

## Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση

Τόμ. 1 (2005)

3ο Πανελλήνιο Συνέδριο «Διδακτική της Πληροφορικής»



Ένα Σενάριο Διδασκαλίας των Βασικών Εννοιών-Αρχών του Αντικειμενοστρεφούς Προγραμματισμού

Σταυρούλα Γεωργαντάκη, Συμεών Ρετάλης

### Βιβλιογραφική αναφορά:

Γεωργαντάκη Σ., & Ρετάλης Σ. (2026). Ένα Σενάριο Διδασκαλίας των Βασικών Εννοιών-Αρχών του Αντικειμενοστρεφούς Προγραμματισμού. *Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση*, 1, 509–518. ανακτήθηκε από <https://eproceedings.epublishing.ekt.gr/index.php/cetpe/article/view/8746>

# Ένα Σενάριο Διδασκαλίας των Βασικών Εννοιών-Αρχών του Αντικειμενοστρεφούς Προγραμματισμού

Σταυρούλα Γεωργαντάκη, Συμεών Ρετάλης  
Τμήμα Διδακτικής Τεχνολογίας και Ψηφιακών Συστημάτων  
Πανεπιστήμιο Πειραιά  
[rgeo@unipi.gr](mailto:rgeo@unipi.gr), [retal@unipi.gr](mailto:retal@unipi.gr)

## ΠΕΡΙΛΗΨΗ

Η εργασία αυτή αναφέρεται στη διδασκαλία του Αντικειμενοστρεφούς Προγραμματισμού και την εφαρμογή μίας συγκεκριμένης διδακτικής προσέγγισης η οποία εφαρμόστηκε πιλοτικά σε σύντομης διάρκειας σεμινάριο. Η συγκεκριμένη διδακτική μεθοδολογία βασίστηκε στην αντιμετώπιση των διδακτικών προβλημάτων που έχουν καταγραφεί στη διεθνή βιβλιογραφία και είναι απόλυτα σύμφωνη με τις οδηγίες διδασκαλίας που προβλέπονται από την ACM. Από τη σύντομη μελέτη αξιολόγησης, προέκυψαν θετικά συμπεράσματα ως προς την αποτελεσματικότητα της διδακτικής μεθοδολογίας και επίσης αναδείχθηκαν μη καταγεγραμμένες στη βιβλιογραφία δυσκολίες και παρανοήσεις.

**ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ:** Αντικειμενοστρεφής Προγραμματισμός, Μεθοδολογία διδασκαλίας, Διδακτικά προβλήματα

## ΕΙΣΑΓΩΓΗ

Η διδασκαλία του Αντικειμενοστρεφούς Προγραμματισμού έχει εισαχθεί τα τελευταία χρόνια στο Πρόγραμμα Σπουδών των Πανεπιστημιακών Τμημάτων και συνήθως πραγματοποιείται αφού έχει προηγηθεί η εισαγωγή στον Προγραμματισμό με την διαδικαστική (procedural) φιλοσοφία Προγραμματισμού. Ανεξάρτητα από τη θέση που έχει στο Πρόγραμμα Σπουδών, η εκμάθηση του Αντικειμενοστρεφούς Προγραμματισμού παρουσιάζει δυσκολίες (Satztrzemí et al. 2003).

Στόχος της εργασίας αυτής είναι να παρουσιαστούν τα έως τώρα πορίσματα έρευνας σχετικής με τη διδασκαλία του αντικειμενοστρεφούς προγραμματισμού. Η έρευνα αφορά στο να αναπτυχθεί μία ολοκληρωμένη διδακτική μεθοδολογία που θα περιλαμβάνει το σχεδιασμό μαθησιακών δραστηριοτήτων, την ανάπτυξη υποστηρικτικού μαθησιακού υλικού και την αξιοποίηση εκπαιδευτικών κι άλλων εργαλείων για την διδασκαλία του Αντικειμενοστρεφούς Προγραμματισμού. Η διδακτική μεθοδολογία θα πρέπει να δίνει λύσεις στα διδακτικά προβλήματα που αναφέρονται στη διεθνή βιβλιογραφία και επιβάλλεται να συνάδει με τις οδηγίες διδασκαλίας (ενότητες, στόχοι, κοκ) που προβλέπονται από τον οδηγό σπουδών σε θέματα πληροφορικής της ACM (ACM 2001).

Η διδακτική μεθοδολογία που σχεδιάστηκε χρησιμοποιήθηκε για την διεξαγωγή σεμιναρίου το οποίο απευθυνόταν στους φοιτητές του ΠΜΣ του Τμήματος Διδακτικής της Τεχνολογίας και Ψηφιακών Συστημάτων (ΔΤΨΣ) του Πανεπιστημίου Πειραιά.

