

1. String class'ı nasıl immutable olmayı sağlamaktadır örnek ve çizimlerle açıklayınız.(5 PUAN)

Immutable(Değişmez) sınıflar en bilinen şekliyle nesnesi oluşturulduktan sonra nesne içeriği değiştirilemeyen sınıfları ifade etmektedir.

İlkel bir tip olan integer üzerinden durum değerlendirmesi yapalım,

int sayi=1; (integer verip tipindeki sayi isimli değişkene 1 değerini atadık.)

sayi=2; (sayi değişkeninin 1 olan değerini 2 olarak değiştirdik.)

integer veri tipi mutable bir veri tip olduğu için bu değişikli yapmamıza izin verir ancak immutable olan String veri tipi/sınıfında durum biraz daha farklıdır.

String text="word"; (String veri tipindeki text değişkenine "word" değerini atadık.)

text="sentence"; (text değişkeninin "word" değerini "sentence" olarak güncellemeye çalışır ancak böyle bir değişiklik gerçekleşmez.)

String text="word";

Bilindiği üzere String'ler referans tipli değişkenler olduklarından değer atamaları referanslar üzerinden gerçekleşir. text değişkenine "word" değerini atadığımızda text belleğin stack kısmında referans tutucu olarak tanımlanırken "word" belleğin heap kısmında referans olarak tanımlanır. Yani text referans tutucusu "word" referansını tutar.

text="sentence";

Yukarıdaki şekilde text referans tutucusunun tutmuş olduğu referansı güncellemeye çalıştığımızda heap bellekte "sentence" değerine sahip yeni bir referans oluşturulur ve text referans tutucusunun bu referansı tutması sağlanır. Ancak "word" referansı bellekte tutulmaya devam eder. Yani kısaca referans'ın değeri değiştirilmez yeni bir referans yaratılarak değer bu referansa atanır. (Referans tutucusu olmayan referanslar bir süre sonra bellek sağlığı için Java Çöp Toplayıcısı tarafından bellekten otomatik olarak silinir.)

Peki bu yapı nasıl sağlanır?

Öncelikle String sınıfı final olarak tanımlanmıştır. Böylelikle herhangi bir başka sınıf tarafından bu sınıftan kalıtım alınamaz.

```
public final class String
    implements java.io.Serializable, Comparable<String>, CharSequence,
        Constable, ConstantDesc {
```

Value field'ı da private ve final tanımlanmıştır. Set methodu da bulunmamaktadır. Böylelikle sınıf objesinin bu değişkenine dışarıdan get methodu yoluyla atama yapılamaz. Private olduğu için dışarıdan erişilemez. Final olduğu için sadece sınıf yapıcı methodu kullanılarak bir kereye mahsus atama yapılabilir.

```
@Stable  
private final byte[] value;
```

```
public String() {  
    this.value = "".value;  
    this.coder = "".coder;  
}
```

Böylece bu sınıftan oluşturmuş olduğumuz nesne'nin değeri oluşturma sırasında verilen değerden başka değere değiştirilemez. Bu şekilde String class'ı immutable olmayı sağlar.