

Tasarım Kalıpları (Design Patterns)

7. Bölüm: Diğer Kalıplar



Eğitmen:

Akın Kaldıroğlu

Çevik Yazılım Geliştirme ve Java Uzmanı

Konular



- Giriş
- Null Object

Giriş

Davranışsal Kalıplar



- Bu bölümde GoF'da yer almayan bazı yaygın kalıplar ele alınacaktır.

Null Object

Null Object Kalıbı



- **Intent:** Provide an object as a surrogate for the lack of an object of a given type. The Null Object provides intelligent “do nothing” behavior, hiding the details from its collaborators.
- **Amaç:** Verilen tipin nesnesi olmaması durumu için vekil olarak bir nesne sağla. Null Object, detayları diğer nesnelerden saklayarak akıllı bir “hiç bir şey yapma” davranışı sağlar.
- İlk defa Bobby Woolf tarafından **Null Object** başlıklı makale ile **Pattern Languages of Program Design 3**'te teklif edilmiştir.

Null Object Kalibi



- <https://www.cs.oberlin.edu/~jwalker/nullObjPattern/>
- <https://refactoring.guru/introduce-null-object>
- https://en.wikipedia.org/wiki/Null_object_pattern

Problem - I



- Bir referansın hiç bir nesneyi göstermediği durumlarla sıklıkla karşılaşılır.
- Tony Hoare 1965 yılında şöyle demişti:

I call it my billion-dollar mistake. It was the invention of the null reference in 1965. I couldn't resist the temptation to put in a null reference, simply because it was so easy to implement.

- Bu durumda `null` olan referansa erişmek sıkıntılı olduğundan, `NullPointerException` vb. sıra dışı durum fırlatılmasını önlemek için `null` kontrolü yapılır.

Problem - II



- Aslolan hiç bir zaman null referens geçmemek ve döndürmemektir.
- Ama hem kontrol edemediğimiz third party vb. APIlerde hem de geliştirdiğimiz kodlarda code smell olarak null referanslar yaygındır.

Problem - III

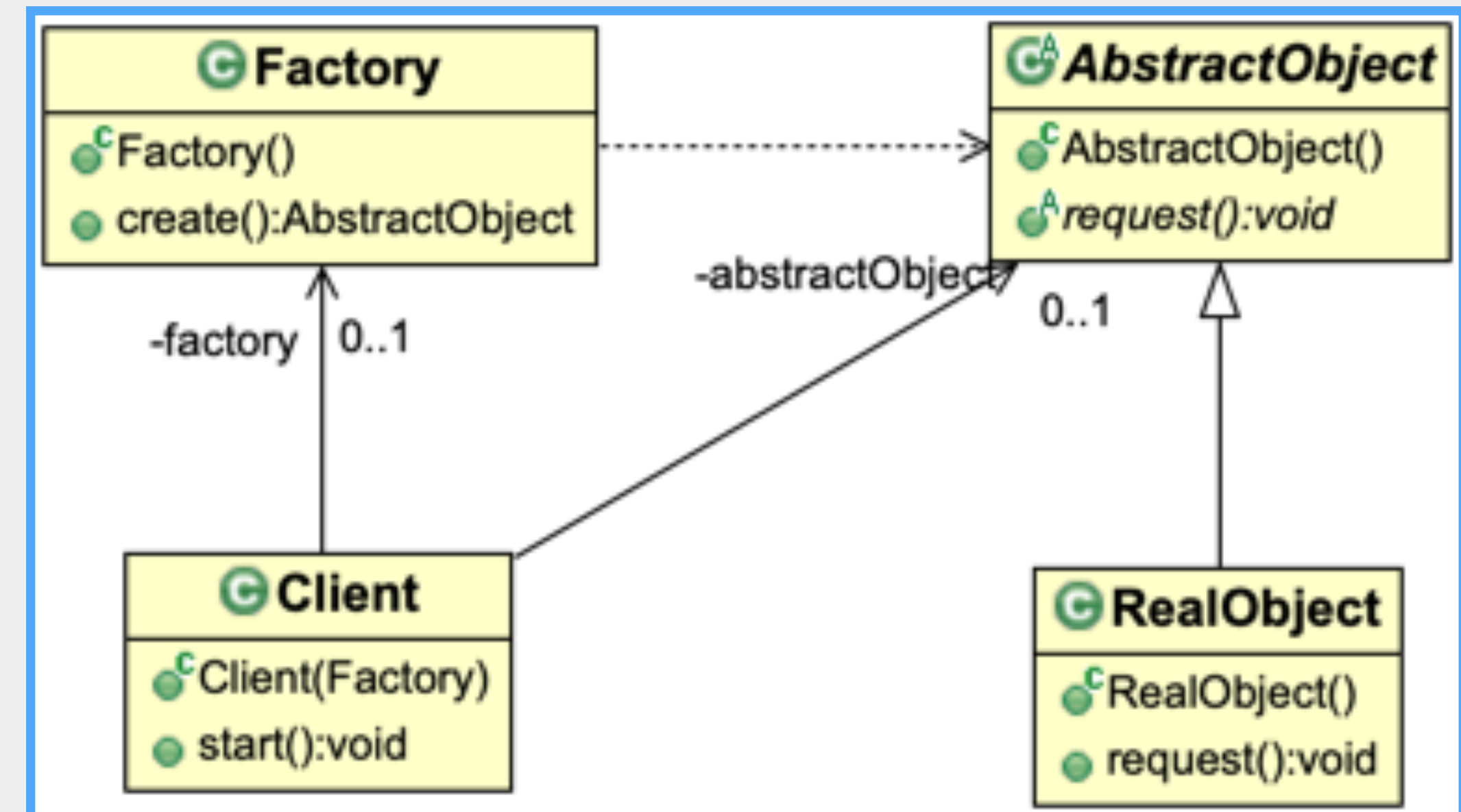


```
String version = computer.getSoundcard().getUSB().getVersion();
```

