



ΔΙΔΑΣΚΩΝ
Αντώνιος Σαββίδης

ΕΝΟΤΗΤΑ 5

ΣΧΕΔΙΑΣΤΙΚΑ ΠΡΟΤΥΠΑ

Αριθμός διαλέξεων 5 – Διάλεξη 1η



Περιεχόμενα

- *Επαναχρησιμοποίηση σχεδίασης*
- Επαναλαμβανόμενα πρότυπα λύσεων
- Τεκμηρίωση προτύπων
- Γένεση προτύπων
- Κατηγορίες προτύπων

Επαναχρησιμοποίηση σχεδίασης (1/2)

- Η δυνατότητα τυποποίησης και τεκμηρίωσης μίας μεθόδου για τη λύση ενός συγκεκριμένου προβλήματος, με τρόπο που επιτρέπει την επιτυχή εφαρμογή της σε παρόμοιες κλάσεις προβλημάτων.
 - Στην τεχνολογία λογισμικού, περισσότερο σημαντικό από την υλοποίηση ενός αλγορίθμου για μία συγκεκριμένη «εργασία» (coding), είναι η αναζήτηση και οριοθέτηση της μεθόδου που επιλύει το πρόβλημα στη γενικότητά του.

Επαναχρησιμοποίηση σχεδίασης (2/2)

- Για να δοθεί υπόσταση και μορφή στη μέθοδο η οποία υφίσταται επαναχρησιμοποίησης, θα πρέπει να εντοπιστούν δομικές, συμπεριφεριολογικές, και λειτουργικές ομοιότητες σε παρόμοιες, αλλά διαφορετικές περιπτώσεις προβλημάτων, για την συνεχή τυποποίηση της επαναλαμβανόμενης σχεδιαστικής λογικής.
- ➔ "ο εντοπισμός ομοιοτήτων, που απαιτεί την ανακάλυψη κοινών συμπεριφεριολογικών σχημάτων μεταξύ των εξεταζόμενων υποκειμένων, θεωρείται ως θεμελιωδώς **νοητική** διεργασία, ενώ η φάση εντοπισμού διαφορών, που απαιτεί την αντιπαράθεση των αντίστοιχων φυσικών προοπτικών των υποκειμένων, χαρακτηρίζεται ως **μηχανιστική** διεργασία", μετάφραση από «*Unified Interface Development...*», A. Savidis, 1998.

Περιεχόμενα

- Επαναχρησιμοποίηση σχεδίασης
- **Επαναλαμβανόμενα πρότυπα λύσεων**
- Τεκμηρίωση προτύπων
- Γένεση προτύπων
- Κατηγορίες προτύπων

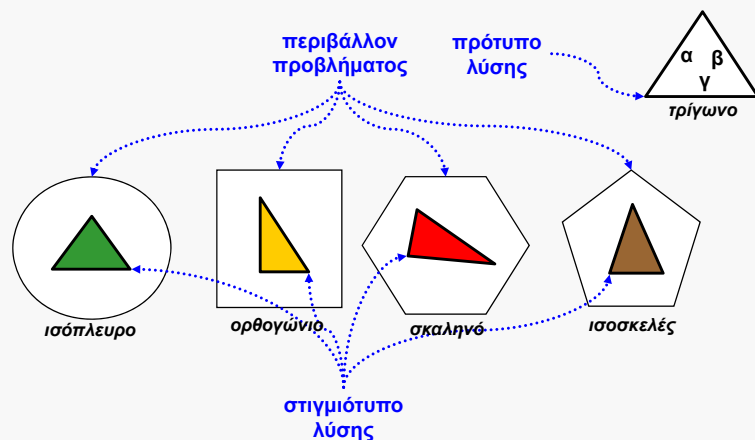
Επαναλαμβανόμενα πρότυπα λύσεων (1/3)

- "Κάθε πρότυπο περιγράφει ένα πρόβλημα το οποίο λαμβάνει χώρα συνεχώς στο περιβάλλον μας, και έπειτα περιγράφει την ουσία της λύσης του προβλήματος, με τέτοιο τρόπο ώστε να μπορείς να χρησιμοποιήσεις τη λύση αμέτρητες φορές, χωρίς ωστόσο να κάνεις το ίδιο πράγμα δύο φορές", μετάφραση από Alexander et al, 1977, «*A Pattern Language*».

