

Πρότυπα Λογισμικού (Software Patterns)

Παρουσίαση στα πλαίσια
του μαθήματος Τεχνολογία
Λογισμικού του Μ.Π.Σ.
«Πληροφοριακά
Συστήματα» του Ο.Π.Α.

Επιμέλεια: Οικονομίδης Δημήτρης

Τι Είναι Πρότυπο Γενικά (1)

- Ένας προκατασκευασμένος τύπος αντικειμένου που χρησιμεύει ως υπόδειγμα για την κατασκευή άλλων ομοίων αντικειμένων.
Συνώνυμα: μοντέλο, μήτρα, καλούπι.
- Μια αφαίρεση από ένα διακεκριμένο τύπο γεγονότων, καταστάσεων κτλ., τα οποία επαναλαμβάνονται σε ένα συγκεκριμένο, μη τυχαίο περιβάλλον.
- Μια επαναλαμβανόμενη λύση για ένα κοινό πρόβλημα σε ένα δοσμένο περιβάλλον (context) και σύστημα δυνάμεων (forces) που δρουν μέσα σε αυτό [Alexander].

Τι Είναι Πρότυπο Γενικά (2)

Οι λέξεις κλειδιά στον ορισμό του Alexander είναι περιβάλλον (ή αλλιώς πλαίσιο), πρόβλημα και λύση.

- Πλαίσιο : αναφέρεται σε ένα επαναλαμβανόμενο σύνολο συνθηκών στις οποίες μπορεί να εφαρμοστεί το πρότυπο.
- Πρόβλημα : αναφέρεται σε ένα σύνολο δυνάμεων (forces) (στόχοι και περιορισμοί) που αναπτύσσονται μέσα στο πλαίσιο εργασίας.
- Λύση : αναφέρεται σε ένα κανόνα ή φόρμα σχεδίασης (Rule or Design Form) που κάποιος μπορεί να εφαρμόσει για να επιλύσει το πρόβλημα.

Τι Είναι Πρότυπο Λογισμικού

- Ένας κανόνας αποτελούμενος από τρία μέρη, ο οποίος εκφράζει μια σχέση μεταξύ ενός συγκεκριμένου περιβάλλοντος (context), ενός συγκεκριμένου συστήματος δυνάμεων (παραγόντων ενός προβλήματος) που παρουσιάζονται επανειλημμένα σε αυτό το περιβάλλον, και μια συγκεκριμένη διάταξη λογισμικού που επιτρέπει σε αυτές τις δυνάμεις να επιλυθούν π.χ. τρόπος αναπαράστασης πολύπλοκων δομών δεδομένων στην μνήμη.
- Μια επιτυχημένη επαναλαμβανόμενη 'βέλτιστη πρακτική' που έχει αποδείξει την αξία της στην αληθινή καθημερινή χρήση.
- Μία έγγραφη, τυποποιημένη τεχνική για την σύλληψη της γνώσης και της εμπειρίας των ειδικών σχεδιαστών και την επικοινωνία τους στους αρχάριους.

Τι Δεν Είναι

- Η εφαρμογή τους δεν περιορίζεται στην αντικειμενοστραφή σχεδίαση, ή την σχεδίαση λογισμικού, εν γένει
- Δεν είναι μη δοκιμασμένες ιδέες / θεωρίες ή νέες εφευρέσεις
- Δεν είναι λύσεις που έχουν λειτουργήσει σωστά μόνο μία φορά
- Δεν είναι απλώς οποιαδήποτε γνώση γραμμένη σε μορφή προτύπου
- Δεν είναι αφηρημένες αρχές (principles) ή ευρετικές μέθοδοι (heuristics)
- Δεν είναι εφαρμόσιμα σε όλα τα υπάρχοντα περιβάλλοντα
- Δεν είναι πανάκεια για τα προβλήματα της τεχνολογίας λογισμικού ("silver bullet"!!!)

