



Ανάλυση Πληροφοριακών Συστημάτων

**«Βασικές Έννοιες
Αντικειμενοστρεφούς
Προγραμματισμού –
Διαγράμματα κλάσεων»
Βασίλειος Καρακώιδας**

7 – 12 – 2005



Περιεχόμενα Παρουσίασης

- Βασικές Έννοιες Αντικειμενοστραφούς προγραμματισμού
- Διαγράμματα Κλάσεων
- Ασκήσεις στην τάξη



Διαγράμματα της UML

- Διάγραμμα περιπτώσεων χρήσης (*use case diagram*)
- Διαγράμματα δομής
 - Διάγραμμα κλάσεων (*class diagram*)
 - Διάγραμμα αντικειμένων (*object diagram*)
- Διαγράμματα συμπεριφοράς
 - Διάγραμμα καταστάσεων (*statechart diagram*)
 - Διάγραμμα δραστηριοτήτων (*activity diagram*)
 - Διαγράμματα αλληλεπίδρασης
 - Διάγραμμα ακολουθίας (*sequence diagram*)
 - Διάγραμμα συνεργασίας (*collaboration diagram*)
- Διαγράμματα δομής υλοποίησης
 - Διάγραμμα εξαρτημάτων (*component diagram*)
 - Διάγραμμα ανάπτυξης (*deployment diagram*)



Χρήση Διαγραμμάτων

<i>Διεργασία</i>	<i>Διάγραμμα</i>	<i>Απεικονίζει</i>
Ανάλυση απαιτήσεων	Διάγραμμα περιπτώσεων χρήσης (use case)	Οπτική χρήση στην συμπεριφορά του συστήματος
	Διάγραμμα δραστηριοτήτων (activity)	Δραστηριότητες χρήστη – αλληλεπίδραση με το σύστημα
	Διάγραμμα κλάσεων (class)	Βασικές οντότητες συστήματος και σχέσεις τους
	Διάγραμμα καταστάσεων (statechart)	Κύκλος ζωής σημαντικών δραστηριοτήτων
Σχεδιασμός	Διάγραμμα κλάσεων (class diagram)	Εσωτερική δομή συστήματος
	Διαγράμματα αλληλεπίδρασης (Sequence/Collaboration)	Αλληλεπίδραση των κλάσεων του συστήματος
	Διάγραμμα εξαρτημάτων (component)	Απεικόνιση του συστήματος σε επαναχρησιμοποιούμενα τμήματα
	Διάγραμμα ανάπτυξης (deployment)	Φυσική απεικόνιση του λογισμικού
	Διάγραμμα καταστάσεων (statechart)	Κύκλος ζωής βασικών κλάσεων
Τεκμηρίωση	Όλα τα διαγράμματα	Διαγραμματική υποστήριξη στην τεκμηρίωση του λογισμικού



Κλάσεις

- Περιγράφουν μια διακριτή οντότητα του συστήματος που μοντελοποιείται
- Καθορίζεται ένα σύνολο αντικειμένων με κατάσταση και συμπεριφορά
- Κατάσταση = Χαρακτηριστικά και Σχέσεις
- Συμπεριφορά = Μεθόδους



Κλάσεις (2)

Medicine
name : String cost : float FPA : float requiresPrescription : boolean
getCost() : float setFPA(percentage : float) : void

Class Name

Attributes

Operations



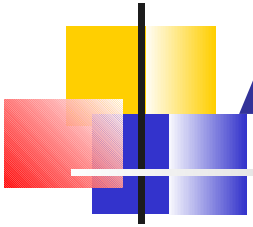
Κλάσεις (3)

```
Public class Medicine {  
    private String name;  
    private float cost, FPA;  
    private boolean requiresPrescription;  
  
    public float getCost() {  
        return cost;  
    }  
  
    public void setFPA(float percentage) {  
        this.FPA = percentage;  
    }  
}
```



Διεπαφή

- Περιγραφή συμπεριφοράς χωρίς υλοποίηση ή κατάσταση
- Περιέχει μεθόδους, αλλά δεν έχει χαρακτηριστικά ή σχέσεις
- Η συμπεριφορά μιας διεπαφής μπορεί να υλοποιείται από μια ή περισσότερες κλάσεις



Διεπαφή



~~Reserve~~

~~updateReserve(quantity: int): int
isAvailale(): boolean~~

~~InterfaceName~~

~~Operations~~



Διεπαφή

```
<<Interface>>
```

```
Reservable
```

```
updateReserves(in quantity : int) : int
```

```
isAvailable() : boolean
```

