçe adından da anlaşılacağı üzere yığına benzer bir işleyişi vardır. Stackler run-time da hafızanın belli bir alanına belli bir boyutta atanan yapılardır. Stack yapısı bir thread tarafından oluşturulur ve ömrü ona bağlıdır.

Heap ise objeleri barındırıdan hafıza alanlarına denir. Heap yapısı stackte olduğu gibi sıralı bir hafıza alanı yerleşimi ile veya "son giren ilk çıkar" mantığı yerine daha serbest biçimde hareket eder. Heapte yer alan objeler Garbage Collector tarafından işleri bitince temizlenir, stackte yer alanlar ise method return edildiğinde temizlenir. Heap yapısı program koşmaya başladığı anda oluşturulur ve bütün threadler tarafından ortak kullanılabilir. Ömrü programa bağlıdır.

8-OpenJDK ve OracleJDK arasındaki farklar nelerdir ?

OpenJDK ve OracleJDK arasında teknik olarak çok büyük bir fark görülmemektedir. Fakat tıpatıp anı oldukları söylenemez. Farklar şu şekilde sıralanabilir; 1)OracleJDK 3 senede bir yeni versiyon sunarken OpenJDK'da bu süre 6 ayda birdir. 2)OpenJDK ile OracleJDK lisanları farklıdır. 3)Performans olarak arada çok ciddi bir fark bulunmasa da OracleJDK daha stabildir. 4)OracleJDK farklı versiyonları için destek sunarken, OpenJDK'da bu son versiyon ile sınırlıdır.

9-@FunctionalInterface anotasyonu nerelerde kullanılabilir, neleri sağlar ?

Functional Interfaceler tek bir tane abstract method içeren interfacelerdir. İstedikleri kadar default method, static method olabilir ama sadece bir adet abstract method olmalıdır. Functional interfaceler sayesinde lambda expressionlar kullanılabilir.

Bir functional interface tanımlarken @FunctionalInterface kullanmak zorunlu değildir fakat bunu kullanamak, interface'in functional interface olduğu tanımlamak için ve içine bir tane abstract method yazdığımızı doğruladığı için yararlıdır. Eğer @FunctionalInterface anotasyonu kullanılır ve içine birden fazla abstract method yazılırsa compile edilirken error alırız.

10-Java'da hangi functional interface'ler yer almaktadır ? Yaptığınız araştırmada en popüler/göze çarpanlar hangileridir ?

Bazı kullanılan built-in func. Interfaceler şunlardır; Predicate, BinaryOperator, Function, Consumer…

Predicate ile yapılan işlem sonucu, kullanılan lambde expression sonucu boolen bir değer döndürülür. BinaryOperator ile iki adet aynı tipte input alıp sonuç olarak aynı tipte bir adet sonuç alabiliriz. Consumer ile aslında void bir değer döner, bu genelde birşeyleri print out etmek için kullanılır.