Στην παρούσα εργασία παρουσιάζονται οι καταγεγραμμένες δυσκολίες στην εκμάθηση του Αντικειμενοστρεφούς Προγραμματισμού, η προτεινόμενη διδακτική μεθοδολογία (φιλοσοφία, υλικό, εργαλεία), οι στόχοι της πιλοτικής έρευνας αξιολόγησης, στοιχεία για την πιλοτική έρευνα που διεξήχθη και τέλος τα αποτελέσματα της.

## **ΚΑΤΑΓΕΓΡΑΜΜΕΝΕΣ ΔΥΣΚΟΛΙΕΣ ΚΑΙ ΠΑΡΑΝΟΗΣΕΙΣ**

Στη διεθνή βιβλιογραφία πλήθος διδακτικών προβλημάτων είναι καταγεγραμμένα. Προέρχονται από έρευνες σε μαθητές Γυμνασίου, Λυκείου, προπτυχιακούς και μεταπτυχιακούς σπουδαστές. Συγκεκριμένα:

Από έρευνα που διεξήχθη σε μαθητές Γυμνασίου (Teif & Hazzan 2004) που παρακολούθησαν μαθήματα εισαγωγής σε Αντικειμενοστρεφείς έννοιες με την χρήση γραφικού περιβάλλοντος «χελώνας» (μικρόκοσμος) και συλλογή δεδομένων μέσω ερωτηματολογίων, εργασιών, συνεντεύξεων και παρατήρησης, προέκυψε ότι:

- συγχέουν τις ιδιότητες (attributes) ενός αντικειμένου με τα μέρη (parts) από τα οποία αυτό αποτελείται και επίσης με τις ενέργειες που μπορεί να εκτελέσει
- θεωρούν ότι ένα σύνολο από αντικείμενα (set of objects) π.χ. μια ομάδα ποδοσφαίρου, είναι κλάση και όχι αντικείμενο και
- συγχέουν τις σχέσεις κλάσης-αντικειμένου (class-object), ιεραρχίας (hierarchy) και όλου-μέρους (whole-part).

Από έρευνα που αφορούσε στην αντίληψη περί της δημιουργίας στιγμιοτύπων (instantiation) με τον **κατασκευαστή (constructor)** και τους διαφορετικούς τρόπους **αρχικοποίησης στιγμιοτύπων (initialization)**, η οποία διεξήχθη σε μαθητές Λυκείου (Ragonis & Ben-Ari 2002) που παρακολούθησαν εισαγωγικά μαθήματα Αντικειμενοστρεφούς Προγραμματισμού με την χρήση του Bluej.

- αναδείχθηκε η δυσκολία στην κατανόηση του κατασκευαστή γενικά και της θέσης του μέσα στην κλάση, αλλά και ιδιαίτερα στις περιπτώσεις που η αρχικοποίηση γίνεται με άλλον τρόπο από αυτόν της χρήσης κατασκευαστή με παραμέτρους
- επίσης κάποιοι μαθητές θεωρούν πως μόνον η δήλωση του κατασκευαστή μέσα στην κλάση αρκεί για να δημιουργηθεί ένα αντικείμενο και δυσκολεύονται να κατανοήσουν την χρήση του τελεστή new και τις λειτουργίες που αυτή συνεπάγεται.

Από παρατήρηση των σπουδαστών στα πλαίσια του εξ' αποστάσεως προπτυχιακού μαθήματος "Computing: An object-oriented approach" και του εξ' αποστάσεως μεταπτυχιακού μαθήματος "Object-oriented Software Technology" και τα δύο με τη χρήση της γλώσσας Smalltalk (Holland et al. 1997), αναδείχθηκαν παρανοήσεις, όπως:

- Ταύτιση αντικειμένου/μεταβλητής, αν στα παραδείγματα αναφέρονται συχνά κλάσεις με μια μόνο μεταβλητή.

- Λανθασμένη θεώρηση ότι όλες οι μεταβλητές μιας κλάσης είναι του ίδιου τύπου π.χ. αριθμοί.
- Λανθασμένη αντίληψη ότι τα αντικείμενα είναι απλές εγγραφές, χωρίς συμπεριφορά, όταν υπερτονίζεται το κομμάτι των μεταβλητών μιας κλάσης σε βάρος της συμπεριφοράς της (methods).
- Ταύτιση κλάσης/αντικειμένου ιδίως όταν χρησιμοποιούνται συχνά παραδείγματα κλάσεων από τις οποίες δημιουργείται μόνο ένα αντικείμενο.
- Σύγχυση ταυτότητας/μεταβλητής που καλείται name ή κάπως παρόμοια. Το name συγγέεται με την ταυτότητα του αντικειμένου και αντικείμενα που έχουν την ίδια τιμή σε αυτή την μεταβλητή θεωρούνται το ίδιο αντικείμενο. Επίσης το name συγγέεται και με την μεταβλητή αναφοράς που συνδέεται με το αντικείμενο. Αυτές οι συγχύσεις οδηγούν και σε άλλες παρανοήσεις που αφορούν στην σύνδεση περισσότερων της μιας μεταβλητής αναφοράς στο ίδιο αντικείμενο, στην δυνατότητα αποσύνδεσης μιας μεταβλητής αναφοράς από ένα αντικείμενο και σύνδεσης της με άλλο και στην διάκριση αντικειμένων με ακριβώς ίδια κατάσταση.
- Μια μέθοδος μεταβάλλει την κατάσταση ενός αντικειμένου απλά και μόνο με εντολές ανάθεσης και όχι με αποστολή μηνυμάτων.

