



Ανάλυση Πληροφοριακών Συστημάτων

**«Διαγράμματα *Sequence* και
Collaboration»**

Βασίλειος Καρακώιδας

19 – 12 – 2005



Περιεχόμενα Παρουσίασης

- Διαγράμματα Sequence
- Διαγράμματα Collaboration
- 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



Διαγράμματα Αλληλεπίδρασης

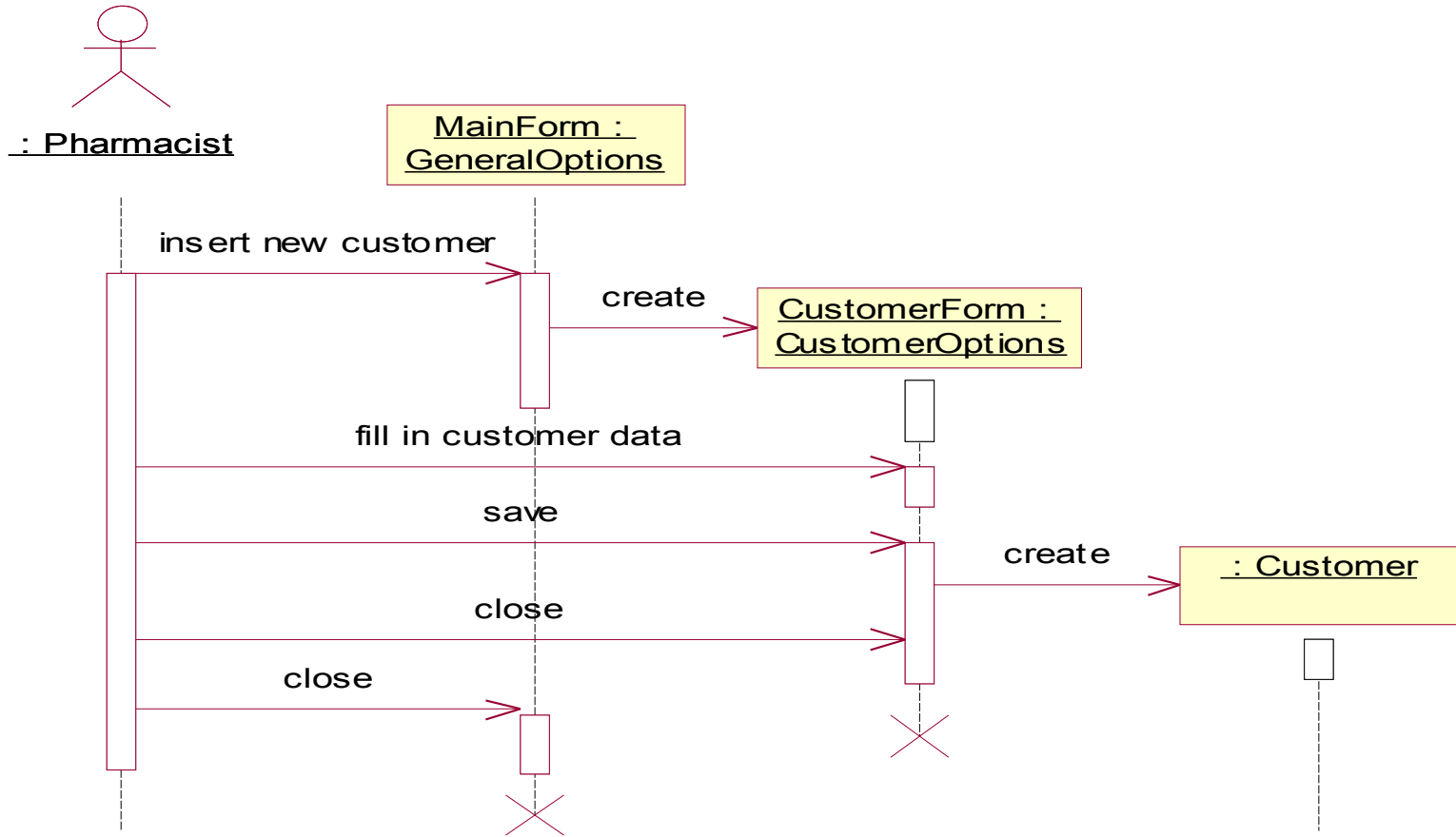
- Περιγράφουν σειρές μηνυμάτων που ανταλλάσσονται μεταξύ οντοτήτων, τα οποία υλοποιούν την συμπεριφορά του συστήματος
- Αποτυπώνει την ροή των δεδομένων ανάμεσα στα αντικείμενα
- Υπάρχουν δύο είδη διαγράμματος
 - Sequence
 - Collaboration