Ιστορία

- Literate προγραμματισμός (ενσωμάτωση τεκμηρίωσης στον κώδικα).
- Τεκμηρίωση βέλτιστων πρακτικών και εγχειρίδια μηχανικών και αρχιτεκτόνων.
- Τέλη '70: Τεκμηρίωση προτύπων για τον σχεδιασμό πόλεων και την αρχιτεκτονική κτιρίων από τον αρχιτέκτονα Christopher Alexander.
- 1987: Kent Beck και Ward Cunningham: Σχεδιασμός GUI με χρήση προτύπων στην Smalltalk,
- 1995 Gang Of Four Design Patterns: Elements of Reusable Object-Oriented Software καλύτερο Object-Oriented βιβλίο όλων των εποχών.
- 1995 1ο PloP συνέδριο "Pattern Languages of Program Design".
- 1998 1ο Chili PloP συνέδριο (για την εκμάθηση του τρόπου συγγραφής προτύπων και γλωσσών προτύπων).

Βασικά Μέρη Ενός Προτύπου Λογισμικού (1)

- **Όνομα :**
 - μοναδικό, περιγραφικό του προβλήματος
- **Πρόβλημα :**
 - Δήλωση του προβλήματος / σκοπός (intent) της περιγραφόμενης λύσης
- **Περιβάλλον :**
 - πότε εφαρμόζεται η λύση που περιγράφεται
 - Αρχική διαμόρφωση (configuration) του συστήματος πριν την εφαρμογή του προτύπου (preconditions)
- **Δυνάμεις (Forces) :**
 - trade-offs στόχοι & περιορισμοί

Βασικά Μέρη Ενός Προτύπου Λογισμικού (2)

■ **Λύση :**

- Αναφέρεται στον τρόπο παραγωγής της λύσης του προβλήματος
- Περιλαμβάνει τη δομή της λύσης, τα συμμετέχοντα μέρη και τις ενδεχόμενες συνεργασίες μεταξύ τους

■ **Παραδείγματα** χρήσης του προτύπου (προαιρετικό)

■ **Προκύπτων περιβάλλον :**

- Περιγράφει το τελικό αποτέλεσμα, τα οφέλη και τις συνέπειες (τόσο καλές όσο και κακές) από την εφαρμογή του προτύπου

Βασικά Μέρη Ενός Προτύπου Λογισμικού (3)

- **Λογική (Rationale)** (προαιρετική) :
 - Επεξηγεί γιατί η λύση πραγματικά παράγει το αναμενόμενο αποτέλεσμα.
 - Παρέχει μια οπτική των βαθύτερων δομών και των κύριων μηχανισμών.
- **Κατευθυντήριες αρχές** (guidelines) υλοποίησης του προτύπου και δείγματα κώδικα υλοποίησης.
- **Σχετικά Πρότυπα.**
- **Γνωστές Χρήσεις.**
 - Κανόνας των 3.

Οδηγίες Χρήσης Προτύπου

1. IF βρεθείς σε κάποιο CONTEXT
2. για παράδειγμα EXAMPLES,
3. με κάποιο PROBLEM,
4. συνεπαγόμενο την ύπαρξη ορισμένων FORCES,
5. THEN για κάποιους REASONS (Rationale),
6. εφάρμοσε DESIGN FORM AND/OR RULE
7. για να κατασκευάσεις SOLUTION
8. που οδηγεί σε ένα NEW CONTEXT και OTHER PATTERNS.

Ανάλυση Των Δυνάμεων

Ο όρος των δυνάμεων γενικεύει τα είδη των κριτηρίων που χρησιμοποιούν οι μηχανικοί λογισμικού για να δικαιολογήσουν σχεδιαστικές αποφάσεις και υλοποιήσεις. Στην κλασική θεωρία π.χ. αλγορίθμων ο κύριος παράγοντας που πρέπει να ικανοποιηθεί είναι η αποδοτικότητα (πολυπλοκότητα σε χρόνο). Τα πρότυπα λογισμικού όμως ασχολούνται με ένα ευρύτερο σύνολο αντικρουόμενων στόχων και περιορισμών κατά την ανάπτυξη οποιουδήποτε έργου (artifact).

