

1. Java'nın platform bağımsızlığını nasıl sağladığını anlatınız.(5 PUAN)

Cevap:

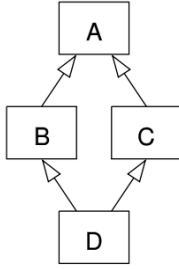
Java, JVM yani Java Virtual Machine üzerinde çalışır. Bu nedenle işlemciden bağımsız yazılım geliştirilmesine olanak sağlar. Javac compiler'ı, Java programını alıp bunu makine koduna çevirir böylece bilgisayar programı anlayabilir. Compiler programı derlediğinde her bilgisayar için farklı makine kodu yaratmamış olur ve tek bir makine kodu elde edilir. JVM'nin kullandığı makine dili byte kodu'dur. JVM bu tek tip makine kodunu kullanarak çıktı üretir ve Java'yı platform bağımlılığından kurtarır.

2. Java neden çoklu kalıtımı desteklemez? Hangi diller bu duruma neden izin verir? (5 PUAN)

Cevap:

Çoklu Kalıtım: Bir sınıfın veya bir nesnenin, birden fazla üst sınıf veya nesnenin özelliklerini miras almasıdır.

Çoklu kalıtım karmaşık ve büyük projelerde kullanıldığında karmaşıklığa yol açabilir. Anlaşılması zor kalıtım ağaçları ortaya çıkabilir. Single Responsibility (Projedeki her sınıfın sadece tek bir sorumluluğu olması gerektiği.) prensibi ihlal edilebilir. Bir sınıfa birden fazla sınıfın davranışlarını eklemek bu kuralın ihlaline yol açabilir. Çoklu kalıtım kullanılması sonucu Diamond Problem de görülebilir.



Burada D class'ı B ve C'nin child class'ı konumundadır. Miras yolu ile B ve C'de bulunan aynı imzaya sahip metodlardan hangisini alacağını bilemez. Bu probleme Diamond Problem denir.

Karmaşıklığı ve belirsizliği azaltmak adına ve dili basitleştirmek adına, Java'da çoklu kalıtım desteklenmez.

Çoklu kalıtımı destekleyen programlama dillerine örnek olarak C ++, Python, Perl, Eiffel, Dylan, Curl, Eulisp ve Tcl verilebilir. Çoklu kalıtım kullanımına izin vermelerinin nedeni karmaşık ilişkilerin modellenmesine olanak sağlaması olabilir.

3. Build Tool nedir? Java ekosistemindeki toollar neler?

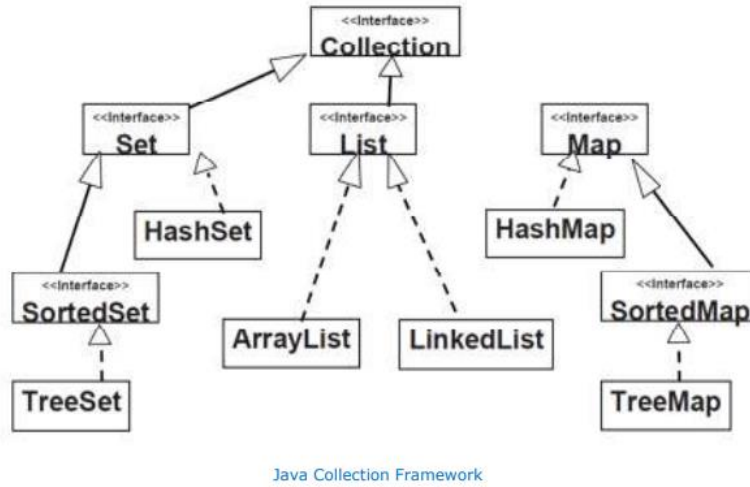
Cevap:

Bir proje oluşturmak, derlendikten sonra gerekli kütüphaneleri ve gereksinimleri sisteme entegre etmek için kullanılan araçlardır. Build Tool'lar Dependency'leri yönetmemizi sağlar.