Null Object



- Java `nullObject.problem`
- C# `NullObjectProblem`



```
public class Client {
    ...
    public void start() {
        abstractObject = factory.create();
        if(abstractObject != null)
            abstractObject.request();
        else
            System.err.println("Null object!");
    }
}
```

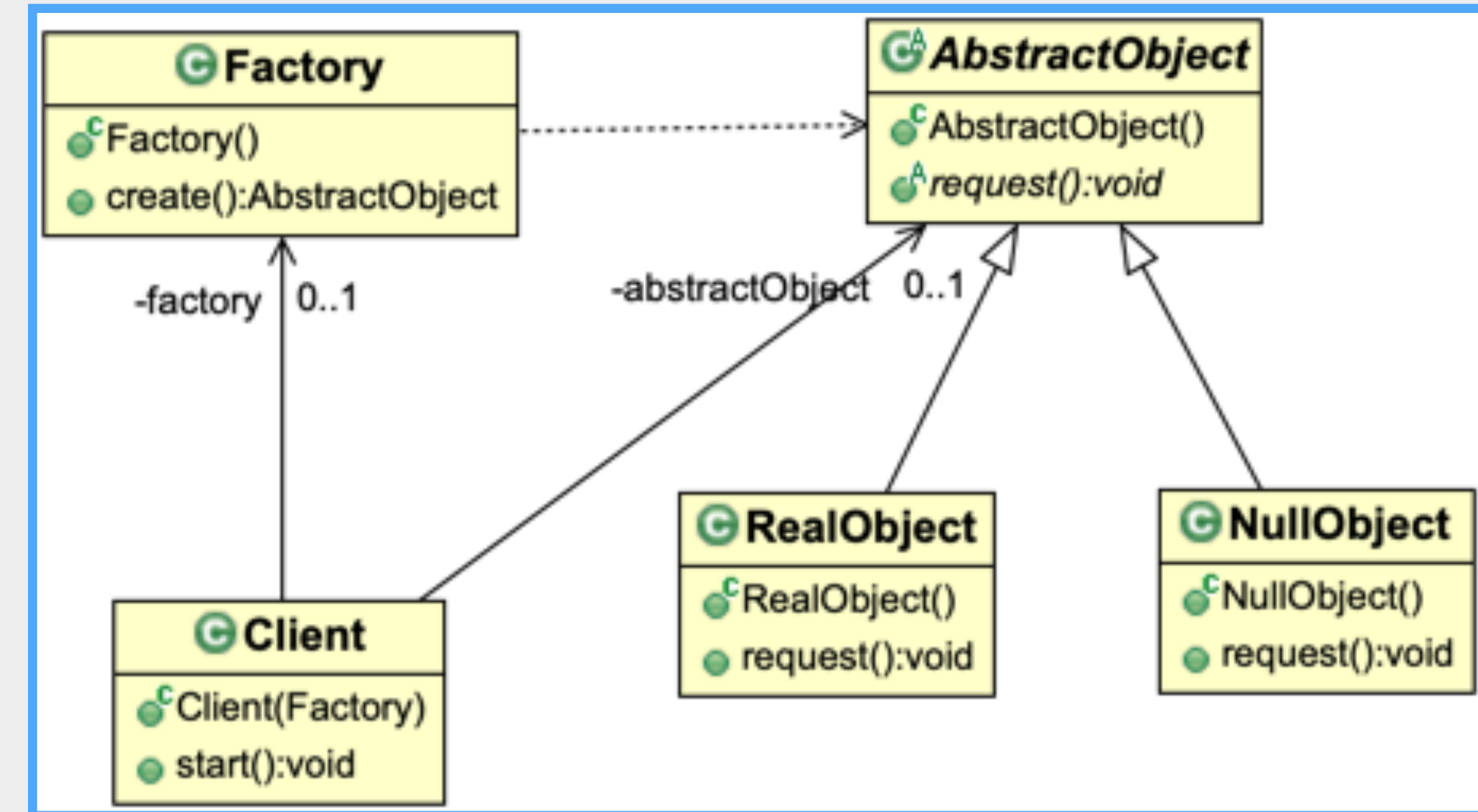


Çözüm

Çözüm - I