Από συνεντεύξεις σπουδαστών που παρακολούθουν εισαγωγικά μαθήματα με την χρήση της Java (Fleury 2000), αναδείχθηκαν κανόνες που οι σπουδαστές λανθασμένα «οικοδομούν» όταν την αποκτημένη γνώση την επεκτείνουν και πέραν της ορθής περιοχής εφαρμογής της, όπως:

- Μέθοδοι διαφορετικών κλάσεων μπορούν να έχουν το ίδιο όνομα μόνο αν έχουν διαφορετικές υπογραφές.
- Η χρήση του κατασκευαστή είναι προαιρετική. Ο μόνος λόγος χρήσης του είναι η αρχικοποίηση των μεταβλητών του αντικειμένου. Αν καταργήσεις τον κατασκευαστή και τον αντικαταστήσεις με μια μέθοδο που αρχικοποιεί τις μεταβλητές θεωρούν ότι δεν υπάρχει πρόβλημα. Δεν αντιλαμβάνονται δηλαδή την κατανομή μνήμης για τα αντικείμενα, που συμβαίνει με την κλήση του κατασκευαστή, αλλά μόνον το κομμάτι της αρχικοποίησης.
- Η χρήση του dot operator είναι έγκυρη μόνο για την κλήση μεθόδων και όχι για την αναφορά σε ιδιοχαρακτηριστικά (επειδή αυτά είναι συνήθως private και ο dot operator δεν χρησιμοποιείται συχνά).

Αναδείχθηκαν επίσης (Fleury 2001) προβλήματα σχετικά με την κατανόηση της αρχής της κελυφοποίησης (encapsulation) και τον σχεδιασμό κώδικα που μπορεί να επαναχρησιμοποιηθεί (code re-use), όπως:

- Ο σχεδιασμός μιας κλάσης με public ιδιοχαρακτηριστικά αντί private (που αντιβαίνει την αρχή του encapsulation) θεωρήθηκε προτιμητέα από μερικούς σπουδαστές λόγω της μικρότερης έκτασης του κώδικα.
- Η κατάργηση μιας κλάσης σε ένα πρόγραμμα και η μεταφορά όλης της ‘δουλειάς της’ στην μέθοδο main (που μειώνει την κελυφοποίηση και την πιθανότητα επαναχρησιμοποίησης κώδικα), θεωρήθηκε από κάποιους σπουδαστές λόγος για

ευκολότερη κατανόησή του, αφού δεν χρειάζεται να «μεταπηδάς» από κλάση σε κλάση όταν το διαβάσεις.

- Η δημιουργία δύο παρόμοιων κλάσεων και η χρήση ενός αντικειμένου από την κάθε μια, θεωρήθηκε προτιμότερη από την γενικότερη περίπτωση της ύπαρξης μιας κλάσης και την δημιουργία δύο αντικειμένων από αυτή, γιατί εκτιμούν ότι ο κώδικας είναι σαφέστερος, πιο εύκολος στην συντήρηση και στην ανίχνευση λαθών. Δηλαδή προτιμούν την ιδιαιτερότητα των διπλών κλάσεων από την γενικότητα της επαναχρησιμοποίησης κλάσεων.

## ΔΙΔΑΚΤΙΚΗ ΜΕΘΟΔΟΛΟΓΙΑ

### Φιλοσοφία

Η προτεινόμενη διδακτική μεθοδολογία βασίστηκε στη θεώρηση ότι οι Αντικειμενοστρεφείς έννοιες-θέματα πρέπει να εισάγονται σταδιακά.. Δεν υιοθετήθηκε η προσέγγιση της παρουσίας όλων των Αντικειμενοστρεφών εννοιών-αρχών από την αρχή. Η επιλογή αυτή έγινε, εκτιμώντας ότι είναι αναγκαίο οι φοιτητές να εισαχθούν βαθμιαία στην Προγραμματιστική αυτή φιλοσοφία. Γίνεται χρήση παραδειγμάτων κλιμακούμενης δυσκολίας, σε αντίθεση με τους ισχυρισμούς του Kristen Nygaard (2005), ότι η διδασκαλία της Αντικειμενοστρέφειας θα πρέπει να ξεκινά με επαρκώς πολύπλοκα παραδείγματα. Ο ισχυρισμός αυτός έχει αξία στην περίπτωση της ανάλυσης, αλλά όχι στο πλαίσιο διδασκαλίας Προγραμματισμού, όπου τα ‘μοντέλα’ χρειάζεται να υλοποιηθούν σε μια γλώσσα προγραμματισμού (Bennedsen & Caspersen 2004).