Επαναλαμβανόμενα πρότυπα λύσεων (2/3)

- Ένα πρότυπο λύσης μπορεί να είναι κάτι από τα παρακάτω, χωρίς να περιορίζεται μόνο σε αυτά:
 - Οδηγίες αντιμετώπισης ενός προγραμματιστικού προβλήματος
 - Σκιαγράφηση του τρόπου οργάνωσης των κλάσεων και του σχεδιαστικού σχήματος των μελών για κάποιο συγκεκριμένο σκοπό
 - Κανόνες συνδεσμολογίας μεταξύ προγραμματιστικών τμημάτων
 - Αρχιτεκτονικό διάγραμμα για μία συγκεκριμένη κατηγορία συστημάτων
 - Ένα αντιπροσωπευτικό λειτουργικό σενάριο (use case) των διαφόρων τμημάτων ενός συστήματος στόχου
 - Τρόποι ονοματολογίας, διαχωρισμού σε αρχεία, και κανόνες κλήσεων για τον προγραμματισμό υποτιμήματος
 - Μία συγκεκριμένη σχεδιαστική προοπτική της επαναλαμβανόμενης δομής

Επαναλαμβανόμενα πρότυπα λύσεων (3/3)



Περιεχόμενα

- Επαναχρησιμοποίηση σχεδίασης
- Επαναλαμβανόμενα πρότυπα λύσεων
- *Τεκμηρίωση προτύπων*
- Γένεση προτύπων
- Κατηγορίες προτύπων

Τεκμηρίωση προτύπων (1/6)

- Όταν τα πρότυπα λύσεων τυποποιούνται και παρέχουν όλη την απαραίτητη πληροφορία, και όχι περισσότερη, για την εφαρμογή του προτύπου σε πλήρη ισχύ, τότε γίνονται τεκμηριωμένα σχεδιαστικά πρότυπα.
- Για την τεκμηρίωση χρειαζόμαστε γενικά:
 - όνομα
 - πρόβλημα
 - λύση
 - επιπτώσεις

Τεκμηρίωση προτύπων (2/6)

- **Όνομα ή φράση**
 - Σύντομο, οικείο, περιγραφικό, μοναδικό, ενδεικτικό της λύσης (παρά του προβλήματος), ανεξάρτητο υλοποίησης. Π.χ.
 - ◆ *Decoder* (τύπος constructor)
 - ◆ *Replicator* (τύπος μέλους για παραγωγή αντιγράφων)

Τεκμηρίωση προτύπων (3/6)

■ Πρόβλημα

- Τι λύνει το πρότυπο
- Εξηγεί τον προγραμματιστικό χώρο στον οποίο μπορεί να εφαρμοστεί
- Παρέχει σύνολο κριτηρίων που πρέπει να είναι κομμάτι προβλήματος
- Μπορεί να εισάγει προγραμματιστικές απαιτήσεις, περιορισμούς ή συνθήκες

Τεκμηρίωση προτύπων (4/6)

■ Λύση (1/2)

- Περιγράφει τις οντότητες που συνιστούν την επαναχρησιμοποιούμενη σχεδιαστική λογική, αποκαλύπτοντας σχέσεις, συνδέσεις, ρόλους, συμπεριφορές, έλεγχο και λειτουργικότητα.
- Δεν παρέχει χαμηλού επιπέδου πληροφορίες υλοποίησης, αλλά κυρίως προσφέρεται ως *καλούπι* - template

Τεκμηρίωση προτύπων (5/6)

■ Λύση (2/2)

- Μπορεί να ορίζει την καταλληλότερη κατηγορία γλώσσας στην οποία είναι πρακτικά εφαρμόσιμη (π.χ. OOP / procedural / formatting / hypertext / scripting / configuration / logic / formal). Σε αυτό το πλαίσιο, μπορεί να περιλαμβάνει περιγραφή συγκεκριμένων προγραμματιστικών δομών.
 - ♦ Μερικά πρότυπα μπορεί να παρέχουν λύσεις για συγκεκριμένη κατηγορία γλωσσών, ενώ μπορεί οι ίδιες λύσεις να είναι ακατάλληλες ή και τετριμμένες για άλλες γλώσσες.

→ Τα σχεδιαστικά πρότυπα δεν είναι εφαρμόσιμα μόνο στον οντοκεντρικό προγραμματισμό

Τεκμηρίωση προτύπων (6/6)

■ Επιπτώσεις

- Περιγράφει ποια αναμένεται να είναι η κατάσταση μετά την εφαρμογή του προτύπου
 - ♦ Αυτή η πληροφορία μπορεί να επηρεάσει τις σχεδιαστικές αποφάσεις μέσα στο προγραμματιστικό τμήμα στο οποίο το εφαρμοζόμενο πρότυπο θα ανήκει
- Περιγράφει όσο το δυνατόν πιο συγκεκριμένα τις εσωτερικές συνέπειες από την επιλογή του συγκεκριμένου προτύπου

Περιεχόμενα

- Επαναχρησιμοποίηση σχεδίασης
- Επαναλαμβανόμενα πρότυπα λύσεων
- Τεκμηρίωση προτύπων
- *Γένεση προτύπων*
- Κατηγορίες προτύπων

Γένεση προτύπων (1/4)

- Η κατασκευή ενός σχεδιαστικού προτύπου είναι τέχνη, επιστήμη, παιχνίδι και πονοκέφαλος
- Δεν υπάρχει συστηματική μέθοδος ελέγχου του κατά πόσο το πρότυπο είναι έτοιμο «να βγει από το φούρνο»
- Πρακτικά, ότι ξέρουμε είναι ότι ένα πρότυπο συνιστά:
 - μία κοινή λύση σε ένα επαναλαμβανόμενο πρόβλημα, ή
 - μία επαναλαμβανόμενη λύση σε ένα κοινό πρόβλημα
- Σε αντίθεση με τις διάφορες σχεδιαστικές προοπτικές, δεν υπάρχει καλά ορισμένη και γενικά αποδεκτή τυποποίηση των μεθόδων σχεδίασης προτύπων.

