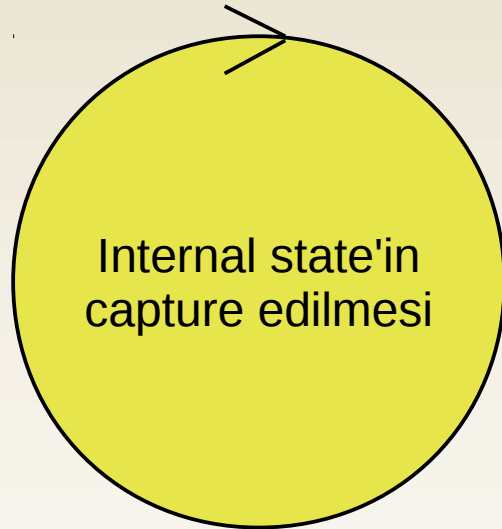
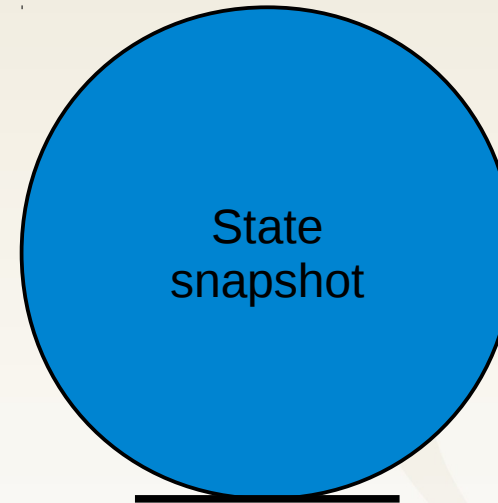


# GOF Memento Pattern

# Memento

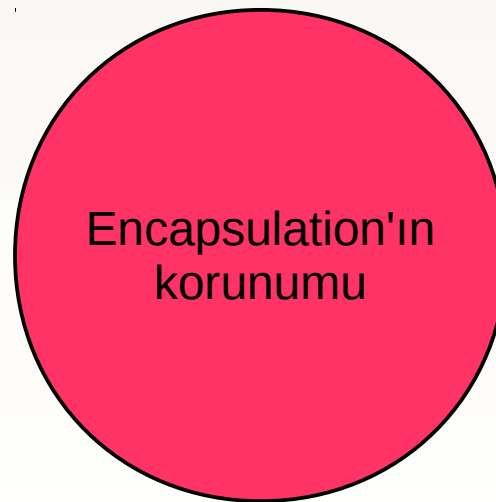


Belirli durumlarda **nesnelerin dahili state bilgisinin dışarıda bir ortamda saklanması** gerekebilir



Ancak encapsulation prensibi nedeni ile nesneler kendi **dahili state bilgilerinin dış dünya tarafından erişilmesine** izin vermeyebilir

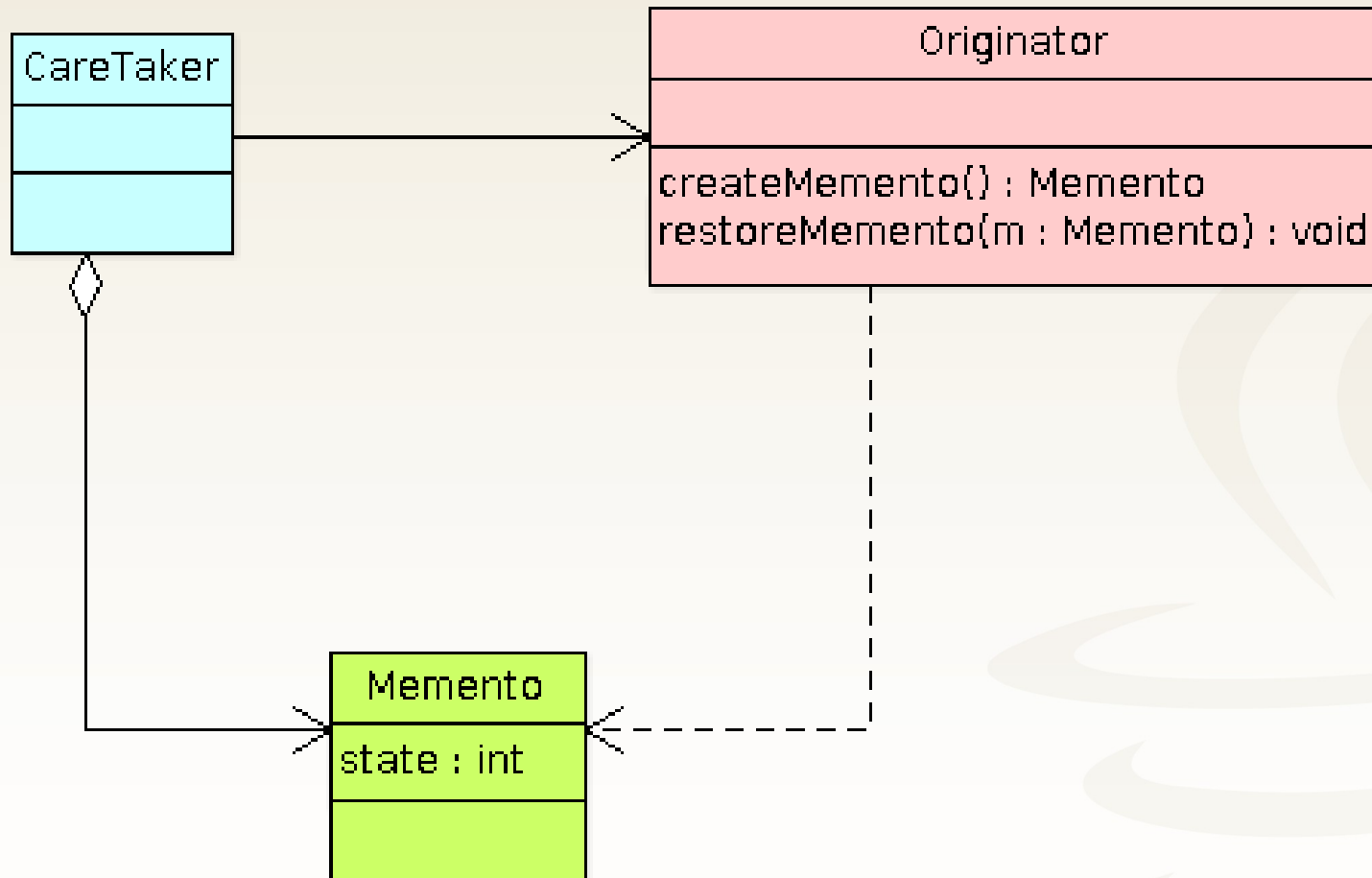
Çünkü bütün state bilgisinin dışarıya erişilebilir kılınması **encapsulation prensibi** ile uyuşmaz



