Java ile Nesne Merkezli ve Fonksiyonel Programlama

5. Bölüm
Alt Sınıflar (Nested Classes)

Akın Kaldıroğlu

www.javaturk.org

Ocak 2017

www.selsoftacademy

Küçük Ama Önemli Bir Konu

- Bu dosya ve beraberindeki tüm, dosya, kod, vb. eğitim malzemelerinin tüm hakları **Selsoft Yazılım, Danışmanlık, Eğitim ve Tic. Ltd. Şti**.'ne aittir.
- Bu eğitim malzemelerini kişisel bilgilenme ve gelişiminiz amacıyla kullanabilirsiniz ve isteyenleri **http://www.selsoft.academy** adresine yönlendirip, bu malzemelerin en güncel hallerini almalarını sağlayabilirsiniz.
- Yukarıda bahsedilen amaç dışında, bu eğitim malzemelerinin, ticari olsun/olmasın herhangi bir şekilde, toplu bir eğitim faaliyetinde kullanılması, bu amaca yönelik olsun/olmasın basılması, dağıtılması, gerçek ya da sanal/Internet ortamlarında yayınlanması yasaktır. Böyle bir ihtiyaç halinde lütfen benimle, **akin.kaldiroglu@selsoft.academy** adresinden iletişime geçin.
- Bu ve benzeri eğitim malzemelerine katkıda bulunmak ya da düzeltme ve eleştirilerinizi bana iletmek isterseniz çok sevinirim.
- Bol Java'lı günler dilerim.

www.selsoft.academy

İçerik

- Bu bölümde, alt sınıflar (nested classes) ele alınacaktır.

- Bu amaçla aşağıdaki alt sınıflar işlenecektir:
  - Statik alt sınıflar (static nested classes)
  - (Statik olmayan) iç sınıflar (Inner classes)
  - Yerel sınıflar (Local classes)
  - Adsız iç sınıflar (Anonymous inner classes)

www.solsoftacademy

Üst ve Alt Sınıflar

www.selsoft.academy

Yüksek Seviyeli Sınıflar - I

- Herhangi bir başka sınıfın içinde tanımlanmayan sınıflara yüksek seviyeli sınıf (top level class) denir.
  - Bundan sonra üst sınıf olarak da isimlendirileceklerdir.
- Yüksek seviyede tanımlanan sınıflar erişim belirteci olarak sadece **public** anahtar kelimesini alabilirler.
  - Bu durumda yüksek seviyeli sınıf kendi kaynak kodu dosyasında tanımlanmalıdır.
- Bir üst sınıf, public anahtar kelimesinin olmadığı durumda paket erişimine (package access) sahip olur.
  - Bu durumda yüksek seviyeli sınıf herhangi bir kaynak kodu dosyasında tanımlanabilir.