Παραδείγματα Δυνάμεων (1)

■ Ορθότητα

- Πληρότητα και ορθότητα λύσης
- Ανοχή σε σφάλματα
- Ασφάλεια και ευρωστία του συστήματος

■ Χρήση πόρων

- Αποδοτικότητα : πολυπλοκότητα χρόνου, αριθμός αποσταλλέντων μηνυμάτων, απαιτήσεις σε bandwidth
- Χρησιμοποίηση χώρου: αριθμός κελιών μνήμης, αντικειμένων, διεργασιών, καναλιών επικοινωνίας

Παραδείγματα Δυνάμεων (2)

■ Δομή.

- Διάρθρωση σε ενότητες, encapsulation, ανεξαρτησία.
- Επεκτασιμότητα, συντηρησιμότητα.
- Επαναχρησιμοποίηση, μεταφερσιμότητα, ενσωμάτωση.
- Διαλειτουργικότητα.

■ Κατασκευή.

- Κατανοησιμότητα, απλότητα, ελαχιστοποίηση.
- Συνύπαρξη με άλλο λογισμικό.
- Συντηρησιμότητα.

■ Χρήση.

- Ανθρώπινος παράγοντας : εκμάθηση.
- Προσαρμοστικότητα στις μεταβαλλόμενες απαιτήσεις.

Κατηγορίες Προτύπων

- Πρότυπα Αρχιτεκτονικής (Architectural Patterns)
- Πρότυπα Σχεδίασης (Design Patterns)
- Ιδιώματα (Idioms - coding patterns)
- Πρότυπα Ελέγχου
- Πρότυπα Ανάλυσης
- Πρότυπα Οργάνωσης
- Πρότυπα Διεργασιών

Architectural Patterns

- Εκφράζουν μια θεμελιακή δομική οργάνωση ή ένα schema για ένα software σύστημα.
- Παρέχουν ένα σύνολο από προκαθορισμένα υποσυστήματα, καθορίζουν τις υπευθυνότητες τους.
- Περιλαμβάνουν κανόνες και κατευθυντήριες γραμμές (guidelines) για την οργάνωση των σχέσεων μεταξύ τους.
- High-level στρατηγικές που αφορούν large-scale components και τις global ιδιότητες και μηχανισμούς ενός συστήματος.

Design Patterns

- Περιγραφή επικοινωνούντων μεταξύ τους αντικειμένων και κλάσεων που μπορούν να παραμετροποιηθούν για την επίλυση ενός γενικού και συχνά επαναλαμβανόμενου (recurring) προβλήματος σχεδίασης λογισμικού σε ένα συγκεκριμένο context
- Παρέχουν ένα scheme για την εκλέπτυνση των υποσυστημάτων ή των components ενός software συστήματος, ή των μεταξύ τους σχέσεων
- Δείχνουν τις στατικές και δυναμικές συσχετίσεις μεταξύ αντικειμένων και κλάσεων σε ένα αντικειμενοστραφές περιβάλλον
- Medium-scale τακτικές, ορίζουν μικρό - αρχιτεκτονικές (micro-architectures) των υποσυστημάτων και των components χωρίς να επηρεάζουν την δομή του συστήματος

Design Patterns - Κατηγορίες

- Δημιουργικά Πρότυπα (Creational patterns)
 - Αρχικοποίηση και διαμόρφωση των κλάσεων και των αντικειμένων
- Δομικά Πρότυπα (Structural patterns)
 - Διάκριση της διεπαφής και της υλοποίησης των κλάσεων και των αντικειμένων
- Πρότυπα συμπεριφοράς (Behavioral patterns)
 - Δυναμικές αλληλεπιδράσεις μεταξύ κοινωνιών κλάσεων και αντικειμένων

Idioms (Coding Patterns)