Στην αρχή κάθε ενότητα διατυπώνονται οι Διδακτικοί της Στόχοι και ειδικά σχεδιασμένες ερωτήσεις-ασκήσεις, στοχεύουν στην εκτίμηση της επίτευξης των στόχων αυτών. Έγινε συστηματική χρήση κατάλληλα επιλεγμένων παραδειγμάτων για την παρουσίαση εννοιών. Έμφαση δόθηκε στις διαπιστωμένες δυσκολίες, φροντίζοντας να υπάρχουν παραδείγματα και λυμένες ασκήσεις που στόχευαν στην αποφυγή δημιουργίας παρανοήσεων και παρερμηνειών. Το υλικό που σχεδιάστηκε δεν στόχευε να αποτελεί την μοναδική αναφορά των σπουδαστών που διδάσκονται Αντικειμενοστρεφή Προγραμματισμό, αλλά συμπληρωματικό-υποστηρικτικό υλικό για την κατανόηση-αποσαφήνιση εννοιών και την εμπάθυνση σε αυτές. Περιέχει σύντομα και περιεκτικά κομμάτια θεωρίας, αντιπροσωπευτικά παραδείγματα, λυμένες ασκήσεις, δραστηριότητες σε εκπαιδευτικά περιβάλλοντα.

Οι εκπαιδευτικές δραστηριότητες και το υλικό καλύπτουν επτά ενότητες.

1. **Φιλοσοφία Αντικειμενοστρεφούς Προγραμματισμού:** Οι φοιτητές έρχονται σε μια πρώτη επαφή με την νέα φιλοσοφία, τις βασικές της έννοιες και τα οφέλη χρήσης της.
2. **Βασικές έννοιες (Κλάση, Αντικείμενο, Ιδιότητες, Μέθοδος):** Περιλαμβάνονται κατάλληλα παραδείγματα για την διάκριση των εννοιών κλάση-αντικείμενο και την αποφυγή των καταγεγραμμένων παρανοήσεων και συγχύσεων π.χ. ταύτιση αντικειμένου/κλάσης, αντικειμένου/μεταβλητής, αντικειμένων με ακριβώς ίδια κατάσταση.

3. **Αφαιρετικότητα – Κελυφοποίηση - Απόκρυψη πληροφορίας – Διαχωρισμός συμπεριφοράς και υλοποίησης:** Παρουσιάζονται οι έννοιες με κατάλληλα επιλεγμένα παραδείγματα οντοτήτων, κλάσεων και κλάσεων με διαφορετικούς τρόπους σχεδίασης.
4. **Κατασκευαστής – Δημιουργία αντικειμένων – Αναφορές σε αντικείμενα:** Παρουσιάζονται παραδείγματα χρήσης του `by default constructor` σε μια κλάση, διαφορετικών τρόπων αρχικοποίησης των μεταβλητών μιας κλάσης, δήλωσης ενός και πολλαπλών `constructor` στην κλάση και δημιουργίας αντικειμένων με την κλήση του/ς. Δόθηκε έμφαση στην διαδικασία δημιουργίας αντικειμένων, την κατανομή μνήμης και την έννοια των μεταβλητών αναφοράς.
5. **Μεταβλητές και μέθοδοι κλάσης:** Παρουσιάζονται οι έννοιες με κατάλληλα επιλεγμένα παραδείγματα κλάσεων.
6. **Υπερφότωση μεθόδων:** Χρησιμοποιούνται κατάλληλα σχεδιασμένα παραδείγματα προγραμμάτων που επιδεικνύουν την έννοια.
7. **Κληρονομικότητα – Πολυμορφισμός:** Παρουσιάζονται οι έννοιες της ιεραρχίας κλάσεων, υπερκλάσης, υποκλάσεων, σχέσης «είναι ένα» (“is a”), υπερκάλυψης (`overriding`) μεθόδων, παραμετρικού πολυμορφισμού, πολυμορφισμού υποτύπων, δυναμικής δέσμευσης.