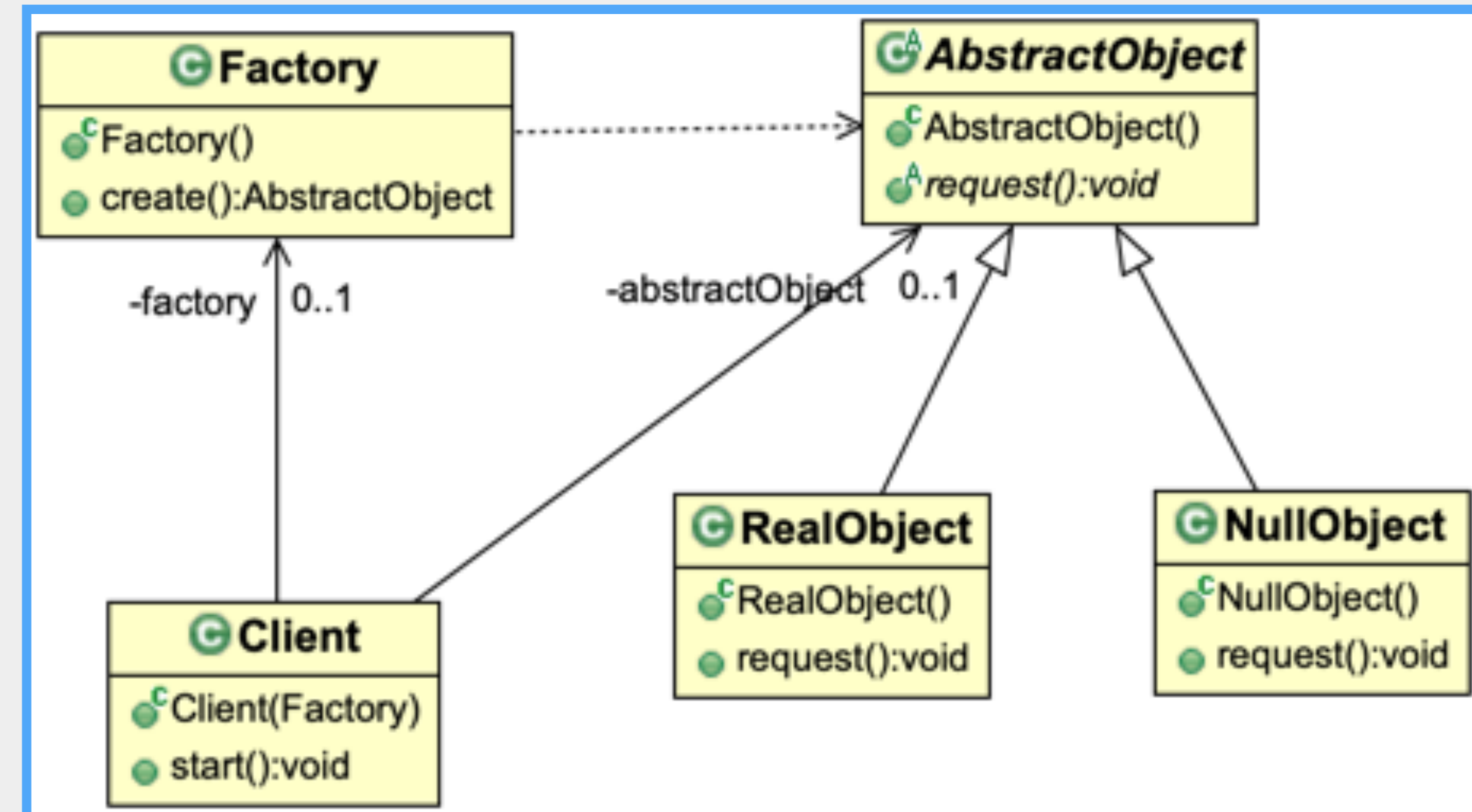
- Yazarı tarafından yapısal (structural) olarak nitelendirilen **Null Object** kalıbında referansın hiç bir nesneyi göstermediği durumlar için gerçekte istenilen şekilde “hiç bir şey yapma” davranışına sahip bir nesne kullanılır.
- Bu **null nesne**, aslında null olmayan nesnelerle aynı hiyerarşide bulunur ve aynı metotları “hiç bir şey yapma”yacak şekilde gerçekleştirir.