www.selsoft.academy

Yüksek Seviyeli Sınıflar - II

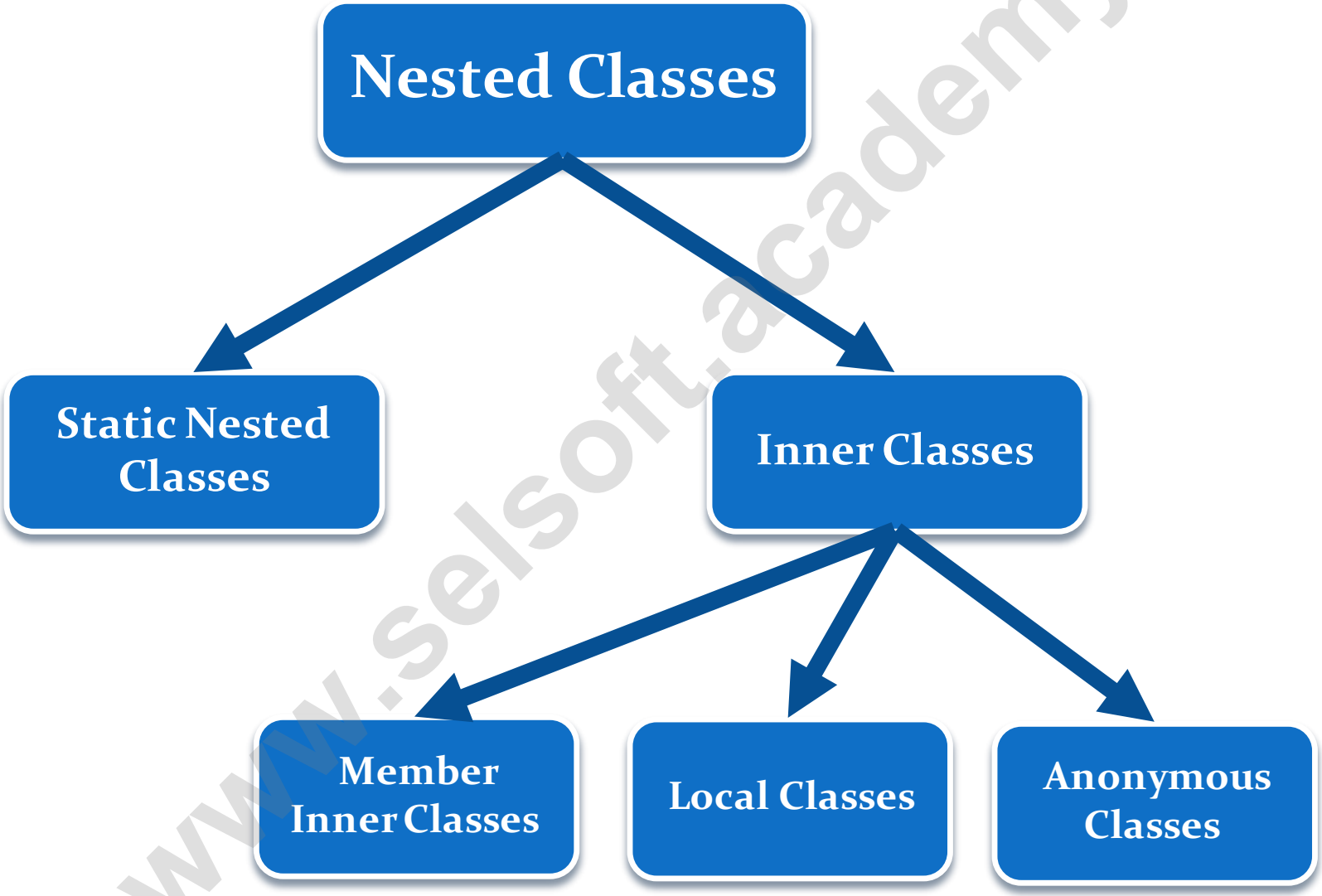
- Yüksek seviyeli sınıflar, protected, private ve static kelimeleriyle betimlenemezler.

www.selsoft.academy

Alt Sınıflar (Nested Classes) - I

- Başka bir sınıfın içinde tanımlanan sınıflara alt sınıf (nested class) denir.
- Alt sınıflar temelde statik olan ve olmayanlar olarak ikiye ayrılır.
  - Statik alt sınıflar (static nested classes)
  - İç sınıflar (inner classes)
- İç sınıfların ise 3 çeşidi vardır:
  - (Statik olmayan) iç sınıflar (Inner classes)
  - Yerel sınıflar (Local classes)
  - Adsız iç sınıflar (Anonymous inner classes)

www.selsoft.academy



www.selsoft.academy

İç Sınıflar (Nested Classes) - II

- Statik olan ve olmayan üye iç sınıflar, içerildikleri sınıfın üyesidirler.
- Yerel sınıflar ve isimsiz sınıflar ise bir sınıfın bir alt blokunda çoğunlukla da bir metodunda tanımlanırlar.
- Statik olmayan iç sınıflar, parçası oldukları sınıfın tüm üyelerine, private tanımlansalar bile, ulaşabilirler.
- Statik iç sınıflar, parçası oldukları sınıfın sadece static üyelerine, private tanımlansalar bile, ulaşabilirler.
- Üye iç sınıflar, statik olsun – olmasın, public , protected, private, ve paket erişimli olarak tanımlanabilirler.

www.selsoft.academy

NestedClassesExample.java

www.selsoft.academy

Neden Alt Sınıf?

- Alt sınıf kullanımının en temelde 3 sebebi vardır:
  - Sadece bir yerde kullanılan bir grup özelliğin mantıksal olarak bir bütün halinde ifadesi için.
  - Sarmalamayı (encapsualtion) arttırmak için.
    - Bir sınıfın private tanımlanan alanlarına ulaşmak amacıyla.
  - Bir sınıfı, içinde kullanıldığı sınıfa en yakın yerde tanımlayarak okunabilirliği ve bakımı kolaylaştırmak.

www.selsoft.academy

Statik İç Sınıflar

www.selsoft.academy

Statik İç Sınıflar (Static Nested Classes)

