ΤΕΙ ΚΕΝΤΡΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ

ΑΝΟΙΚΤΑ ΑΚΑΔΗΜΑΙΚΑ ΜΑΘΗΜΑΤΑ

# ΤΕΧΝΟΛΟΓΙΑ ΛΟΓΙΣΜΙΚΟΥ Ι


κ. ΠΕΤΑΛΙΔΗΣ

***ΤΜΗΜΑ*** **ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΕ**



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

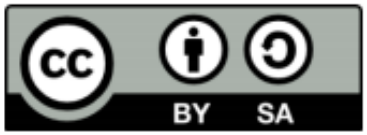
ΕΣΠΑ 2007-2013
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

# 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 ΤΕΙ Κεντρικής Μακεδονίας» 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.

# 7η Διάλεξη

## *Η καταγραφή των απαιτήσεων*

# Καταγραφή απαιτήσεων

Μια διαδικασία ανάπτυξης ξεκινά συνήθως με την καταγραφή απαιτήσεων

Το βήμα αυτό είναι και το πιο σημαντικό γιατί προσπαθούμε να καταλάβουμε **τι** πρέπει να κάνουμε

Ένας αρκετά συνηθισμένος τρόπος καταγραφής απαιτήσεων είναι οι περιπτώσεις χρήσης

# Καταγραφή απαιτήσεων

Σε μια επαναληπτική διαδικασία ανάπτυξης στόχος δεν είναι να βρούμε όλες τις απαιτήσεις στην πρώτη επανάληψη ούτε να τις αναλύσουμε πλήρως

Συνήθως κοιτούμε πρώτα τις πιο σημαντικές απαιτήσεις που θα καθορίσουν την αρχιτεκτονική του συστήματός μας και

# Απαιτήσεις

Με τον όρο “απαίτηση” εννοούμε ένα χαρακτηριστικό που πρέπει να έχει το σύστημα προκειμένου να εκπληρώνει το σκοπό του.

Οι απαιτήσεις περιγράφουν τι θέλουμε να κάνει το σύστημα, **αλλά όχι πως να το κάνει**

Η καταγραφή τους είναι το πρώτο και το πιο σημαντικό βήμα που πρέπει να

# Η διαδικασία συλλογής και καταγραφής απαιτήσεων

Μπορεί να περιλαμβάνει διάφορα στάδια:

- συζήτηση με τον πελάτη, παρουσίαση ανάλογων συστημάτων και κατανόηση του συστήματος
- ορισμός των απαιτήσεων σε κάποιο κείμενο
- Επαλήθευση ότι οι απαιτήσεις δεν αλληλοαναιρούνται

# Κάποιες απαιτήσεις είναι πιο σημαντικές

- Κάποιες απαιτήσεις πρέπει να υλοποιηθούν οπωσδήποτε
- Κάποιες είναι περισσότερο αναγκαίες από κάποιες άλλες
- Και κάποιες μπορούν να μην υλοποιηθούν

# ειτουργικές και μη-λειτουργικές απαιτήσεις

Οι απαιτήσεις μπορεί να είναι **λειτουργικές** ή **μη-λειτουργικές**

Οι απαιτήσεις λειτουργικότητας αφορούν κυρίως τις συναλλαγές του συστήματός μας με το περιβάλλον του: τι μπορεί να δεχθεί σαν είσοδο και τι θα δώσει σαν έξοδο

Μη λειτουργικές απαιτήσεις είναι αυτές που θέτουν περιορισμούς στο σύστημα, πχ: το

# Βήματα για την καταγραφή των απαιτήσεων

Βήμα 1

- Ζητήστε μια περιγραφή του προβλήματος που προσπαθείτε να λύσετε. Την περιγραφή αυτή θα σας τη δώσει ο πελάτης σας, οι χρήστες του συστήματος που θα υλοποιήσετε ή ακόμα και οι ανταγωνιστές σας!

Βήμα 2

- Φτιάξτε ένα γλωσσάρι που να εξηγεί τους

# Βήματα για την καταγραφή των απαιτήσεων

## Βήμα 3

- Κάντε μια βασική καταγραφή και αρίθμηση των απαιτήσεων που πρέπει να πληροί το σύστημά σας

## Βήμα 4

- Βρείτε πόσους διαφορετικούς τύπους χρηστών έχει το σύστημα που θα φτιάξετε

## Βήμα 5

- **Βρείτε τις βασικές έννοιες(κλάσεις)** του συστήματός σας (domain model) και αναπαραστήστε τες με ένα

# Βήματα για την καταγραφή των απαιτήσεων

Βήμα 6

- Βρείτε τις περιπτώσεις χρήσης. Φτιάξτε μια περίπτωση χρήσης για τη δημιουργία/αλλαγή/διαγραφή των βασικών εννοιών του συστήματός σας

Βήμα 7

- Περιγράψτε τις περιπτώσεις χρήσης και δώστε και μια συνολική περιγραφή του συστήματός σας.