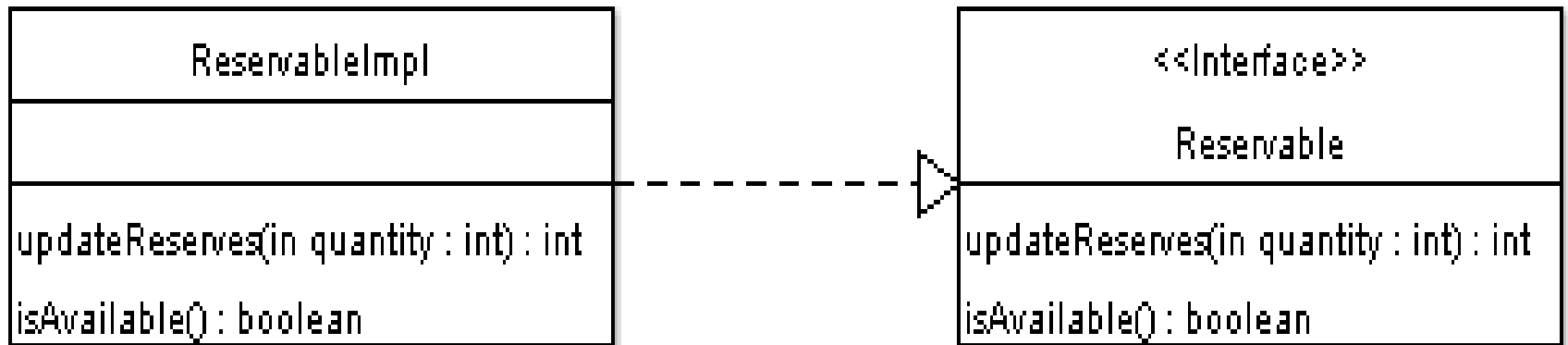


Διεπαφή

```
public interface Reservable {  
    public int updateReserves(int quantity);  
    public boolean isAvailable();  
}
```



Διεπαφή





Διεπαφή

```
Public ReservableImpl implements Reservable {  
    private int quantity;  
    private boolean available = true;  
  
    public int updateReserves(int quantity) {  
        this.quantity = quantity;  
        return this.quantity;  
    }  
  
    public boolean isAvailable() {  
        return available;  
    }  
}
```



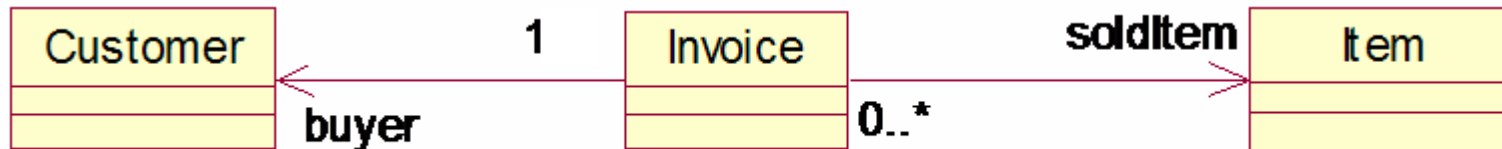
Είδη Σχέσεων

- Συσχέτιση (Association)
 - Σύνθεση (Aggregation/Composition)
- Γενίκευση (Generalization)
- Εξάρτηση (Dependency)