- Statik iç sınıflar, statik metotlar gibi, sadece içinde tanımlandığı sınıfın statik alanlarına doğrudan ulaşabilirler.
- Statik iç sınıflar, diğer yüksek seviyeli sınıflar gibi, içinde bulunduğu sınıfın nesne değişkenlerine ve metotlarına, nesneleri üzerinden ulaşabilirler.
- Statik iç sınıflara, içinde bulundukları sınıf üzerinden ulaşılır.
- Ayrıca statik iç sınıfın nesnesini oluştururken özel bir söz dizimine gerek vardır.

```
OuterClass.StaticInnerClass o =
                new OuterClass.StaticInnerClass();
```

www.selsoft.academy

NestedStaticClassExample.java

www.selsoft.academy

Neden Statik İç Sınıflar?

- Eğer bir sınıfın bir alan kümesi, o sınıfın nesnelerinden bağımsız, sadece sınıfa bağlı olarak mantıksal bir grupta toplanmak istenirse, statik iç sınıf olarak kurgulanabilir.

- Statik iç sınıfın nesnesinin, içinde bulunduğu sınıfın nesnelerinin ortak durumu olduğu da hatırlanmalıdır.

www.selsoft.academy

Citizen.java

www.selsoft.academy

Üye İç Sınıflar

www.selsoft.academy

Üye İç Sınıflar

- Statik olmayan üye iç sınıflar (non-static member classes), içinde bulunduğu sınıfın nesnelerine özel durumu ifade etmede kullanılır.
- Temelde, beraberce bir anlam taşıyan bir özellik kümesinin mantıksal olarak gruplanması için kullanılır.
- Üye iç sınıfın nesnesini oluşturmak için önce üst sınıfın nesnesi oluşturulmalıdır.
- Ayrıca üye iç sınıfın nesnesini oluştururken özel bir söz dizimine gerek vardır.

```
OuterClass oc = new OuterClass();
OuterClass.InnerClass o = oc.new InnerClass();
```

www.selsoftacademy

Car.java

www.selsoft.academy

Yerel İç Sınıflar

www.selsoft.academy

Yerel İç Sınıflar (Local Inner Classes) - I

- Yerel iç sınıflar (local inner classes), adından da anlaşıalcağı üzere, bir sınıfın bir alt kapsamında (scope) tanımlanan sınıflardır.
  - Alt kapsam tipik olarak metottur.
- Yerel iç sınıfların kapsamı içinde bulundukları blok ile kısıtlıdır.
- Yerel sınıflar, içinde bulundukları sınıfların değişkenlerine ulaşabilirler.
- Statik metotta tanımlanan bir yerel sınıf sadece statik özelliklere ve metotlara ulaşırken, bir nesne metodunda tanımlanan bir yerel sınıf tüm üyelere erişebilir.

www.sdsoft.academy

Yerel İç Sınıflar (Local Inner Classes) - II

- Yerel sınıflar final olmadıkça statik özellik de tanımlayamazlar.

www.selsoft.academy

Yerel İç Sınıflar (Local Inner Classes) - II

- Ayrıca yerel sınıflar, içinde bulundukları bloğun yerel değişkenlerine de ulaşabilirler.
  - Fakat bu değişkenlerin final olması gereklidir.
  - Java 1.8'den itibaren doğrudan final olarak tanımlanmadığı halde ilk değer atamasından sonra değeri değişmeyen (effective final) değişkenlere de ulaşabilirler.

www.selsoft.academy

LocalClassExample.java

www.selsoft.academy

İsimsiz İç Sınıflar

www.selsoft.academy

İsimsiz Sınıflar (Anonymous Classes)

www.selsoft.academy

İsimsiz İç Sınıflar

- İsimsiz iç sınıflar (anonymous inner classes) Arayüzler (Interfaces) bölümünde detaylıca ele alınmıştı.
- Konu bütünlüğü açısından aynı malzeme buraya da konmuştur.

www.selsoft.academy

İsimsiz Sınıflar

- Genelde geri çağırma metotlarının üzerinde bulunduğu sınıfların tek kullanımlık bir nesnesine ihtiyaç duyulur.
  - Yani arayüzü gerçekleştiren sınıfın bir tek nesnesine ihtiyaç vardır ve bu nesne sadece bir yerde kullanılır.

- Bu durumda Java, arayüzü yerine getiren sınıfın isimsiz bir şekilde, hızlıca oluşturulmasına ve bunun yapıldığı yerde tek bir nesnesinin yaratılıp kullanılmasına izin verir.