Διαγράμματα Sequence

- Χρήσιμα σε πρώιμες φάσεις του σχεδιασμού
- Είναι εύκολα αναγνώσιμα και από μη εξειδικευμένα άτομα
- Πολλές φορές χρησιμοποιούνται για να αποτυπώσουν περιπτώσεις χρήσης στο επίπεδο σχεδιασμού

Παραδείγμα





Παράδειγμα (Java)

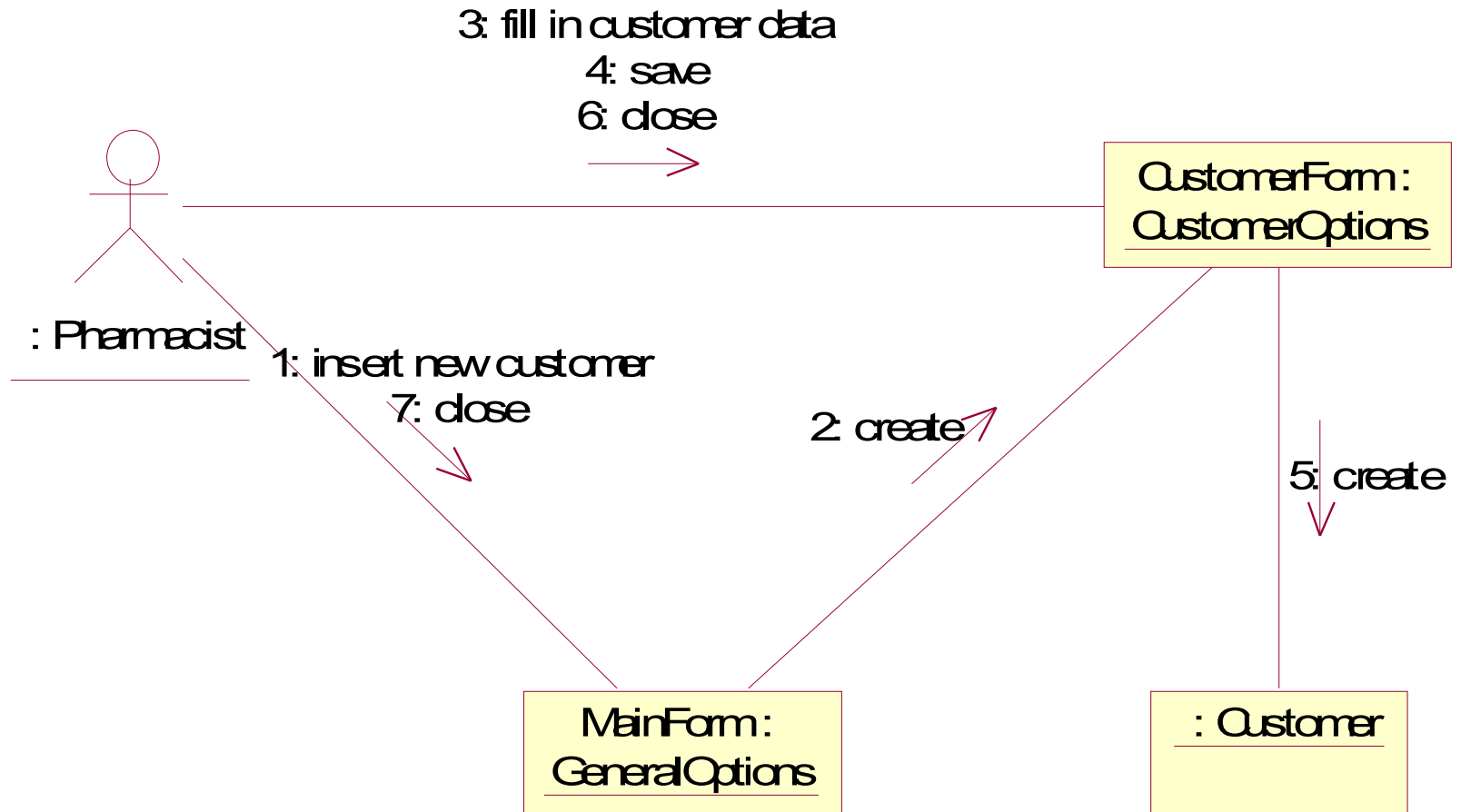
```
public class Pharmacist {  
    MainForm mf;  
    CustomerForm cf;  
  
    public void start() {  
        cf = mf.insert_new_customer();  
        cf.fill_in_customer_data();  
        cf.save();  
        cf.close();  
        mf.close();  
    }  
}  
  
public class MainForm {  
    public CustomerForm insert_new_customer() {  
        return (new CustomerForm());  
    }  
}  
  
public class CustomerForm {  
    public void save() { (new Customer()); [...] }  
}
```




