11 Σχεδίαση Εφαρμογών

Ανάλυση απαιτήσεων και σχεδίαση κλάσεων

Η Μέθοδος Ρήμα/Ουσιαστικό

- **Τεχνολογία Λογισμικού** (Software Engineering): Το αρχικό βήμα στην ανάπτυξη λογισμικού είναι η **ανάλυση απαιτήσεων**, δηλ. να καταλάβουμε και να διατυπώσουμε τι απαιτείται να κάνει το πρόγραμμα.
- Τα ουσιαστικά στην περιγραφή τω απαιτήσεων αφορούν σε 'οντότητες'.
  - Από εδώ προκύπτουν κλάσεις και στιγμιότυπα.
- Τα ρήματα αφορούν σε ενέργειες.
  - Πηγή για αλληλεπιδράσεις μεταξύ αντικειμένων
  - Υποδηλώνουν συμπεριφορά και έτσι προκύπτουν οι μέθοδοι.

Μελέτη Περίπτωσης: Κρατήσεις Θέσεων

1. Το σύστημα κρατήσεων αποθηκεύει κρατήσεις θέσεων για κινηματογραφικές αίθουσες.
2. Κάθε αίθουσα έχει καθίσματα οργανωμένα σε σειρές.
3. Οι πελάτες μπορούν να κάνουν κράτηση θέσης και τους δίδεται θέση με αριθμό σειράς και καθίσματος. Μπορούν να ζητήσουν κράτηση πολλών συνεχόμενων καθισμάτων.
4. Κάθε κράτηση γίνεται για συγκεκριμένη προβολή ταινίας (σε ορισμένη ώρα).
5. Οι προβολές γίνονται συγκεκριμένη ημερομηνία και ώρα σε συγκεκριμένη αίθουσα.
6. Το σύστημα αποθηκεύει τον αριθμό τηλεφώνου του πελάτη.

Ουσιαστικά και Ρήματα

Σύστημα κρατήσεων
αποθηκεύει (κρατήσεις θέσεων)
αποθηκεύει (τηλέφωνο)

Αίθουσα
έχει (καθίσματα)

Πελάτης
κάνει κράτηση (καθίσματα)
του δίδεται (αρ. σειράς, αρ. καθίσματος)
ζητάει (κράτηση θέσης)

Προβολή ταινίας
προγραμματίζεται (στην αίθουσα, ημ/νία, ώρα)

Ταινία

* Τα ονόματα των κλάσεων να είναι στον ενικό.

Κάρτες CRC

- Για κάθε οντότητα (ουσιαστικό) φτιάχνουμε **κάρτα CRC** (Class Responsibility Collaborator).
- Χρησιμοποιήστε πραγματικές, χάρτινες κάρτες

Όνομα κλάσης	Συνεργάτες
Ευθύνες	

Παρουσιάστηκε στο:
Beck, K., Cunnigham, W., A Laboratory for Teaching O-O Thinking, OOPSLA 89

Παράδειγμα Καρτών CRC

ΠροβολήΤαινίας	Συνεργάτες
Πελάτης	**Συνεργάτες**
ΣύστημαΚρατήσεων	**Συνεργάτες**
Βρίσκει προβολές με βάση τίτλο και ημέρα. Αποθηκεύει συλλογή προβολών. Ανακαλεί και προβάλει λεπτομέρειες προβολής. …	Προβολή Συλλογή …

Ιωάννης Γαβιώτης
gaviotis@aegean.gr
http://www.syros.aegean.gr/users/gaviotis/tmp2

Σενάρια

- **Σενάριο** είναι η περιγραφή μιας δραστηριότητας που πρέπει να διεξάγει ή να υποστηρίζει το σύστημα.
  - Αναφέρεται και ως **περιστατικά χρήσης** (use case).
- Χρησιμοποιούνται για ν ανακαλύψουμε και να καταγράψουμε τις αλληλεπιδράσεις αντικειμένων (συνεργάτες) και τις ευθύνες κάθε κλάσης.
- Ένας άνθρωπος αφηγείται το σενάριο και άλλοι συμπληρώνουν τις κάρτες CRC.

Ανάλυση Απαιτήσεων με Σενάρια

- Με τα σενάρια ελέγχουμε ότι η περιγραφή του προβλήματος είναι σαφής και πλήρης.
- Τα σενάρια πρέπει να καλύπτουν όσο το δυνατόν περισσότερες περιπτώσεις χρήσης του συστήματος.
- Η ανάλυση απαιτήσεων οδηγεί στη σχεδίαση.
  - Λάθη ή παραλείψεις που εντοπίζονται τώρα, διορθώνονται εύκολα και κοστίζουν λιγότερο απ' ότι αν τα ανακαλύπταμε αργότερα.