Sorter



- Java `nullObject.pattern`
- C# `NullObject`



```
public class Client {
    ...
    public void start() {
        abstractObject = factory.create();
        abstractObject.request();
    }
}
```



- Bu çözümün belki de en problemli tarafı ne zaman null nesne kullanıldığını yani referansın gerçekte ne zaman null olduğunu bilmemeyi hedeflemesidir.
- Zaten teklif edilen de null nesneni bağlamda kabul gören bir “hiç bir şey yapma” prensibini yerine getirmesi bu sebeptendir.
- Bu yaklaşım her zaman kabul edilebilir olmayabilir.
- Bu gibi sebeplerden dolayı uygulamada ve dillerde farklı çözümler öne çıkmıştır.

Çözüm - III



- Null nesne söz konusu olduğunda sıra dışı durum fırlatmak da bir çözümdür.

```
String version = "UNKNOWN";
try{
    version = computer.getSoundcard().getUSB().getVersion();
}
catch(NoSoundCardFoundException e){
    // Handle the situation of having no sound card
}
catch(NoUSBFoundException e){
    // Handle the situation of having no USB
}
```



- C# gibi dillerde bu durum için ?? şeklinde işlemci de geliştirilmiştir.

```
List<int> numbers = null;
int? a = null;

(numbers ??= new List<int>()).Add(5);
Console.WriteLine(string.Join(" ", numbers)); // output: 5

numbers.Add(a ??= 0);
Console.WriteLine(string.Join(" ", numbers)); // output: 5
0
Console.WriteLine(a); // output: 0
```

- Benzer şekilde C#'da `System.Nullable<T>` da null olması muhtemel referansları ifade eder.