Γένεση προτύπων (2/4)

- ➔ Αλλά υπάρχουν οδηγίες, συνήθειες, συστάσεις, και κάποιες γενικές μέθοδοι
- Επτά καλές συνήθειες για για την γένεση προτύπων, μετάφραση από J. Vlissides, «*Pattern hatching: seven habits of successful pattern writers*» C++ Report, 1996.

Γένεση προτύπων (3/4)

- I. Αφιέρωση χρόνου σε αντανάκλαση (εφαρμογή)
- II. Αντικατοπτρισμός μίας οργανωμένης δομής
- III. Συγκεκριμενοποίηση πολύ νωρίς
- IV. Διαχωρισμός και αλληλοσυμπλήρωση προτύπων
- V. Αποτελεσματική τεκμηρίωση
- VI. Ακούραστη και συνεχής αναθεώρηση
- VII. Συλλογή και ενσωμάτωση εμπειρίας χρήσης

Γένεση προτύπων (4/4)

■ Επιπλέον οδηγίες

- Αξιολόγησε συνεχώς το πρότυπο σε εναλλακτικές περιπτώσεις προβλημάτων
- Σταδιακά αναθεώρησε και εκσυγχρόνισε το πρότυπο
- Σύγκρινε με παρόμοια πρότυπα
- Επέλεξε την καταλληλότερη αναπαράσταση, περιγραφή και τεκμηρίωση
- Εντόπισε και τεκμηρίωσε όλα τα δομικά στοιχεία
- Εισήγαγε την μικρότερη απαραίτητη ποσότητα πληροφορίας για την πλήρη εφαρμογή του προτύπου

Περιεχόμενα

- Επαναχρησιμοποίηση σχεδίασης
- Επαναλαμβανόμενα πρότυπα λύσεων
- Τεκμηρίωση προτύπων
- Γένεση προτύπων
- *Κατηγορίες προτύπων*

Κατηγορίες προτύπων (1/5)

- Ενώ υπάρχει κάποια κατηγοριοποίηση στο βιβλίο «Design Patterns», δεν είναι πάντα ξεκάθαρη η αντιστοίχιση ορισμένων προτύπων σε συγκεκριμένη κατηγορία
- Τέτοιου είδους σχήματα ταξινόμησης θα πρέπει να θεωρούνται ως ανοικτά:
 - κάθε νέο πρότυπο δεν πρέπει απαραίτητως να ανήκει σε μία υπάρχουσα κατηγορία,
 - αλλά ούτε και θα πρέπει να ορίζει από μόνο του μία νέα κατηγορία

Κατηγορίες προτύπων (2/5)

- Μερικές ενδεικτικές κατηγορίες (1/2):
 - *Constructional, creational*
 - ◆ Τρόποι παραγωγής στιγμιότυπων
 - *Communicational*
 - ◆ Τρόποι επικοινωνίας μεταξύ τμημάτων
 - *Control, coordination*
 - ◆ Τρόποι διαχείρισης και διεύθυνσης άλλων τμημάτων
 - *Compositional*
 - ◆ Τρόποι σύνθεσης τμημάτων για σύνθετες λειτουργίες
 - *Search*
 - ◆ Τρόποι αναζήτησης

Κατηγορίες προτύπων (3/5)

■ Μερικές ενδεικτικές κατηγορίες (2/2):

- **Browsing**
 - ◆ Τρόποι διερεύνησης και επίσκεψης πολύπλοκων δομών
- **Memory**
 - ◆ Έξυπνοι τρόποι διαχείρισης της μνήμης
- **Correctness**
 - ◆ Τρόποι πιστοποίησης και ελέγχου λειτουργικής ορθότητας
- **Optimizers**
 - ◆ Τρόποι βελτιστοποίησης απόδοσης
- **Sharing**
 - ◆ Ειδικοί τρόποι κοινής χρήσης εσωτερικών δεδομένων
- **Transformers**
 - ◆ Τρόποι μετάλλαςξης τμημάτων χωρίς να επηρεάζονται τα αυθεντικά

Κατηγορίες προτύπων (4/5)

■ Ποια θα μελετήσουμε (1/2)

- **Browsing**
 - ◆ Iterator
- **Constructional**
 - ◆ Factory
 - ◆ Prototype
- **Sharing**
 - ◆ Singleton
 - ◆ State
- **Control / coordination / communication**
 - ◆ Proxy
 - ◆ Dispatch table
 - ◆ Black board

Κατηγορίες προτύπων (5/5)

■ Ποια θα μελετήσουμε (2/2)

- ◆ Listener
- ◆ Progress monitoring
- ◆ Command
- ◆ Undo / redo
- **Transformers**
 - ◆ Adapter
 - ◆ View
- **Memory**
 - ◆ Flyweight
- **Compositional**
 - ◆ Decorator