- Περιγράφουν τον τρόπο υλοποίησης συγκεκριμένων μερών των components ή των σχέσεων μεταξύ τους κάνοντας χρήση των τεχνικών / στυλ της δεδομένης γλώσσας.
- Low-level πρότυπα εξειδικευμένα για μια γλώσσα προγραμματισμού.
- Π.χ. Συγκεκριμένες χρήσεις εμφωλιασμένων κλάσεων στην C++, διεπαφών (interfaces) στην Java, cascaded κλήσεων στην Smalltalk.
Ιδίωμα της C: `while(*dest++ = *src++);`

Γλώσσες Προτύπων

- Τα πρότυπα σχηματίζουν γλώσσες παρόμοιες με τις φυσικές.
- Γλώσσα προτύπων = Λεξικό + γραμματική για το σχηματισμό έγκυρων προτάσεων.
- Τα πρότυπα συνδέονται νοητικά μεταξύ τους έτσι ώστε το ένα πρότυπο οδηγεί σε ένα παρόμοιο πρότυπο κ.ο.κ. έως ότου επιλύονται κοινά προβλήματα.
- Μια γλώσσα προτύπων ορίζει μια συλλογή από πρότυπα και τους κανόνες για τον συνδυασμό μεταξύ τους σε μια αρχιτεκτονική μεθοδολογία.
- Κανόνες και guidelines που εξηγούν πώς, πότε και με ποια σειρά πρέπει να εφαρμοστούν τα πρότυπα της γλώσσας ώστε να επιλύσουν ένα πρόβλημα πολύπλοκο.

Χαρακτηριστικά Ενός Καλού Προτύπου

- Επιλύει ένα πρόβλημα
 - Τα πρότυπα αποτυπώνουν λύσεις και όχι αφηρημένες αρχές
- Αποτελεί ένα αποδεδειγμένο concept
- Η λύση δεν είναι φανερή
 - Παράγεται έμμεσα μέσω της παρατήρησης και της εμπειρίας
- Περιγράφει μια σχέση
 - βαθύτερη δομή και τους μηχανισμούς ενός συστήματος
- Περιέχει τον ανθρώπινο παράγοντα σαν συστατικό στοιχείο

Χρησιμότητα Προτύπων

- Συμβάλλουν στην επαναχρησιμοποίηση επιτυχημένων αρχιτεκτονικών λογισμικού
- Προάγουν την τεκμηρίωση των συστημάτων (σχεδιαστικών αποφάσεων και της λογικής – rationale)
- Αποτυπώνουν την εμπειρία των ειδικών σε έναν τομέα
- Συνιστούν ένα λεξικό τεχνολογίας λογισμικού
- Προσφέρουν στην βελτίωση της επικοινωνίας μεταξύ των developers
- Δείχνουν κάτι παραπάνω από την λύση (περιβάλλον, δυνάμεις, τελική επίλυση)
- Διευκολύνουν την μετάβαση στην αντικειμενοστραφή τεχνολογία

Μειονεκτήματα Προτύπων

- Δεν οδηγούν στην άμεση επαναχρησιμοποίηση υλοποιημένου κώδικα.
- Φαίνονται απλά σαν ιδέα ενώ στην πραγματικότητα είναι αποτέλεσμα πολυετούς εμπειρίας και προγραμματιστικού και σχεδιαστικού μόχθου.
- Έλλειψη μεθόδων και εργαλείων αυτόματου ελέγχου και επικύρωσης προτύπων.
- Η ολοκλήρωση-ενοποίηση προτύπων σε ένα περιβάλλον ανάπτυξης λογισμικού είναι δραστηριότητα αρκετά επίπονη.
- Κίνδυνος υπερφόρτωσης με πρότυπα των ομάδων ανάπτυξης λογισμικού.
- Αδυναμία άμεσης χρήσης από αρχαίους λόγω δυσκολίας κατανόησης των βαθύτερων concepts ενός προτύπου.