# Πώς ανακαλύπτουμε τις βασικές κλάσεις ενός συστήματος;

- Κοιτάξτε την περιγραφή του προβλήματος που σας έχουν δώσει.
- Υπογραμμίστε ό,τι ουσιαστικά εμφανίζονται
- Υπογραμμίστε ό,τι οργανισμοί, ό,τι ρόλοι εμφανίζονται

# Παράδειγμα

Ποιες κλάσεις αναγνωρίζετε στην παρακάτω περιγραφή;

- *Φτιάξτε ένα σύστημα που να το χρησιμοποιούν οι μαθητές για να αξιολογούν τα βιβλία τους. Οι προϊστάμενοι του ΤΕΙ ή οι γραμματείς τους θα μπορούν να εισάγουν νέα βιβλία για αξιολόγηση και να βλέπουν τις αξιολογήσεις που έχουν γίνει. Οι εκδότες θα πρέπει να μπορούν να εισέρχονται στο σύστημα και να βλέπουν τις αξιολογήσεις που έχουν γίνει. Τέλος ένας ειδικός χρήστης θα μπορεί να δημιουργεί νέους χρήστες (πχ*

# Προσοχή

Πολλοί κάνουν το λάθος και προσπαθούν να ανακαλύψουν ενέργειες όταν διαβάζουν μια περιγραφή ενός προβλήματος

Αυτός ήταν ο τρόπος δημιουργίας προγραμμάτων στο διαδικασιακό προγραμματισμό (procedural programming)

# Παράδειγμα

Ένας μαθητής εγγράφεται σε ένα μάθημα, ή μπορεί να διαγραφεί από αυτό

# Λύση

Λανθασμένη λύση

- Υπάρχει η κλάση μαθητής, η κλάση εγγραφή και η κλάση διαγραφή
- Αυτή η προσέγγιση είναι λάθος!
- Έτσι θα σκεφτόσασταν αν φτιάχνατε συναρτήσεις στη C αλλά όχι στον αντικειμενοστραφή προγραμματισμό!

Ορθότερη λύση

- Υπάρχει η κλάση μαθητής και η κλάση

# Χαρακτηριστικά (attributes)

Κάθε κλάση έχει κάποια χαρακτηριστικά ή ιδιότητες

Για παράδειγμα μια κλάση «Τραπεζικός Λογαριασμός» θα έχει ως χαρακτηριστικό ένα «Αριθμό Λογαριασμού»

Για παράδειγμα η «Πιστωτική κάρτα» έχει ως χαρακτηριστικά *(αριθμό, τύπο και PIN)*

# Μέθοδοι

Συνήθως στα αντικείμενα μιας κλάσης μπορούν να γίνουν και κάποιες ενέργειες. Για παράδειγμα στα αντικείμενα της κλάσης Πιστωτική Κάρτα θα μπορούσε να γίνει η ενέργεια «Χρέωση» ή «Συναλλαγή»

# Πως βρίσκουμε τις μεθόδους;

Ψάξτε στην περιγραφή του προβλήματος σας για ρήματα. Αυτά συνήθως δηλώνουν ενέργειες που μπορούν να γίνουν σε μια κλάση ή λειτουργίες που ξέρει να διεκπεραιώνει.

# Παράδειγμα

*Φτιάξτε ένα σύστημα που να το χρησιμοποιούν οι μαθητές για να αξιολογούν τα βιβλία τους. Οι προϊστάμενοι του ΤΕΙ ή οι γραμματείς τους θα μπορούν να εισάγουν νέα βιβλία για αξιολόγηση και να βλέπουν τις αξιολογήσεις που έχουν γίνει. Οι εκδότες θα πρέπει να μπορούν να εισέρχονται στο σύστημα και να βλέπουν τις αξιολογήσεις που έχουν γίνει. Τέλος ένας ειδικός χρήστης θα μπορεί να δημιουργεί νέους χρήστες (πχ προϊσταμένους, εκδότες κτλ).*

# Συσχετίσεις

Μια κλάση μπορεί να συσχετίζεται με μια άλλη.

Στο παράδειγμα με τους μαθητές και τα βιβλία, ένας μαθητής μπορεί να συσχετιστεί με τα βιβλία που έχει αξιολογήσει

# Συσχετίσεις

Μια συσχέτιση απλά τονίζει συνήθως ότι μια κλάση έχει ως χαρακτηριστικό, κάτι που από μόνο του είναι και αυτό μια κλάση.

Για παράδειγμα ένας πελάτης έχει μια διεύθυνση. Και ο «πελάτης» είναι κλάση και η διεύθυνση είναι «κλάση». Επειδή η κλάση πελάτης περιέχει ως χαρακτηριστικό την κλάση «Διεύθυνση»

# Ένα ακόμα παράδειγμα

Ένα ηλεκτρονικό κατάστημα έχει πελάτες είτε εταιρείες είτε ιδιώτες.

Οι εταιρείες έχουν έκπτωση στα προϊόντα που αγοράζουν ενώ οι ιδιώτες όχι. Οι ιδιώτες μπορούν να ζητούν δώρα από το κατάστημα αν κάνουν πολλές αγορές

Κάθε εταιρεία δηλώνει τη διεύθυνση, την επωνυμία  και το ΑΦΜ της

Κάθε ιδιώτης δηλώνει μόνο τη διεύθυνση

# (συνέχεια)

Υπάρχει η κλάση προϊόν, η κλάση πελάτης, η κλάση εταιρεία και η κλάση ιδιώτης

Οι τρεις τελευταίες όμως έχουν πολλά κοινά μεταξύ τους!

Στον αντικειμενοστραφή προγραμματισμό προσπαθούμε να βρούμε τέτοιες κλάσεις που έχουν κοινά στοιχεία

# Κληρονομικότητα

Χρησιμοποιούμε την κληρονομικότητα ως εξής

- Ορίζουμε την κλάση **πελάτης** με
  - τα κοινά χαρακτηριστικά των κλάσεων **εταιρεία**, **ιδιώτης**
  - τις κοινές ενέργειες που μπορούν να κάνουν οι δύο κλάσεις
- Ορίζουμε ότι την κλάση **εταιρεία** με
  - μόνο τα ιδιαίτερα χαρακτηριστικά της
  - και τις ιδιαίτερες ενέργειες
- Παρόμοια ορίζουμε και την κλάση **ιδιώτης**
- Τα υπόλοιπα χαρακτηριστικά τους τα