Çözüm - IV



- Benazer şekilde Java SE 8 ile birlikte gelmiş, **final** bir sınıf olan `java.util.Optional` bu amaç içindir.

```
public class Client {  
  
    public void start() {  
        Optional<AbstractObject> o = factory.create();  
        if (o.isPresent()) {  
            AbstractObject abstractObject = o.get();  
            abstractObject.request();  
        } else  
            System.out.println("Nothing in there!");  
        }  
    }  
}
```



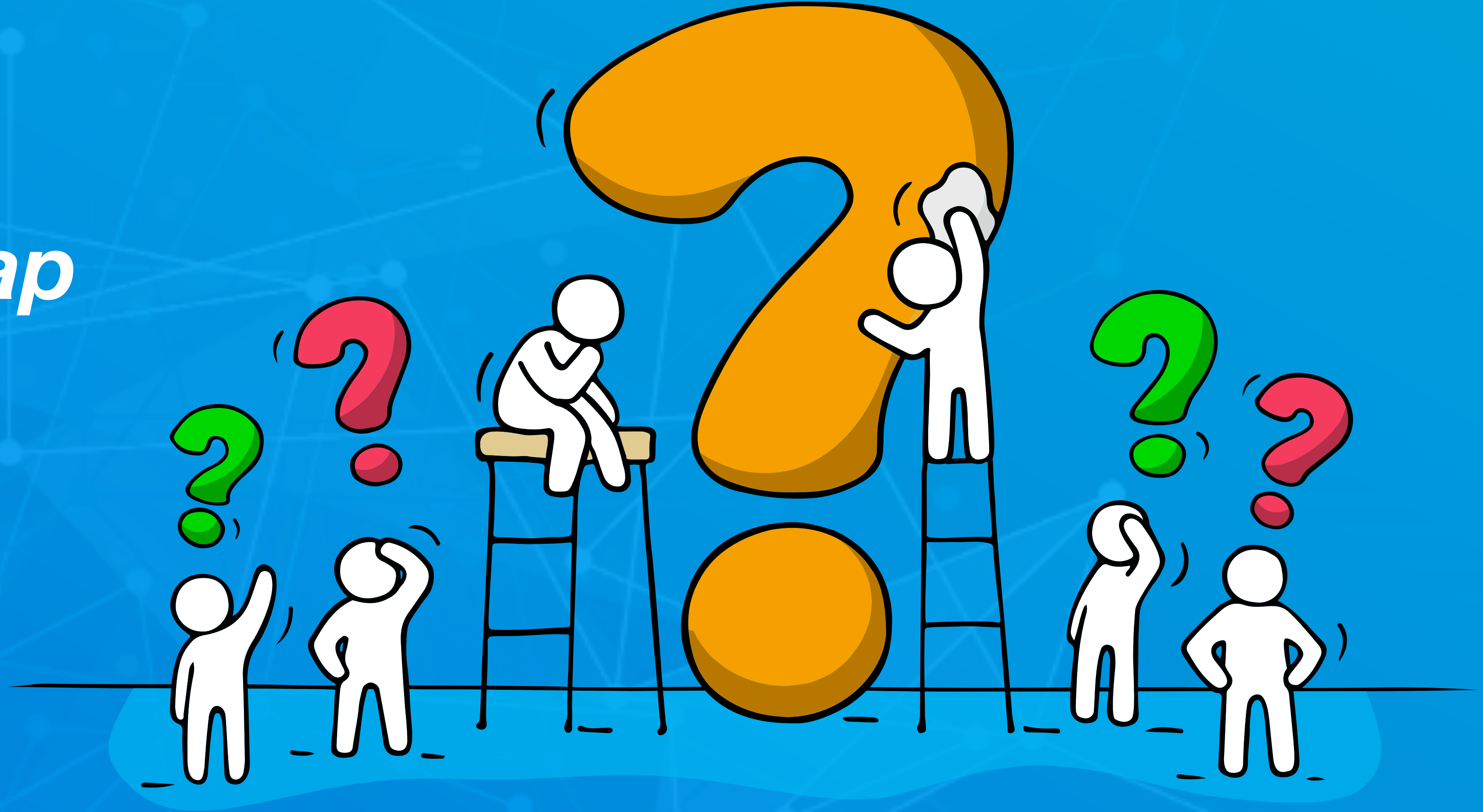

Sonuçlar

Diğer Kalıplarla İlişki



- **Null Object**, **Singleton** olabilir.

*Soru ve Cevap
Zamanı!*



*Soru ve Cevap
Zamanı!*