Θεσμοθέτηση Προτύπων Στο Χώρο Εργασίας

- Εξόρυξη του καλύτερου κώδικα για εύρεση προτύπων.
- Επέκταση της τεκμηρίωσης του σχεδιασμού και επισκόπηση πρακτικών και τεχνικών για αναζήτηση σχεδιαστικών προτύπων.
- Εκτέλεση σεμιναρίων για την χρήση υπαρχόντων σχεδιαστικών προτύπων.
- Επισκόπηση υπαρχόντων προτύπων σε workshops συγγραφέων προτύπων.
- Χρήση pattern-oriented templates για την ανάπτυξη συστημάτων λογισμικού.
- Σχηματισμός ομάδων μελέτης και συζήτησης για πρότυπα μέσα στον οργανισμό.

Τεχνική Συγγραφής Προτύπου

- Επικέντρωση σε ένα πρότυπο και αποφυγή γενικών αναφορών περί προτύπων
- Αναζήτηση σε υπάρχον λογισμικό ή σε γνώριμη προβληματική περιοχή για νέα πιθανά πρότυπα
- Αναζήτηση σε βιβλιογραφία για αποφυγή συγγραφής υπαρχόντων προτύπων
- Στόχος η ποιότητα και όχι η ποσότητα των προτύπων
- Αναζήτηση λόγου ύπαρξης ή και εφαρμογής του προτύπου
- Κυκλοφορία του προτύπου μέσω του Διαδικτύου (ιστοσελίδες, mailing lists, discussion groups) και λήψη παρατηρήσεων – Παρουσίαση σε συνέδρια και άρθρα
- Συνεχής επεξεργασία και βελτίωση - προσθήκες

Frameworks και πρότυπα

- Τα frameworks υποστηρίζουν την άμεση επαναχρησιμοποίηση λεπτομερειακής σχεδίασης και πηγαίου κώδικα.
- Τα frameworks υλοποιούνται σε μια συγκεκριμένη γλώσσα.
- Τα πιο εξελιγμένα και καλώς δομημένα frameworks εμπεριέχουν την υλοποίηση αρκετών προτύπων λογισμικού.
- Τα πρότυπα έχουν χαρακτηριστεί σαν πιο αφαιρετικές περιγραφές από τα frameworks.
- Πρότυπα έχουν χρησιμοποιηθεί επιτυχώς για την τεκμηρίωση frameworks.
- Πρότυπα → Frameworks → Libraries.