- Bu şekilde oluşturulan sınıflara isimsiz sınıf (anonymous class) denir.

www.selsoft.academy

İsimsiz Sınıflar - I

- İsimsiz sınıflar sıklıkla olayları (event) yakalamada kullanılırlar.
  - Çünkü çoğu zaman özel bir duruma işaret eden olay nesnesi sadece bir yerde yakalanır ve gereği yapılır.

```
Timer t = new Timer(1_000, new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        ...
    }
});
```

- Örnekteki **`ActionListener`**, sadece **`actionPerformed()`** metoduna sahip bir arayüzdür.

www.sersoftacademy

TimerExample.java

- *org.javaturk.oofp.ch04.anonymous.timer* paketi.

www.selsoft.academy

İsimsiz Sınıflar - I

- Yakalanan sıklıkla GUI olaylarıdır..
- Benzer şekilde çoğu zaman bir GUI bileşeninin durumundaki bir değişikliğe işaret eden olay nesnesi sadece bir yerde yakalanır ve gereği yapılır.

```
button.setOnAction(new EventHandler<ActionEvent>() {

        @Override
        public void handle(ActionEvent event) {
                ...
        }
});
```

- Örnekteki **`EventHandler`**, sadece **`handle()`** metoduna sahip bir arayüzdür.

www.selsoft.academy

MyApplication.java

- *org.javaturk.oofp.ch04.anonymous.event* paketi.

www.selsoft.academy

İsimsiz Sınıflar - II

- Sınıflar, tanımlamaya (class declaration) sahip oldukları halde isimsiz sınıflar ifadedirler (expression).
- İsimsiz sınıf ifadesi, bir kurucu çağrısına benzer ama içinde tekrar tanımlanan (override) metot ya da metotlar vardır.
- İsimsiz sınıflar hem arayüzleri gerçekleştirmede hem de sınıfları genişletmede kullanılabilirler.
- İsimsiz sınıflar genelde sadece bir metota sahip arayüzleri gerçekleştirmede kullanılmalarına rağmen birden fazla metodu yeniden tanımlayacak şekilde kullanılabilirler.
  - Olay yapılarında çağrılacak metot bir tane olduğundan, genelde tek metodu tekrar tanımlamada kullanılırlar.

www.selsoft.academy

İsimsiz Sınıflar - III

- İsimsiz sınıf ifadesi şöyledir:
  - new operatörü,
  - Gerçekleştirilecek arayüzün ya da genişletilecek sınıfın ismi,
    - new operatöründen sonra gelen tipin sınıf olması durumunda, kurucuya geçilecek parametreler de sıralanabilir.
    - Eğer tip arayüz ise, arayüzlerin kurucuları olmadığından, sanki varsayılan kurucu çağrılıyormuş gibi içi boş iki parantez bulunur.
  - Sınıf bloğu.
- İsimsiz sınıflar birer ifade olduklarından, bloklarında başka ifadeler olamaz, sadece metot gibi başka bloklar olabilir.
- İsimsiz sınıf ifadesi, arayüz gerçekleştirmesinde new operatöründen sonra arayüzün varsayılan kurucusunu çağırıyor bir görüntüye sahip olduğundan tuhaf görünür.

www.selsoftacademy

```
public interface DoerInterface {

        void doIt();

        void doThat();
}
```

```
new DoerInterface(){
    {
        System.out.println("Instance initializer block.");
    }

    @Override
    public void doIt(){
        System.out.println("I!ll always do it :)");
    }

    @Override
    public void doThat(){
        System.out.println("I'll always do that :)");
    }
}).doIt();
```

www.selsoft.academy

AnonymousDoesClassTest.java

- *org.javaturk.oofp.ch04.anonymous.doer* paketi.

www.selsoft.academy

İsimsiz Sınıflar - IV

- İsimsiz sınıflar, içinde bulundukları sınıfın üyelerine erişebilir.
- İsimsiz sınıflar, içinde bulundukları bloğun yerel değişkenlerine final ya da değeri değişmediği (effectively final) hallerde ulaşabilir.
  - Bu durumda da yerel değişkeni değiştiremez.
- İsimsiz sınıflar, sabite olmaları şartıyla statik alanlar tanımlayabilirler.

www.selsoft.academy

İsimsiz Sınıflar - V

- İsimsiz sınıflar ayrıca şunları tanımlayabilirler:
  - Alanlar,
  - Yerel sınıflar (local classes),
  - Üst tipinde olmayan metotlar,
  - Nesne ilk değer blokları
- İsimsiz sınıflar, statik ilk değer atama blokları ile üye arayüzler tanımlayamazlar.