Memento, nesnenin encapsulation'ını bozmadan **dahili state bilgisininin snapshot'unun** saklanmasını sağlar

Daha sonra ise bu snapshot kullanılarak nesne **o anki state'e geri** döndürülebilir

# Memento Sınıf Diagramı



# Java'da Memento

- Java'da bir **nesnenin state'ini binary biçime** dönüştürerek bir dosya sistemine yazmak veya networkte transfer etmek mümkündür
- Daha sonra bu **binary veri tekrar nesne gösterimine** dönüştürülebilir
- Bu işleme **serialization/deserialization** adı verilmektedir
- Bu işleme tabi tutulacak nesnelerin sınıfları **Serializable** veya **Externalizable** arayüzlerinden birisini implement ederler

# Java Object Serialization

- Serialization'a tabi tutulacak sınıfların **java.io.Serializable** arayüzünü implement etmeleri gerekir
- Aksi takdirde **NotSerializableException** hatası fırlatılacaktır
- Bu arayüz marker interface'tir. Herhangi bir metot barındırmaz
- Sadece ilgili sınıfın state'inin binary stream verisine dönüştürülebileceğini belirtir

# Java Object Serialization

- Serializable bir sınıfın **bağımlı olduğu diğer bütün sınıflar** da Serializable olmak zorundadır
- Herhangi bir field serialization işlemi dışında bırakılmak istenirse **transient** modifier'ı kullanılabilir
- Serializable sınıflar serialization/deserialization işlemini kendilerine göre **özelleştirebilirler**

# Java Object Serialization

```
private void writeObject(java.io.ObjectOutputStream out) throws
IOException {

}

private void readObject(java.io.ObjectInputStream in) throws
IOException,
ClassNotFoundException {

}
```

- Serializable sınıf içerisinde yukarıdaki gibi **writeObject** ve **readObject** metotları implement edilerek işlem özelleştirilebilir
- Diğer bir yöntem ise **java.io.Externalizable** arayüzünü implement etmektir

# Java Object Serialization

- Serialization işlemi çalışma zamanında her Serializable sınıf için bir **versiyon numarası** belirler
- Serialization ve deserialization işlemlerini yapan tarafların **aynı versiyon numarasına** sahip sınıflar ile bu işlemi yapmaları **önemlidir**
- Aksi takdirde deserialization sırasında **InvalidClassException** hatası fırlatılır



# Java Object Serialization

- Sınıflar bu versiyon numarasını kendileri yönetebilirler
- Bunun için sınıf içerisinde  

