

## Αντικειμενοστρεφής Προγραμματισμός

Παναγιώτης Αδαμίδης  
adamidis@it.teithe.gr

### Εσωτερικές κλάσεις

## Εσωτερικές κλάσεις

- Μία εσωτερική κλάση (inner class) είναι δηλωμένη μέσα σε μια άλλη κλάση.
- Μία εσωτερική κλάση είναι γνωστή μόνο μέσα στην κλάση που την περιβάλλει. Επομένως η εμβέλεια μιας εσωτερικής κλάσης περιορίζεται από την εξωτερική κλάση που την περιβάλλει.
- Μία εσωτερική κλάση έχει πρόσβαση στα μέλη της κλάσης που την περιβάλλει, συμπεριλαμβανομένων και των "private" μελών, ενώ η εξωτερική κλάση δεν έχει πρόσβαση στα μέλη της εσωτερικής.

### Εσωτερικές κλάσεις Παράδειγμα

```
class Outer {
    int nums[];
    Outer (int n[]) { nums=n; }
    void Analyze() {
        Inner inOb = new Inner();
        System.out.println("Minimum: " + inOb.min());
        System.out.println("Maximum: " + inOb.max());
        System.out.println("Average: " + inOb.avg());
    } //void Analyze

    class Inner {
        int min() {
            int m=nums[0];
            for (int i=0; i<nums.length; i++)
                if (nums[i]<m) m=nums[i];
            return m;
        }
    }
}
```

*συνεχίζεται*

### Εσωτερικές κλάσεις Παράδειγμα (συνέχεια\_1)

```
int max() {
    int m=nums[0];
    for (int i=0; i<nums.length; i++)
        if (nums[i]>m)
            m=nums[i];
    return m;
}

int avg() {
    int a=0;
    for (int i=0; i<nums.length; i++)
        a+=nums[i];
    return a/nums.length;
} // class Inner
} // class Outer
```

*συνεχίζεται*

### Εσωτερικές κλάσεις Παράδειγμα (συνέχεια\_2)

```
class NestedClassDemo {
    public static void main (String[] args) {
        int x[] = {3, 2, 1, 5, 6, 9, 7, 8};
        Outer outOb = new Outer(x);
        outOb.Analyze();
    }
}
```

- **ΕΞΟΔΟΣ:** Minimum: 1  
Maximum: 9  
Average: 5
- Πως μπορεί η μέθοδος analyze() να έχει πρόσβαση στη μέθοδο min();  
⚡ Απάντηση: Δημιουργία αντικειμένου **Inner**

### Παράδειγμα με 2 Εσωτερικές κλάσεις

```
public class Parcel1 {
    class Contents {
        private int i = 11;
        public int value() {
            return i;
        }
    }
    class Destination {
        private String label;
        Destination(String whereTo) {
            label = whereTo;
        }
        String readLabel() {
            return label;
        }
    }
}
```

*συνεχίζεται*

## Παράδειγμα με 2 Εσωτερικές κλάσεις (συνέχεια)

```
// Using inner classes looks just like
// using any other class, within Parcel1:
public void ship(String dest) {
    Contents c = new Contents();
    Destination d = new Destination(dest);
    System.out.println(d.readLabel());
}
public static void main(String[] args) {
    Parcel1 p = new Parcel1();
    p.ship("Tanzania");
}
}
```

## Επιστροφή αναφοράς σε εσωτερική κλάση

```
public class Parcel2 {
    class Contents {
        private int i = 11;
        public int value() { return i; }
    }
    class Destination {
        private String label;
        Destination(String whereTo) {
            label = whereTo;
        }
        String readLabel() { return label; }
    }
    public Destination to(String s) {
        return new Destination(s);
    }
}
```

## Επιστροφή αναφοράς σε εσωτερική κλάση (συνέχεια)

```
public Contents cont() {
    return new Contents();
}
public void ship(String dest) {
    Contents c = cont();
    Destination d = to(dest);
    System.out.println(d.readLabel());
}
public static void main(String[] args) {
    Parcel2 p = new Parcel2();
    p.ship("Tanzania");
    Parcel2 q = new Parcel2();
    // Defining references to inner classes:
    Parcel2.Contents c = q.cont();
    Parcel2.Destination d = q.to("Borneo");
}
}
```

## Εσωτερικές κλάσεις και Upcasting

- Μια κατάσταση όπου πραγματικά φαίνεται η χρησιμότητα των εσωτερικών κλάσεων είναι το "upcasting". Αυτό γιατί η εσωτερική κλάση - η υλοποίηση της διασύνδεσης - μπορεί να είναι τελείως εκτός πρόσβασης από οπουδήποτε αφού το μόνο που έχουμε είναι μία αναφορά στη βασική κλάση ή την διασύνδεση. Ακολουθεί ένα παράδειγμα.

```
// File: Contents.java
public interface Contents {
    int value();
}
// File: Destination.java
public interface Destination {
    String readLabel();
}
}
```

## Εσωτερικές κλάσεις και Upcasting (2)

```
// File: Parcel3.java
// Returning a reference to an inner class.

public class Parcel3 {
    private class PContents implements Contents {
        private int i = 11;
        public int value() { return i; }
    }
    protected class PDestination
        implements Destination {
        private String label;
        private PDestination(String whereTo) {
            label = whereTo;
        }
        public String readLabel() { return label; }
    }
}
```

## Εσωτερικές κλάσεις και Upcasting (3)

```
public Destination dest(String s) {
    return new PDestination(s);
}
public Contents cont() {
    return new PContents();
}
}

class Test {
    public static void main(String[] args) {
        Parcel3 p = new Parcel3();
        Contents c = p.cont();
        Destination d = p.dest("Tanzania");
        // Illegal -- can't access private class:
        //!! Parcel3.PContents pc = p.new PContents();
    }
}
```

### **κλάσεις Εσωτερικές σε Μεθόδους**

- Είναι δυνατό να οριστεί μια κλάση μέσα σε μια μέθοδο ή σε ένα block με οποιαδήποτε εμβέλεια.
- Έτσι δημιουργείται μία εσωτερική κλάση η οποία δεν είναι γνωστή έξω από το block.