Παραδείγματα Σεναρίων

Σ1: Πελάτης καλεί και θέλει να κάνει κράτηση για δύο άτομα σήμερα για την ταινία «Οι απίθανοι». Το σύστημα αναζητεί ώρες προβολών της ταινίας στις οποίες υπάρχουν διαθέσιμες θέσεις. Ο πελάτης επιλέγει την ώρα προβολής, και δίνει το όνομά του. Γίνεται η κράτηση δύο συνεχόμενων θέσεων και το σύστημα καταγράφει το τηλέφωνο του πελάτη και του ανακοινώνει την αίθουσα που θα προβληθεί η ταινία.

Σ2: Υπάλληλος προγραμματίζει την προβολή τρεις φορές την ημέρα της ταινίας «Καρχαριομάχος» σε κατάλληλες αίθουσες.

Κι Άλλα Σενάρια

Σ3: Πελάτης καλεί και λέει ότι έχει κάνει κράτηση για χτες, αλλά ξέχασε τις ώρες και τον αριθμό του καθίσματος.

Σ4: Μια προβολή αναβάλλεται, γιατί χάλασε το κλιματιστικό. Όλοι οι πελάτες που έχουν κάνει κράτηση πρέπει να ειδοποιηθούν.

Σ5: Πελάτης που έχει κάνει κράτηση για 3 άτομα, θέλει να την επεκτείνει για 5 άτομα.

Σχεδίαση Κλάσεων

- Διατρέχοντας τα σενάρια, έχουμε συμπληρώσει τις κάρτες CRC με τις ευθύνες και τους συνεργάτες κάθε οντότητας.
- Πετάμε τις κάρτες που δεν έχουμε γράψει τίποτα.
- Από τις ευθύνες προκύπτουν οι δημόσιες μέθοδοι της κλάσης,
  - και μερικές φορές πεδία, πχ «αποθηκεύει τηλέφωνο»

Δημόσια Διεπαφή Κλάσεων

- Ξανατρέχουμε τα σενάρια για να προσδιορίσουμε για κάθε μέθοδο τις παραμέτρους της και το επιστρεφόμενο αποτέλεσμα.
- Κατασκευάζουμε τον σκελετό των κλάσεων
  - Όνομα κλάσης και κληρονομιά από άλλες κλάσεις
  - Υπογραφές μεθόδων – χωρίς σώμα
- Εμπειρικός κανόνας: αφιέρωσε (τουλάχιστον ☺)
  - 50% του χρόνου στην ανάλυση / σχεδίαση
  - 50% του χρόνου στην υλοποίηση / έλεγχο

Ιωάννης Γαβιώτης
gaviotis@aegean.gr
http://www.syros.aegean.gr/users/gaviotis/tmp2

Σχεδίαση Διεπαφής Χρήστη

- Πότε θα γίνει η διεπαφή χρήστη
  - Τι θα βλέπουν οι χρήστες στην οθόνη;
  - Πώς θα αλληλεπιδρούν με το σύστημα;
- Σε μια καλοσχεδιασμένη εφαρμογή, η διεπαφή χρήστη είναι ασθενώς συνδεδεμένη με τις κλάσεις που υποστηρίζουν τη λειτουργικότητα της εφαρμογής.
- Άρα η σχεδίαση της διεπαφής μπορεί να γίνει αυτόνομα από τη σχεδίαση / υλοποίηση της βασικής λειτουργικότητας.
  - Ίσως μάλιστα και από διαφορετικές ομάδες ανάπτυξης.

Τεκμηρίωση

- Γράψε σχόλια για κάθε κλάση.
- Γράψε σχόλια για τις μεθόδους.
- Να περιγράφεις το σκοπό καθεμιάς.
- Να επικεντρώνεις στο ***τι*** κάνει και όχι στο ***πώς*** το κάνει.
- Γι' αυτό η τεκμηρίωση πρέπει να γράφεται πριν την συγγραφή του πηγαίου κώδικα.
- Η αναγκαιότητα της τεκμηρίωσης γίνεται προφανής σε πραγματικά έργα ανάπτυξης λογισμικού με ομάδες ανθρώπων και πολλές κλάσεις.
  - Δεν είναι ορατή στις ασκήσεις του μαθήματος.

Ακραίος Προγραμματισμός

- Πρόσφατα προτείνεται η συγγραφή του κώδικα να γίνεται από δύο προγραμματιστές (extreme programming).
  - Ο έλεγχος δεν γίνεται μετά τη συγγραφή του κώδικα.
  - Έχει βρεθεί ότι οδηγεί σε καλύτερο, πιο ποιοτικό κώδικα.
- Η ανάπτυξη σε ομάδες είναι ο κανόνας και όχι η εξαίρεση.
- Η αντικειμενοστρεφής σχεδίαση προάγει την ομαδική ανάπτυξη, επειδή παράγει ασθενώς συνδεδεμένα τμήματα κώδικα.