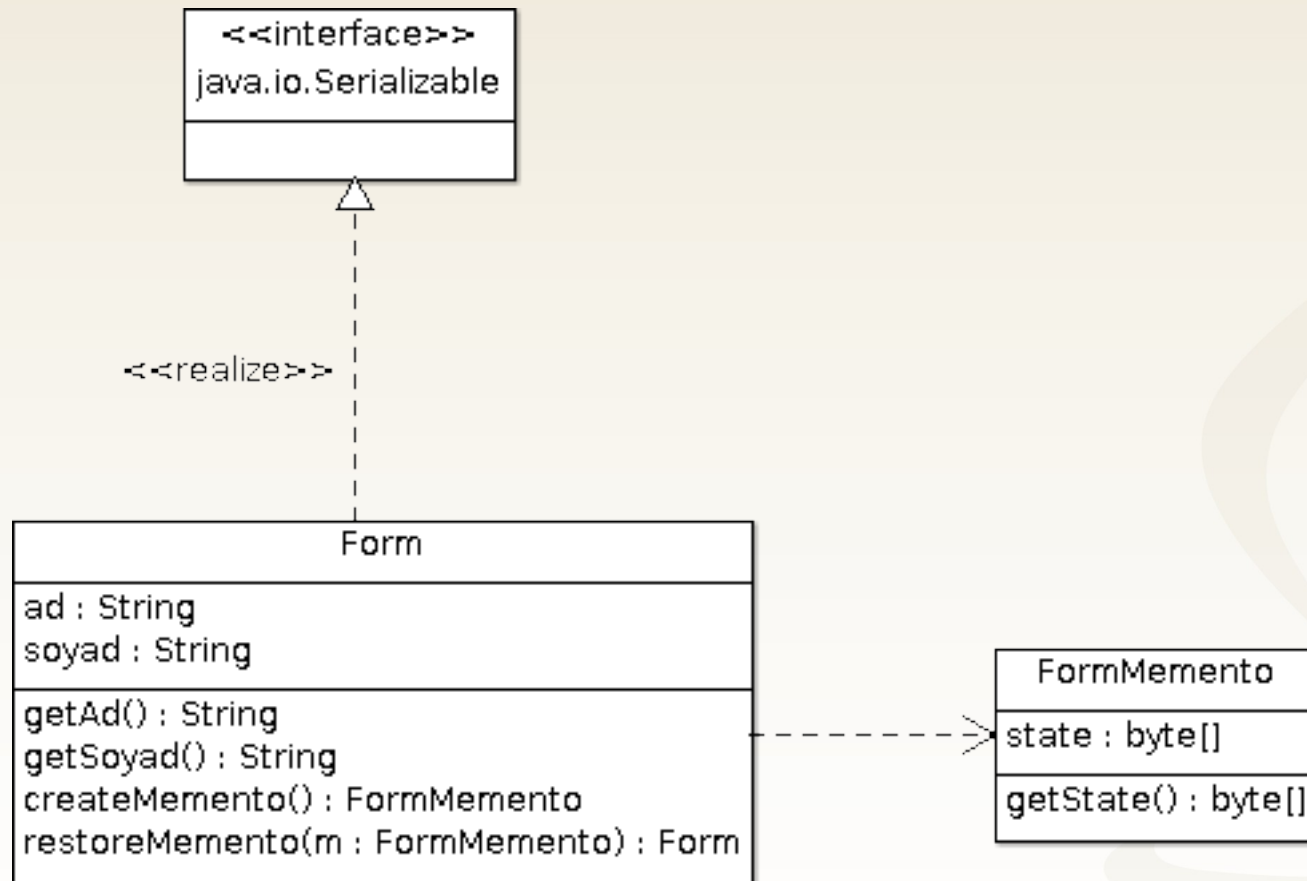
```
private|protected|public static final long serialVersionUID = 1L;
```

 şeklinde bir field tanımlamaları yeterlidir
- Sınıf içerisinde explicit biçimde **serialVersionUID** tanımlanması ve yönetilmesi önerilen yaklaşımdır
- Java serialization sürecinin bu değeri kendisinin üretmesi sınıfın detaylarına bağımlı olduğu için **hayli kırılgandır**

# LAB ÇALIŞMASI: Memento

- İçerisinde ad ve soyad bilgilerini barındıran bir Form sınıfı vardır
- Ad ve soyad bilgisi Form nesnesi yaratıldığı vakit set edilebilmekte, daha sonra değiştirilememektedir. Sadece getter metotlar mevcuttur
- Form sınıfının property'lerinin encapsulation'ına zarar vermeden state'inin dışarıya aktarılması ve bu state kullanılarak daha sonraki bir zaman da nesnenin yeniden oluşturulmasını sağlayınız

# LAB ÇALIŞMASI: Memento



# Memento Örüntüsünün Sonuçları

- Encapsulation bozulmadan **state'in dışarı aktarılması** sağlanır
- Bu sayede nesne daha sonraki bir zamanda tekrar **aynı state bilgisi** ile oluşturulabilir

# İletişim



[www.harezmi.com.tr](http://www.harezmi.com.tr)

[www.java-egitimleri.com](http://www.java-egitimleri.com)



[info@harezmi.com.tr](mailto:info@harezmi.com.tr)

[info@java-egitimleri.com](mailto:info@java-egitimleri.com)



[@HarezmiBilisim](https://twitter.com/HarezmiBilisim)

[@JavaEgitimleri](https://twitter.com/JavaEgitimleri)