Συσχέτιση



Κάθε πελάτης είναι ασφαλισμένος σε ένα ή περισσότερα ασφαλιστικά ταμεία

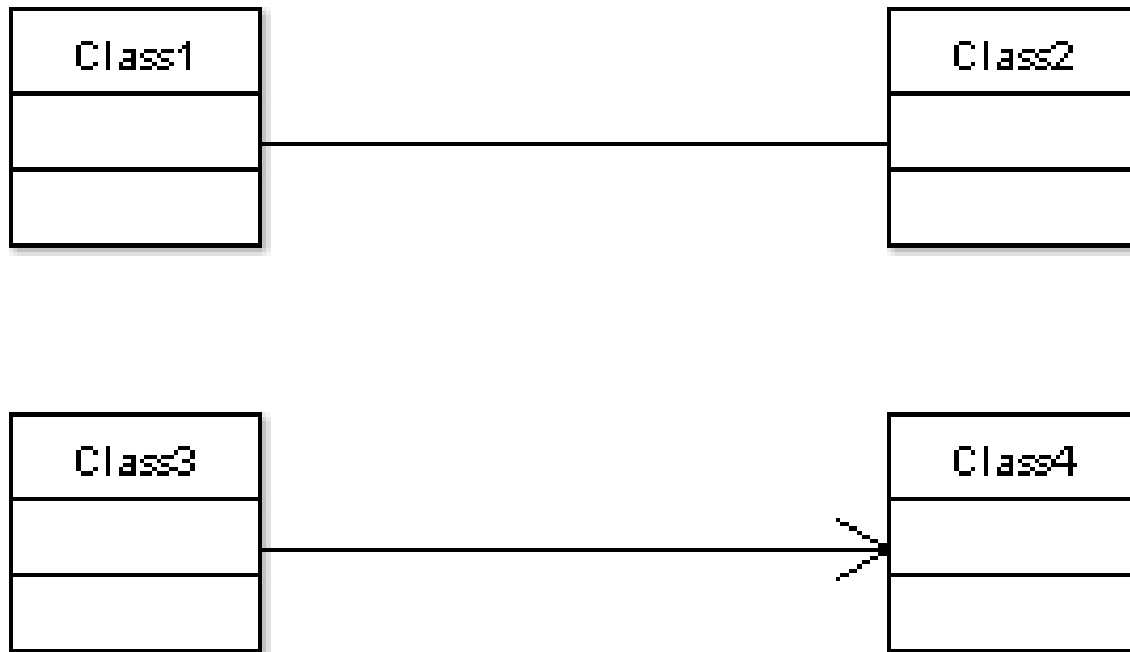


Ένα παραστατικό περιλαμβάνει ακριβώς ένα αγοραστή

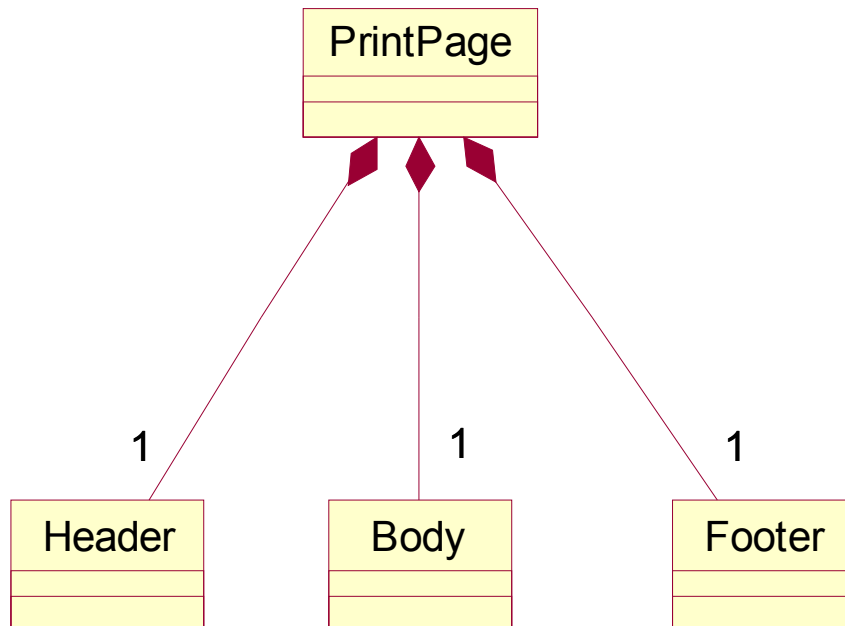
Ένα παραστατικό περιλαμβάνει 1 ή περισσότερα είδη προς πώληση



Συσχέτιση



Σύνθεση (Composition)



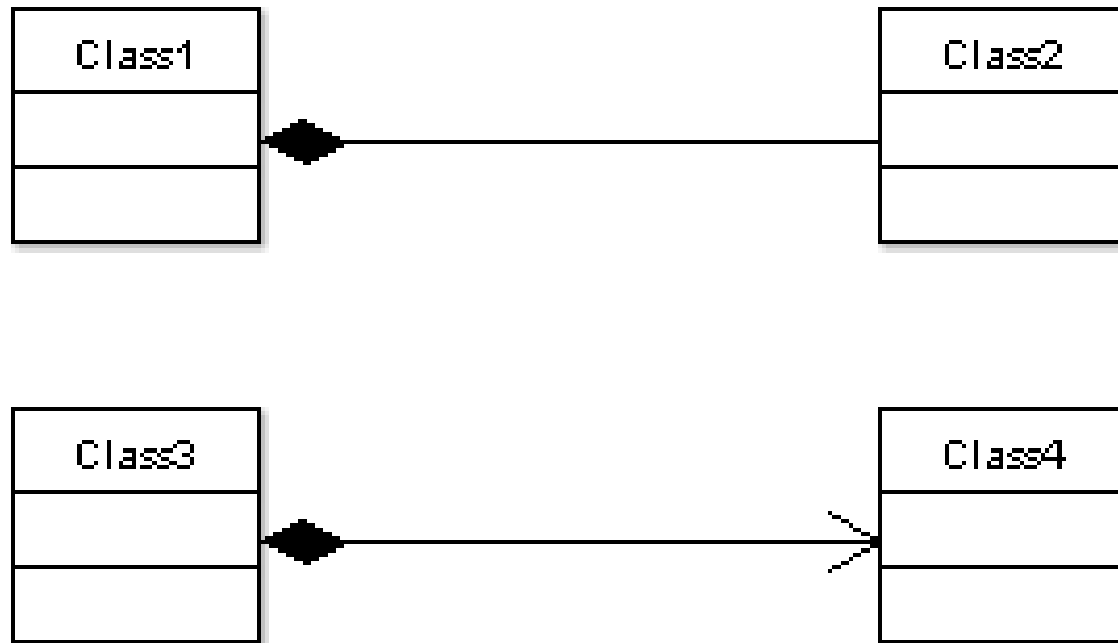
Μια σελίδα εκτύπωσης έχει
κεφαλίδα, κυρίως σώμα
εκτύπωσης και υποσέλιδο