Η όλη προσέγγιση στο σχεδιασμό του μαθησιακού υλικού βρίσκεται σε συμφωνία με τις οδηγίες «μην ξεκινάς με κενή οθόνη», «μελέτησε κώδικα», «μην ξεκινάς με την `main`», «μην ξεκινάς με το γνωστό παράδειγμα ‘Hello World’» που διατύπωσαν οι Kölling & Rosenberg (Kölling & Rosenberg 2001).

### Εκπαιδευτικά Εργαλεία

Για την καλύτερη εμπέδωση των εννοιών και την πρακτική εξάσκηση, οι σπουδαστές έκαναν χρήση εκπαιδευτικών εργαλείων. Τα εργαλεία που προτείνονται είναι:

**1. Εκπαιδευτικό Προγραμματιστικό Περιβάλλον Bluej** (<http://www.bluej.org>) (Kölling et al. 2003). Στις μεθοδολογίες διδασκαλίας που έχουν διατυπωθεί για την αντιμετώπιση προβλημάτων, συγκαταλέγεται και η χρήση εκπαιδευτικών προγραμματιστικών περιβαλλόντων. Τέτοιο είναι το Bluej, το οποίο εστιάζει στην αντικειμενοστρέφεια και αποτελεί ένα πλήρες περιβάλλον ανάπτυξης εφαρμογών σε Java. **Σημαντικά χαρακτηριστικά του Bluej:** Είναι εύκολο στη χρήση του σε σχέση με άλλα IDE – Παρέχει την δυνατότητα να αντιληφθείς ότι μια εφαρμογή είναι ένα σύνολο από συνεργαζόμενες κλάσεις – Αποκτούν σε αυτό οπτική υπόσταση οι κλάσεις και τα αντικείμενα και δεν αποτελούν απλά γραμμές κώδικα – Διαθέτοντας μόνο τον κώδικα μιας κλάσης, μπορούμε να δημιουργήσουμε αντικείμενα και να εκτελέσουμε μεθόδους χωρίς να γράψουμε ούτε μια γραμμή κώδικα γι’ αυτό το σκοπό – Επιτρέπεται να αλληλεπιδράς με τις κλάσεις δημιουργώντας αντικείμενα και επίσης με τα αντικείμενα εκτελώντας μεθόδους – Διαθέτει όψη υλοποίησης και όψη διαπροσωπείας των κλάσεων – Εύκολο πέρασμα από το οπτικό περιβάλλον στο περιβάλλον κώδικα και εύκολη μεταγλώττιση με επισήμανση των λαθών κατευθείαν στον συντάκτη του κώδικα, με τονισμό της γραμμής του λάθους και επίδειξη του κειμένου του μηνύματος λάθους.

**2. Προγραμματιστικός Μικρόκοσμος Jeroo (<http://www.jeroo.org>)** (Sanders & Dorn 2003). Η χρήση των Προγραμματιστικών Μικρόκοσμων έχει προταθεί επίσης για την αντιμετώπιση προβλημάτων. Επιλέχθηκε η χρήση του Jeroo για την προσέγγιση των εννοιών αντικείμενο και μέθοδος στις οποίες από κατασκευής το περιβάλλον αυτό εστιάζει. **Σημαντικά χαρακτηριστικά του Jeroo:** Είναι απλό και εστιάζει στις έννοιες object και method – Όλα τα συστατικά του περιβάλλοντος είναι ορατά σε ένα παράθυρο διαρκώς – Οπτικοποίηση της εκτέλεσης του προγράμματος, με κίνηση των αντικειμένων – Επισήμανση με διαφορετικό χρώμα της γραμμής κώδικα που εκτελείται – Δυνατότητα βήμα προς βήμα και συνεχόμενης εκτέλεσης – Επιλογή ταχύτητας εκτέλεσης – Ευκολία σύνταξης της γλώσσας του περιβάλλοντος και συνάφεια της με τη Java – Δυνατότητα εύκολης δημιουργίας πολλών διαφορετικών Island layouts (test cases) για τον έλεγχο των προγραμμάτων – Δυνατότητα επεξεργασίας προβλημάτων με ενδιαφέρον και όχι προβλημάτων επεξεργασίας συμβόλων και αριθμών.

## Η ΠΙΛΟΤΙΚΗ ΕΡΕΥΝΑ ΑΞΙΟΛΟΓΗΣΗΣ

### Στόχος

Στην αξιολόγηση της διεξαγωγής του σεμιναρίου, έμφαση δόθηκε στην αντιμετώπιση των καταγεγραμμένων δυσκολιών, στην ανίχνευση μη καταγεγραμμένων δυσκολιών και παρανοήσεων και στην διαπίστωση της αντίληψης των φοιτητών για θέματα που δεν έχουν μελετηθεί.