Μέλλον των προτύπων

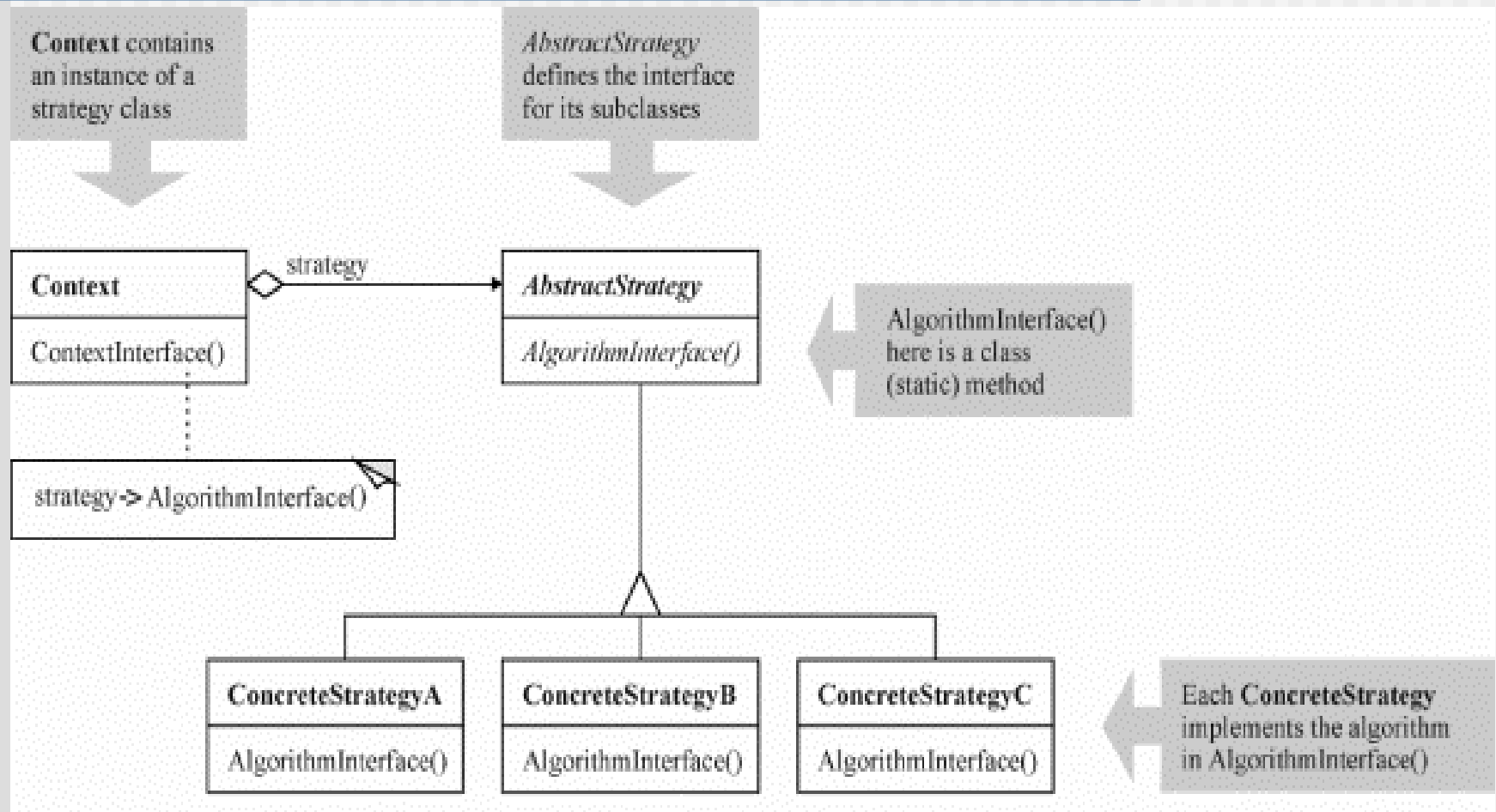
- Εκτεταμένη ενσωμάτωση των προτύπων με τα frameworks.
- Σχηματισμός νέων γλωσσών προτύπων.
- Υποστήριξη της μοντελοποίησης και αναπαράστασης προτύπων από αντικειμενοστραφείς μεθοδολογίες / συμβολισμούς σχεδίασης λογισμικού.
- Υποστήριξη από εργαλεία και περιβάλλοντα ανάπτυξης λογισμικού.
- Ανάπτυξη εμπορικών βιβλιοθηκών λογισμικού που παρέχουν επαναχρησιμοποιήσιμες υλοποιήσεις γνωστών σχεδιαστικών προτύπων.
- Ειδικό συντακτικό για την αναπαράσταση των design patterns σαν ξεχωριστές προγραμματιστικές δομές.

Σύνοψη – Τι Είναι Τα Πρότυπα

- Λύσεις για επαναλαμβανόμενα κοινά προβλήματα, κυρίως σχεδίασης λογισμικού
- Εξειδικευμένα για κάποιο συγκεκριμένο περιβάλλον δράσης (context)
- Ένας έγγραφος, δόκιμος και τυποποιημένος τρόπος για την τεκμηρίωση βέλτιστων πρακτικών.
- Μέθοδος για την επαναχρησιμοποίηση, διαμοιρασμό και ανάπτυξη πάνω σε υπάρχουσα σοφία / εμπειρία ειδικών
- Ένα κοινό λεξικό για προβλήματα ανάπτυξης λογισμικού
- Μαζικά υπέρ – προβεβλημένα ! (Massively hyped)

Παραδείγματα Προτύπων (1)

Στρατηγική (Strategy)



Παραδείγματα Προτύπων (2)

Singleton