Πρωτοτυποποίηση

- Υποστηρίζει την 'εξερεύνηση' του συστήματος που πρόκειται να αναπτυχθεί.
  - Έτσι ανιχνεύονται νωρίς τα προβλήματα.
- Χωρίς πλήρη λειτουργικότητα
  - Συγκεκριμένες «δύσκολες» λειτουργίες απλώς προσομοιώνονται – δεν υλοποιούνται.
  - Πχ επιστρέφουν ένα (λογικό) αποτέλεσμα που είναι πάντα το ίδιο.
- Αντλούνται συμπεράσματα και μετά το πρωτότυπο μπορεί να πεταχτεί (throw-away prototype).

Μοντέλο Καταρράκτη

- Το μοντέλο καταρράκτη (waterfall model) για την ανάπτυξη λογισμικού έχει τις εξής φάσεις:
  - Ανάλυση
  - Σχεδίαση
  - Υλοποίηση
  - Έλεγχος
  - Εγκατάσταση
  - Συντήρηση
- Είναι ο παραδοσιακός και συντηρητικός τρόπος ανάπτυξης.
  - Είναι άκαμπτος, με την έννοια ότι υποθέτει ότι κάθε φάση ολοκληρώνεται καλώς και τότε προχωρούμε στην επόμενη.

Ελικοειδής Ανάπτυξη

- Το λογισμικό δεν σχεδιάζεται και υλοποιείται άπαξ., αντίθετα αναπτύσσεται ως οργανισμός.
  - Οι πρώτες εκδόσεις του έχουν λάθη / μειονεκτήματα.
  - Προκύπτουν νεότερες εκδόσεις που αντιμετωπίσουν τα ζητήματα, κοκ.
- Οι φάσεις:
  - Ανάλυση
  - Σχεδίαση
  - Πρωτοτυποποίηση
  - Μελέτη αποτελεσμάτων
  - φτου και απ' την αρχή ...

Πρότυπα Σχεδίασης

- Οι σχέσεις μεταξύ των κλάσεων είναι σημαντικές και μπορεί να είναι πολύπλοκες.
- Μερικές σχέσεις επαναλαμβάνονται σε διαφορετικές εφαρμογές
- Αξίζει να 'αναγνωρίσουμε' πρότυπα τέτοιων σχέσεων για να προσπαθήσουμε να επαναχρησιμοποιούμε λύσεις όποτε αυτά εμφανίζονται (design patterns).
  - Όπως έχουμε πρότυπα κώδικα, πχ κώδικας που βρίσκει τον μέγιστο, που ταξινομεί, κλπ.

Σύσταση Προτύπου Σχεδίασης

- Όνομα
- Περιγραφή του προβλήματος που αντιμετωπίζει
- Προσέγγιση που προτείνει για τη λύση του προβλήματος
  - Δομές, συνεργάτες
- Το αποτέλεσμα που δίνει
  - Κύριο αποτέλεσμα, παρενέργειες, συμβιβασμοί

Διακοσμητής

- Το πρότυπο σχεδίασης 'Διακοσμητής' (Decorator) εμφανίζεται όταν έχουμε να προσθέσουμε ή να τροποποιήσουμε λειτουργικότητα σε ένα υπάρχον αντικείμενο.
- Το 'διακοσμημένο' αντικείμενο ενθυλακώνει ένα άλλο αντικείμενο.
  - Έχει παρόμοια δημόσια διεπαφή
  - Κλήσεις μεθόδων μπορεί να παραπέμπονται στο ενθυλακωμένο αντικείμενο,
  - και άλλες να υλοποιούνται εξαρχής.
- Μια μέθοδος υλοποίησης είναι με το μηχανισμό κληρονομικότητας.

Σόλο

- Το πρότυπο σχεδίασης 'Σόλο' (singleton) εξασφαλίζει ότι υπάρχει ένα και μοναδικό στιγμιότυπο μιας κλάσης.
- Στη Java, μια κλάση μπορεί να επιβληθεί ως 'σόλο', δηλώνοντας τον κατασκευαστή της **`private`** και φτιάχνοντας μια δημόσια μέθοδο επιπέδου κλάσης (**`static`**), πχ `getInstance` που τον καλεί.

```
public class Auction {
  //φτιάχνει ένα μοναδικό στιγμιότυπο
  private static Auction instance = new Auction();
  //επιστρέφει το μοναδικό στιγμιότυπο
  public static Auction getInstance() { return instance; }
  //κατασκευαστής δεν μπορεί να κληθεί από άλλη κλάση
  private Auction () { ... }

}
```


Όροι Ενότητας 11

- Ανάλυση απαιτήσεων
- Μέθοδος ρήμα/ουσιαστικό
- Κάρτες CRC
- Σενάρια
- Ακραίος προγραμματισμός
- Πρωτοτυποποίηση
- Μοντέλο καταρράκτη
- Ελικοειδής ανάπτυξη
- Πρότυπο σχεδίασης
- Διακοσμητής
- Σόλο

Ιωάννης Γαβιώτης
gaviotis@aegean.gr
http://www.syros.aegean.gr/users/gaviotis/tmp2