Java Ekosistemindeki Tool'lardan Bazıları: Jenkins, Apache Ant, Gradle, TeamCity, Maven, CMake, Bamboo...

4. Collection framework içerisindeki yapıları örnekleyip açıklayınız. (20 PUAN)

Cevap:



Set: Kümelerdir. Küme kavramında olduğu gibi unique değerler barındırır ve veri tekrarına izin vermez. Index ile verilere erişilemez.

HashSet: Verileri tablo üzerinde hashCode'lar ile unique bir şekilde depolar. (hashCode'a nesnelerin ya da değişkenlerin DNA'sıdır benzetmesi yapılmıştır.)

SortedSet: Set ara yüzünün bir alt ara yüzüdür. Set ara yüzünün metotlarına ek olarak bazı metotlar getirir.

TreeSet: Unique değerleri hiyerarşik bir sıralama ile saklar.

List: Elemanları sıralı bir şekilde tutar. Elemanları tekrarlanabilir. Index kullanılarak elemanlarına ulaşılabilir.

ArrayList: Elemanları depolamaya yarayan dinamik bir array'dir. Sıralıdır. Elemanların eklenme sırası korunur. Index kullanılarak elemanlara ulaşılabilir.

LinkedList: Veriler node adı verilen yapılar üzerinde tutulur. LinkedList'in başından, ortasında bir yerden ya da sonundan node silinebilir ya da eklenebilir. Bu LinkedList'i bozmaz. Index kullanılarak elemanlara ulaşılabilir.

Map: Key ve value değerleri vardır. Her bir değer bir anahtar ile eşleşir. Anahtar değerleri unique değerlerdir. Value'lar unique olmayabilir.

HashMap: HashMap'de yer alan değerler için (value) unique anahtarlar (key) atanır. Anahtarlar aracılığı ile value'lara ulaşılabilir.

SortedMap: Map ara yüzünün bir alt arayüzüdür. Map ara yüzünün metotlarına ek olarak bazı metotlar getirir.

TreeMap: Key ve value değerleri vardır. Her bir değer bir anahtar ile eşleşir. İçerisinde bulunan değerleri default olarak küçükten büyüğe sıralar. Bu nedenle sıralama algoritmalarında tavsiye ediliyor. İçerisindeki verilerin tekrarlanmasına izin vermez.

Set örneği:

```
HashSet<String> animals = new HashSet<String>();
animals.add("Elephant");
animals.add("Tiger");
animals.add("Lion");
```

Bu örnekte bir HashSet oluşturuldu ve içerisine Elephant, Tiger ve Lion isimleri eklendi. Bu elemanları unique yapmak için arkaplanda hashCode'lar ile eşleştiriliyor. hashCode'lar x,y,z olsun dersek;

Elephant <-----> x

Tiger <-----> y

Lion <-----> z

List örneği:

```
List<String> fruits = new ArrayList<String>();
fruits.add("Mango");
fruits.add("Apple");
fruits.add("Banana");
fruits.add("Grapes");
```

Bu örnekte bir ArrayList oluşturuldu ve içerisine meyve isimleri sırası ile eklendi. Index'leri ile birlikte liste şu şekilde görünür:

Index = 0 -> "Mango"

Index = 1 -> "Apple"

Index = 2 -> "Banana"

Index = 3 -> "Grapes"

Daha sonra indexleri kullanarak meyvelere ulaşmak mümkündür. Mesela fruits.get(0) ile Mango'yu çekebiliriz.

Map örneği:

```
Map<String, Integer> vehicles = new HashMap<>();
vehicles.put("BMW", 5);
vehicles.put("Mercedes", 3);
vehicles.put("Audi", 4);
vehicles.put("Ford", 10);
```

Bu örnekte vehicles adlı bir hashMap oluşturuldu. Daha sonra içerisine key ve value değerleri eklendi.

<u>Key</u>	<u>Value</u>
BMW	5
Mercedes	3
Audi	4
Ford	10

Burada key değerleri araç markaları, value değerleri ise galeride kaç adet olduğunu gösteriyor olsun.
vehicles.get("BMW") ile 5 değerine ulaşılabilir.