### Το δείγμα

Συμμετείχαν 28 φοιτητές, 15 άνδρες και 13 γυναίκες, ηλικίας από 23 έως 47 ετών, οι οποίοι είναι μεταπτυχιακοί φοιτητές στο Τμήμα Διδακτικής της Τεχνολογίας και Ψηφιακών Συστημάτων. Οι φοιτητές προέρχονταν κι από τις τρεις (3) κατευθύνσεις του μεταπτυχιακού προγράμματος (Ηλεκτρονική Μάθηση, Ψηφιακές Επικοινωνίες και Δίκτυα, Δικτυοκεντρικά Συστήματα). Από αυτούς 15 προέρχονταν από Τμήματα Α.Ε.Ι.-Τ.Ε.Ι. του χώρου της Πληροφορικής (Πληροφορικής, Ηλ. Μηχανικών και Μηχ. Η/Υ, ΔΤΨΣ, Εφαρμ. Μαθηματικών και Πληροφορικής), 5 από Σχολές Θετικών Επιστημών, 2 από Τμήματα Ηλεκτρονικής, 2 από τη Φιλοσοφική Σχολή, 1 από Τμήμα Ηλεκτρολογίας, 1 από Τμήμα Πολιτικών Μηχανικών, 1 από Παιδαγωγικό Τμήμα Δημοτ. Εκπαίδευσης και 1 από Οικονομικό Τμήμα. Έντεκα δήλωσαν ότι είχαν διδαχθεί Αντικειμενοστρεφή Προγραμματισμό κυρίως σε προπτυχιακό μάθημα ή σε σεμινάριο, εκ των οποίων οκτώ με την χρήση της γλώσσας Java. Συνολικά 22 άτομα δήλωσαν ότι είχαν διδαχθεί κάποια γλώσσα προγραμματισμού. Δύο κατέταξαν τον εαυτό τους στους Επαγγελματίες ως προς τον Προγραμματισμό, τέσσερις στους Πολύ έμπειρους, οκτώ στους Μέτρια έμπειρους, έξι στους Λίγο έμπειρους και οκτώ στους Αρχάριους.

### Μέθοδος - Διαδικασία

Το σεμινάριο πραγματοποιήθηκε στο διάστημα από 28/2/2005 έως 7/3/2005 και περιελάμβανε 4 τρίαωρα θεωρητικής παράδοσης σε συνδυασμό με εργαστήριο. Διεξήχθη σε εργαστηριακό χώρο και οι φοιτητές είχαν στη διάθεση τους εγκαταστημένα, τα

περιβάλλοντα που χρησιμοποιήθηκαν καθώς και όλο το μαθησιακό υλικό. Έγινε χρήση του εκπαιδευτικού προγραμματιστικού περιβάλλοντος Bluej, του Προγραμματιστικού Μικρόκοσμου Jeroo καθώς και του ολοκληρωμένου προγραμματιστικού περιβάλλοντος (IDE) SUN One Studio. Μετά την παρουσίαση εννοιών-θεμάτων με την χρήση παραδειγμάτων, δινόταν η ευκαιρία στους φοιτητές να τα δοκιμάσουν, καθώς τα αρχεία κώδικα Java, τα Bluej projects και τα αρχεία Jeroo, συμπεριλαμβάνονταν στο υλικό. Για το μαθησιακό υλικό που σχεδιάστηκε, δημιουργήθηκε δικτυακός τόπος στο περιβάλλον του Συστήματος Μαθησιακής Τεχνολογίας Moodle, το οποίο εκτός από μέσο διανομής μαθησιακού υλικού, χρησιμοποιήθηκε για να γίνονται συζητήσεις σχετικά με διάφορα θέματα του γνωστικού αντικείμενου και για να στέλλονται οι ασκήσεις. Στοιχεία για τον έλεγχο του βαθμού κατανόησης, τις δυσκολίες των φοιτητών και την δημιουργία παρανοήσεων, συγκεντρώθηκαν με την χρήση κατάλληλα σχεδιασμένων ερωτήσεων-ασκήσεων που οι φοιτητές παρέδιδαν σε συγκεκριμένες ημερομηνίες και από άμεση παρατήρηση και επαφή με τους φοιτητές κατά την διάρκεια του σεμιναρίου.

## ΑΠΟΤΕΛΕΣΜΑΤΑ

Κάποιοι φοιτητές αποφάνθηκαν για τον Προγραμματιστικό Μικρόκοσμο Jeroo, ότι πρόκειται για κάτι «παιδικό» και άλλοι έκαναν -με θετικό τρόπο- άμεση αναφορά σε άλλους προγραμματιστικούς μικρόκοσμους που γνώριζαν π.χ. Karel the Robot. Εκτιμάται ότι απαιτείται προσεκτικότερος σχεδιασμός παραδειγμάτων και συγκεκριμένων δραστηριοτήτων που θα κινητοποιούν τους φοιτητές στην χρήση του περιβάλλοντος.