Διαγράμματα Collaboration

- Αποτυπώνουν την σχέση μεταξύ των αντικειμένων
- Τα μηνύματα απεικονίζονται ως βέλη τα οποία είναι συνημένα στα βέλη
 - Κάθε μήνυμα έχει ένα αριθμό που καθορίζει την σειρά εκτέλεσης

Παράδειγμα

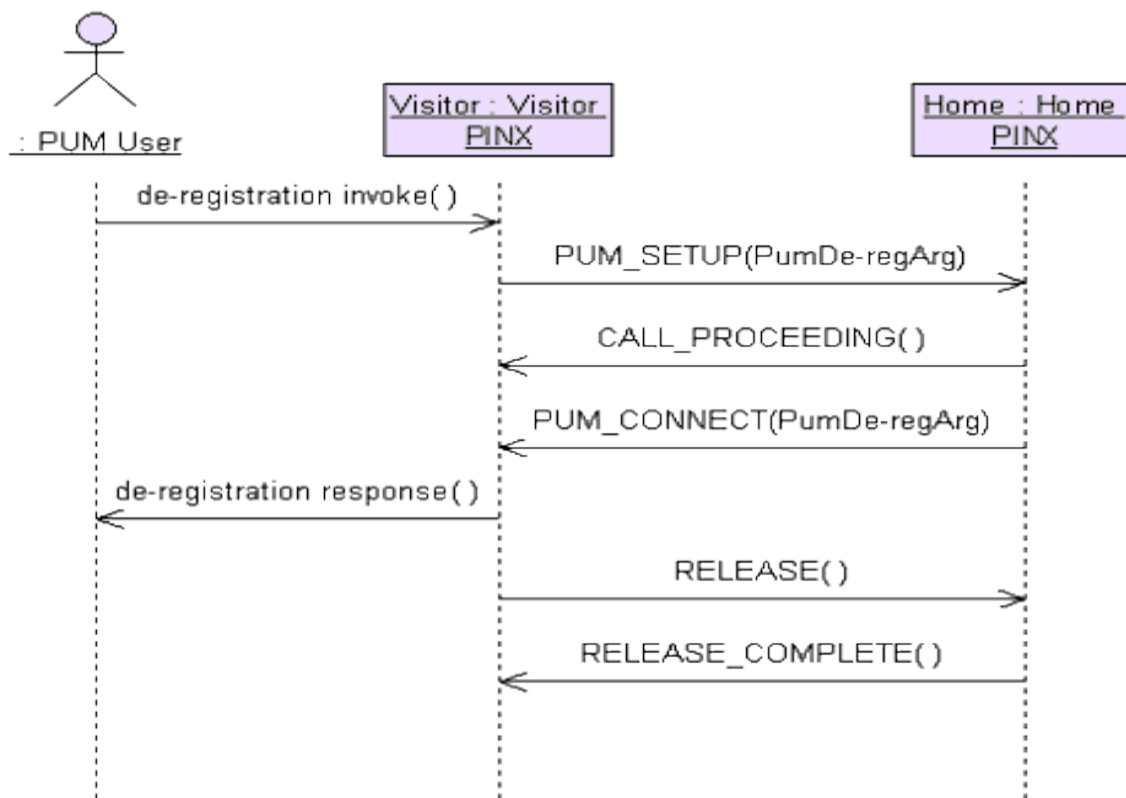




Άσκηση

```
public class Class1 {  
    Class2 class2;  
    Class3 class3;  
  
    public void class1Start() {  
        class2 = new Class2();  
        class3 = class2.operation1();  
        class3.operation2();  
        class3.operation3();  
        class3.operation4();  
        class2.operation1();  
    }  
}
```

Άσκηση



Άσκηση

