



ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

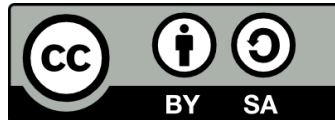
ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΑΝΑΛΥΣΗ

Εισαγωγή

Ιωάννης Σταμέλος
Βάιος Κολοφωτιάς
Πληροφορική

Άδειες Χρήσης

Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons. Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα. Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.



Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Περιεχόμενα

Άδειες Χρήσης.....	2
Χρηματοδότηση.....	2
1. Περιεχόμενο Μαθήματος	4
1.1 Περιεχόμενα ενότητας.....	4
2. ΕΙΣΑΓΩΓΙΚΕΣ ΕΝΝΟΙΕΣ	5
2.1 Τι είναι η UML?	5
2.2 Γιατί μοντελοποιούμε συστήματα πληροφορικής;.....	5
2.3 Γιατί Αντικειμενοστραφές Λογισμικό;.....	6
3. UML και Μεθοδολογίες Ανάπτυξης Λογισμικού	6
3.1 Τύποι Διαγραμμάτων της UML.....	7
3.2 Γιατί τόσοι διαφορετικοί τύποι διαγραμμάτων;.....	7

1. Περιεχόμενο Μαθήματος

Εβδομάδα	Περιεχόμενο
1 ^η	Εισαγωγή στην Αντικειμενοστρεφή Ανάλυση/UML
2 ^η	Rational Unified Process
3 ^η	Περιπτώσεις Χρήσης
4 ^η	Διαγράμματα Κλάσεων
5 ^η	Διαγράμματα Συνεργασίας
6 ^η	Διαγράμματα Ακολουθίας
7 ^η	Πρότυπα Σχεδίασης
8 ^η	Διεργασία ICONIX
9 ^η	Επιχειρηματική Μοντελοποίηση
10 ^η	Υλοποίηση Σχεδίασης με Java
11 ^η	Μετρικές Αντικειμενοστραφούς Σχεδίασης
12 ^η	Επισκόπηση

1.1 Περιεχόμενα ενότητας

Στην ενότητα αυτή θα εξετάσουμε βασικές έννοιες τις Αντικειμενοστραφούς Ανάλυσης. Θα δούμε τι είναι η γλώσσα μοντελοποίησης UML(τύπους διαγραμμάτων της,..) και πως μπορούμε να την χρησιμοποιήσουμε αναπτύσσοντας λογισμικό. Γιατί επιλέγουμε το αντικειμενοστραφές λογισμικό και μεθοδολογίες ανάπτυξης λογισμικού κατά τα στάδια ανάλυσης, σχεδίασης, κατασκευής και εξέλιξης ενός έργου πληροφορικής.

2. ΕΙΣΑΓΩΓΙΚΕΣ ΕΝΝΟΙΕΣ

2.1 Τι είναι η UML?

Η UML αποτελεί μία γλώσσα απεικόνισης ή μοντελοποίησης ενός πληροφοριακού συστήματος βασισμένου σε αντικείμενα(αντικειμενοστραφούς συστήματος).

- Η UML αποτελεί την πρότυπη (standard) γλώσσα μοντελοποίησης αντικειμενοστραφών συστημάτων.
- Η UML αποτελεί πρότυπο του OMG (Object Management Group – www.omg.org)

2.2 Γιατί μοντελοποιούμε συστήματα πληροφορικής;

- Η μοντελοποίηση ενός συστήματος παρέχει
 - τη δυνατότητα της αφαίρεσης των ασήμαντων με αυτό λεπτομερειών και της εστίασης στις σημαντικές λεπτομέρειες του συστήματος
 - τη δυνατότητα του πειραματισμού με διαφορετικές λύσεις ή προσεγγίσεις για το ίδιο πρόβλημα
 - τη δυνατότητα ανάλυσης, σχεδιασμού, καταγραφής και παρακολούθησης της προόδου ενός έργου πληροφορικής
 - μία κοινή γλώσσα για την επικοινωνία όσων εμπλέκονται στην κατασκευή του συστήματος

Χωρίς ένα μοντέλο δεν είναι δυνατόν να προσεγγίσει κανείς την πολυπλοκότητα των σύγχρονων πληροφοριακών συστημάτων.