Διαπιστώθηκε η ευκολία χρήσης του Bluej με αποτέλεσμα να το χρησιμοποιούν ως περιβάλλον ανάπτυξης κώδικα Java. Την τελευταία μέρα και ενώ ήδη είχαν ξεκινήσει να διδάσκονται στο ΠΜΣ μάθημα που περιελάμβανε Java, εκφράστηκαν λέγοντας: «Δυστυχώς, στο μάθημα, χρησιμοποιούμε την γραμμή εντολών και δεν έχουμε την δυνατότητα της οπτικής υπόστασης των πραγμάτων που μας παρέχει το Bluej». Επίσης αντιλήφθηκαν την εξαιρετική χρησιμότητά του για τον έλεγχο κλάσεων που σχεδιάζουμε, χωρίς να χρειάζεται να γράψουμε κώδικα.

Παραθέτουμε μερικά πρώτα συμπεράσματα που προέκυψαν από την κατ' αρχήν ανάλυση των εργασιών των φοιτητών.

### Δυσκολίες-παρανοήσεις που αναδείχθηκαν και προτάσεις αντιμετώπισης

- Οι φοιτητές αποδίδουν ως ιδιότητες οντοτήτων **και** ιδιότητες που μπορούν να προκύψουν από άλλες, π.χ. ιδιότητες Κύκλου: ακτίνα, **διάμετρος, εμβαδόν, περίμετρος**. Χρειάζεται, επιπλέον, έμφαση στο γεγονός ότι στις ιδιότητες περιλαμβάνονται αυτές που καθορίζουν μοναδικά την κατάσταση των αντικειμένων και όχι περιττές πληροφορίες και πληροφορίες που αποτελούν συνάρτηση άλλων.
- Δυσκολεύονται να αποδώσουν ως ιδιότητες οντοτήτων λογικές (boolean) μεταβλητές, ή να κατατάξουν στην κατηγορία ιδιότητα, εκφράσεις που μεταφράζονται σε boolean μεταβλητές. Για παράδειγμα, σε συγκεκριμένη

ερώτηση δεν θεώρησαν το «πεινάει ή όχι» ιδιότητα των οντοτήτων «Άνθρωπος» και «Σκύλος» αλλά αντίθετα το θεωρούν μέθοδο. Η παρανόηση αυτή μπορεί να αποδοθεί στη χρήση ρήματος που παραπέμπει σε ενέργεια (συμπεριφορά και άρα μέθοδο) και προτείνεται η χρήση του «είναι πεινασμένος ή όχι». Επίσης για παρόμοιο λόγο εκτιμάται ότι θεωρούν τα: ομιλία, σκέψη, ακοή, όραση, ιδιότητες και όχι μεθόδους της οντότητας «Άνθρωπος», εξαιτίας της χρήσης ουσιαστικού και όχι ρήματος για την περιγραφή τους.

Γενικά ο προσδιορισμός ιδιοτήτων σε οντότητες, αναδεικνύεται δύσκολο σημείο για τους φοιτητές. Σαν μέθοδος αντιμετώπισης προτείνεται η χρήση ειδικά σχεδιασμένων παραδειγμάτων οντοτήτων στα οποία έχουν αποδοθεί με λανθασμένο τρόπο ιδιότητες και η συζήτησή τους στην τάξη.

- Σε μικρό βαθμό, όταν χρειάζεται να προσδιορίσουν μεθόδους σε οντότητες περιορίζονται μόνον σε μεθόδους get και set.
- Κάποιοι φοιτητές συμπεριλαμβάνουν στην διαπροσωπεία της κλάσης και τον κώδικα των μεθόδων εκτός από την υπογραφή τους.
- Σε μικρό βαθμό συγχέουν τους τύπους αναφοράς που συνδέονται με τα αντικείμενα με τα ίδια τα αντικείμενα.
- Δυσκολεύονται να αντιληφθούν τους τύπους αναφοράς σε composed κλάσεις.
- «Οικοδομούν» λανθασμένα, την αντίληψη ότι όλες οι ιδιότητες μιας κλάσης πρέπει να είναι private και όλες οι μέθοδοί της public και τον εφαρμόζουν σε κάθε περίπτωση χωρίς να εξετάσουν την πιθανή ιδιαιτερότητα της περίπτωσης κάποιας ιδιότητας που δεν απαιτείται να είναι private, αλλά κυρίως μιας μεθόδου που επιβάλλεται να μην είναι public. Χρειάζεται να τονίζεται περισσότερο ότι ο πρώτος τρόπος σχεδιασμού δεν αποτελεί «κανόνα» και να παρουσιάζονται παραδείγματα που τον «παραβιάζουν».
- Σχηματίζουν την εντύπωση, ότι σε static variable μπορεί να γίνει αναφορά μόνο από static method.
- Ευκολότερα αποδίδουν τον προσδιοριστή ορατότητας public σε ένα ιδιοχαρακτηριστικό στο οποίο χρειάζεται να υπάρχει πρόσβαση, παρά σκέφτονται την υλοποίηση μεθόδου get για αυτό.