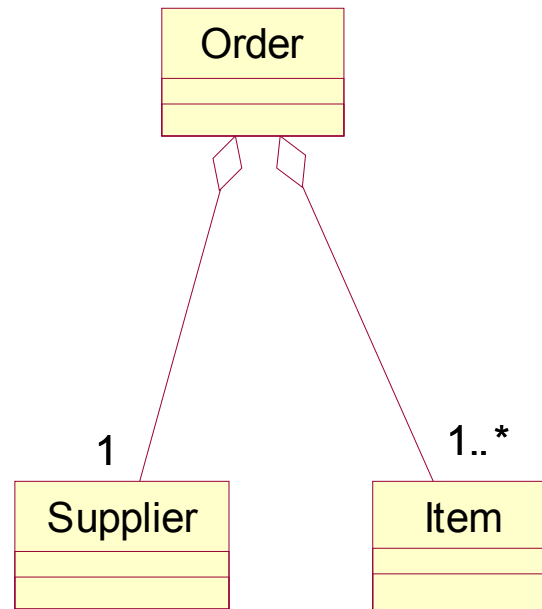
Σύνθεση (Composition)

```
public class PrintPage {  
    public Header header;  
    public Body body;  
    public Footer footer;  
}
```

Σύνθεση (Composition)



Σύνθεση (Aggregation)



