

İç İçe Sınıflar (Nested Classes)

Sınıfları ve arabirimleri daha okunabilir ve bakım yapılabilir şekilde tek bir yerde mantıksal olarak gruplandırmak için iç sınıfları kullanırız.

Ek olarak, özel veri üyeleri ve yöntemleri dâhil olmak üzere dış sınıfın tüm üyelerine erişebilir.

```
1. class Java_Outer_class{
2.     //code
3.     class Java_Inner_class{
4.         //code
5.     }
6. }
```

İç sınıfların üç avantajı vardır:

1. İç sınıflar, özel sınıf da dâhil olmak üzere dış sınıfın tüm üyelerine (veri üyeleri ve yöntemleri) erişebilen belirli bir ilişki türünü temsil eder.
2. İç sınıflar, mantıksal olarak sınıfları ve arabirimleri yalnızca tek bir yerde gruplandığı için daha okunabilir ve korunabilir kod geliştirmek için kullanılır.
3. Kod Optimizasyonu: Yazmak için daha az kod gerektirir.

Nested Class: İç içe sınıflar static yapıdadır.

Inner Class: İç sınıflar static olmayan yapıdadır.

Enum Kavramı:

```
public enum Tip {Dogu, Bati, Kuzey, Guney}
```

Dosyaya Yazma İşlemi:

```
FileWriter fw = new FileWriter("file.txt");
fw.write(string data);
fw.close();
```

Serialization ve **Deserialization** Kavramı: Serileştirme, bir nesnenin durumunu bir bayt akışına yazma mekanizmasıdır. Esas olarak Hibernate, RMI, JPA, EJB ve JMS teknolojilerinde kullanılır. Serileştirmenin ters işlemi, bayt akışının bir nesneye dönüştürüldüğü seri kaldırma (**Deserialization**) olarak adlandırılır. Esas olarak ağdaki nesnenin durumunu dolaşmak için kullanılır.

```
import java.io.Serializable;
```

```
FileOutputStream fos = new FileOutputStream( "fileName");
ObjectOutputStream out = new ObjectOutputStream(fos);
out.writeObject(list);
```

```
ObjectInputStream in = new ObjectInputStream(new FileInputStream("fileName"));
```