www.selsoft.academy

WeirdAnonymousDoesClassTest.java

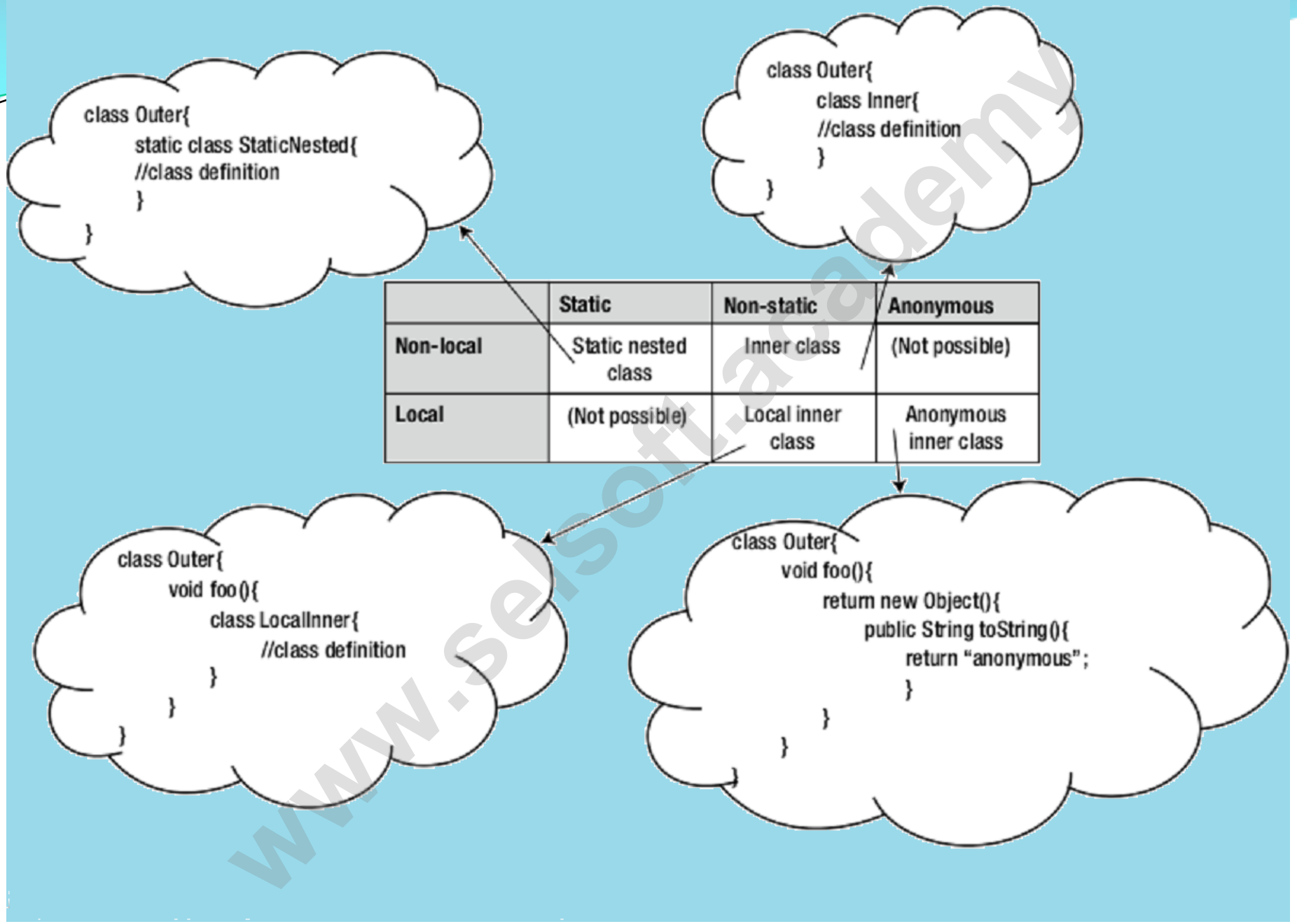
- *org.javaturk.oofp.ch04.anonymous.doer* paketi.

www.selsoft.academy

Özet

- Bu bölümde, üst-alt sınıf kavramı ve alt sınıf çeşitleri ele alındı.

www.selsoft.academy



www.selsoft.academy

Ödevler

www.selsoft.academy

Ödevler

- 4 tip alt sınıfa birer örnek verip kodlarını yazın.

www.selsoft.academy