Aggregation: Μια παραγγελία έχει ένα προμηθευτή και ένα ή περισσότερα προϊόντα

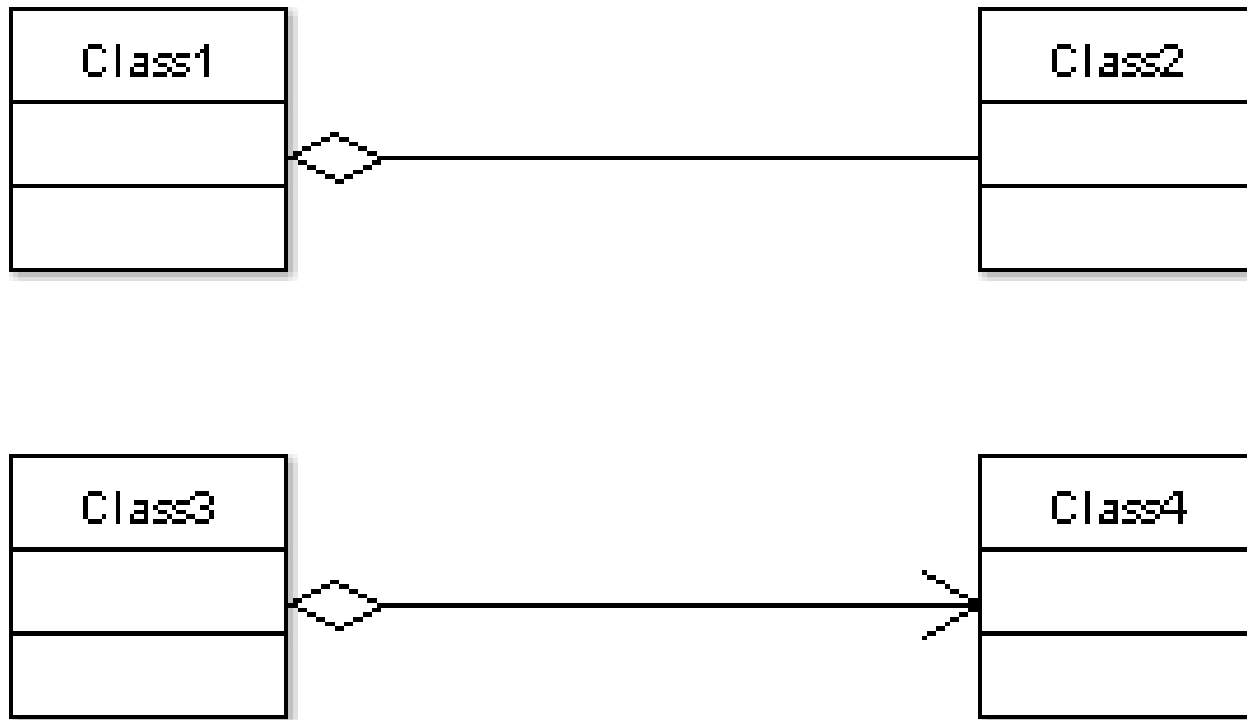


Σύνθεση (Aggregation)

```
public Order {  
    public Supplier supplier;  
    public Item[] items;  
}
```

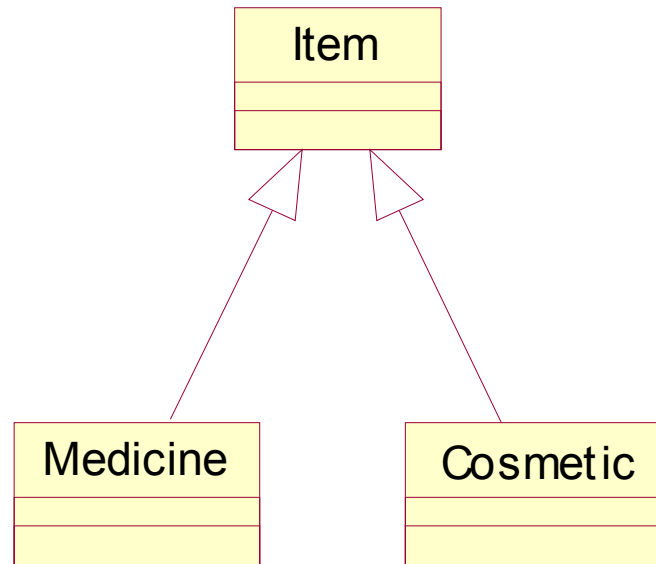


Σύνθεση (Aggregation)





Γενίκευση



Κάθε φάρμακο είναι ένα είδος προϊόντος.

Κάθε καλλυντικό είναι ένα είδος προϊόντος.



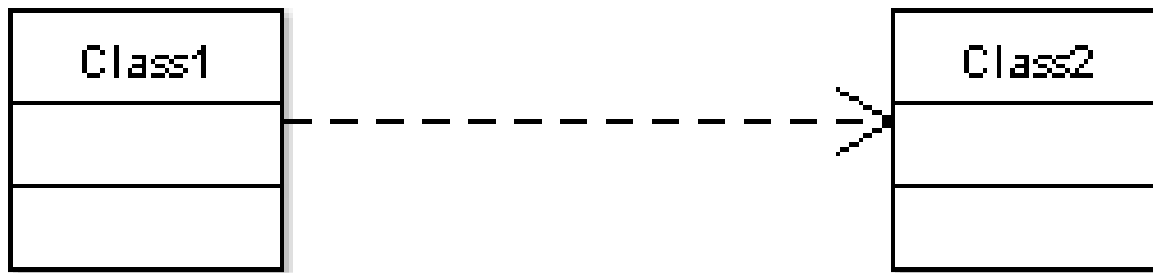
Γενίκευση (2)

```
public class Medicine extends Item {}
```

```
public class Cosmetic extends Item {}
```

```
Public class Item {}
```


Εξάρτηση



- Σχέση εξάρτησης μεταξύ των Class1, Class2.
- Αν αλλάξει η Class2, τότε πιθανό να χρειάζεται αλλαγή και η Class1.



Άσκηση

```
public class Class1 {}

public interface Interface1 {
    public int getInt();
}

public class Class2 implements Interface1 {
    public Class1 instance1;
}

public class Class3 {
    public Class1[] instance1;
    public Class2 instance2;

    public Class1[] getInstance1() { return instance1; }
    public Class2 getInstance2() { return instance2; }
}
```



Άσκηση

```
public class Wheel {}  
public class Engine {}  
public class Gear {}
```

```
public class Driver {}
```

```
public interface Car {}
```

```
public class Mercedes implements Car {  
    public Wheel[] wheels = new Wheel[4];  
    public Engine engine;  
    public Gear gear;  
    public Driver driver;  
}
```