2.3 Γιατί Αντικειμενοστραφές Λογισμικό;

- Το αντικειμενοστραφές λογισμικό είναι ευκολότερο στην αρχική του σύλληψη μια και τα αντικείμενα είναι – εν μέρει – οντότητες του υπαρκτού κόσμου
- Το αντικειμενοστραφές λογισμικό είναι ευκολότερο στην εξέλιξή του.
- Η αντικειμενοστραφής προσέγγιση επιτρέπει τη δημιουργία λογισμικού με βάση τα συστατικά (components)
- Σύγχρονες τεχνολογίες κατασκευής κατανεμημένων συστημάτων προσανατολισμένων στην σύνδεση επιχειρήσεων (Business-to-business – B2B) και στην σύνδεση επιχειρηματικών εφαρμογών (Enterprise Application Integration – EAI), όπως οι Υπηρεσίες του παγκόσμιου ιστού (Web Services), έχουν σαν τεχνολογικό υπόβαθρο αντικειμενοστραφείς γλώσσες προγραμματισμού (π.χ. Java, C#).

3. UML και Μεθοδολογίες Ανάπτυξης Λογισμικού

Μία μεθοδολογία ανάπτυξης λογισμικού παρέχει μία συστηματική προσέγγιση στην διαδικασία ανάλυσης, σχεδίασης, κατασκευής και εξέλιξης ενός έργου πληροφορικής.

- Είναι μία σειρά σταδίων τα οποία περιγράφουν συγκεκριμένες εργασίες.
- **Η UML είναι ουδέτερη σε σχέση με τις μεθοδολογίες χωρίς να επιβάλλει κάποια συγκεκριμένη μεθοδολογία**
- Υπάρχουν πολλές μεθοδολογίες ανάπτυξης λογισμικού (όπως η Unified Process, eXtreme Programming, Catalysis, Syntropy κoκ.). Κάθε μία από αυτές θεωρείται πως είναι πιο κατάλληλη για κάποιον συγκεκριμένο τύπο λογισμικού, για παράδειγμα η Catalysis θεωρείται πως είναι πιο κατάλληλη για συστήματα βασισμένα σε συστατικά (component based systems).

3.1 Τύποι Διαγραμμάτων της UML

- Διάγραμμα Περιπτώσεων Χρήσης (Use Case Diagram)
- Διάγραμμα Κλάσεων (Class Diagram)
- Διαγράμματα Συμπεριφοράς (Behavior Diagrams):
 - Διάγραμμα Καταστάσεων (Statechart Diagram)
 - Διάγραμμα Δραστηριοτήτων (Activity Diagram)
 - Διαγράμματα Αλληλεπίδρασης
 - Διάγραμμα Ακολουθίας (Sequence Diagram)
 - Διάγραμμα Συνεργασίας (Collaboration Diagram)
- Διαγράμματα Υλοποίησης (Implementation Diagrams):
 - Διάγραμμα Συστατικών (Component Diagram)
 - Διάγραμμα Διάταξης (Deployment Diagram)

3.2 Γιατί τόσοι διαφορετικοί τύποι διαγραμμάτων;

- Το σύστημα έχει στατικά στοιχεία, δυναμικά στοιχεία, στοιχεία υλοποίησης κοκ.
- Δεν είναι δυνατόν ένας και μόνο τύπος διαγράμματος να περιγράψει όλες αυτές τις διαφορετικές οπτικές γωνίες ενός συστήματος.
- Έτσι τυπικά χρησιμοποιούμε
 - διαγράμματα κλάσεων για την περιγραφή των στατικών σχέσεων μεταξύ των κλάσεων,
 - διαγράμματα συμπεριφοράς (κατάστασης, ακολουθίας κοκ) για την περιγραφή της δυναμικής συμπεριφοράς του συστήματος,
 - διαγράμματα υλοποίησης (συστατικών, διάταξης) για την καταγραφή των λεπτομερειών υλοποίησης του συστήματος.
- Τα διαγράμματα περιπτώσεων χρήσης είναι μία ειδική περίπτωση υπό την έννοια πως δεν έχουν κάποια σχέση με αντικείμενα ή αντικειμενοστραφή συστήματα.