#### **Δυσκολίες που απαιτούν περαιτέρω αντιμετώπιση**

- Η σύγχυση των ιδιοτήτων ενός αντικειμένου με τα μέρη από τα οποία αποτελείται παρατηρήθηκε έντονα π.χ. Ιδιοχαρακτηριστικά του «Άνθρωπος»: πόδια, χέρια, κεφάλι - Ιδιοχαρακτηριστικά του «Αυτοκίνητο»: τιμόνι, ρόδες, γκάζι, φρένο.

#### **Αντιμετώπιση καταγεγραμμένων δυσκολιών**

Οι φοιτητές δεν παρουσίασαν αρκετές από τις καταγεγραμμένες δυσκολίες και παρανοήσεις, όπως:

- Δεν παρατηρήθηκε σύγχυση των ιδιοχαρακτηριστικών ενός αντικειμένου με τις ενέργειες που μπορεί να εκτελέσει.
- Λίγοι μόνο φοιτητές θεωρούν κλάση ένα σύνολο από αντικείμενα.

- Οι περισσότεροι δεν συγχέουν τις σχέσεις κλάσης-αντικειμένου και ιεραρχίας.
- Αντιλαμβάνονται σωστά την δημιουργία αντικειμένων είτε με την χρήση του `by default` κατασκευαστή είτε με κατασκευαστή ή κατασκευαστές που έχουν δηλωθεί στην κλάση. Επίσης κατανοούν ορθά την αρχικοποίηση των ιδιοχαρακτηριστικών των αντικειμένων με διάφορους τρόπους.
- Δεν θεωρούν δυο αντικείμενα με ακριβώς ίδιες τιμές στα ιδιοχαρακτηριστικά τους σαν ένα αντικείμενο.
- Δεν παρατηρήθηκε η παρανόηση ότι μόνο η ύπαρξη του κατασκευαστή αρκεί για την δημιουργία αντικειμένων.
- Δεν παρατηρήθηκε η ταύτιση αντικειμένου/μεταβλητής, κλάσης/αντικειμένου, η παρανόηση ότι όλες οι μεταβλητές στιγμιοτύπου μιας κλάσης είναι του ίδιου τύπου, η αντίληψη ότι τα αντικείμενα είναι απλές εγγραφές χωρίς συμπεριφορά, η παρανόηση ότι ο κατασκευαστής μπορεί να αντικατασταθεί από μεθόδους που δίνουν αρχικές τιμές στα ιδιοχαρακτηριστικά (αντιλαμβάνονται την ανάγκη εκτέλεσης εντολής με την χρήση του τελεστή `new` για την δημιουργία αντικειμένων).
- Δέχονται την χρήση του `dot operator` για την αναφορά σε ιδιότητες, αν ο προσδιοριστής ορατότητας το επιτρέπει.

#### Διαπιστώσεις

- Οι φοιτητές κατανοούν τις αποδεκτές μορφές υπερφορτωμένων μεθόδων και πώς «επιλύεται» το ποια από τις μεθόδους αυτές θα εκτελεστεί.
- Επίσης κατανοούν την μοναδικότητα μιας `static variable` για όλα τα αντικείμενα, ότι η αναφορά σε `instance variables` και η κλήση `instance methods` δεν γίνεται με το όνομα της κλάσης αλλά απαιτεί την δημιουργία αντικειμένων και ότι οι `static methods` δεν μπορούν να αναφέρονται σε `instance variables`.

#### Σχόλια και κριτική φοιτητών

Οι φοιτητές αξιολόγησαν θετικά τον τρόπο διδασκαλίας στο σεμινάριο. Εξέφρασαν ότι «ήταν πολύ ενδιαφέρον» και ότι «θα ήταν πολύ χρήσιμο για αυτούς αν είχε διεξαχθεί πριν καν αρχίσουν το Μεταπτυχιακό, διότι θα τους έλυne πολλά προβλήματα που αντιμετώπισαν σε μαθήματα π.χ. Ανάλυση και Σχεδιασμός Συστημάτων». Κατά την διάρκεια του τελευταίου μαθήματος ένας φοιτητής ρώτησε: «είχατε σχεδιάσει το υλικό με κάποιο εκπαιδευτικό-παιδαγωγικό τρόπο; Γιατί εγώ βλέπω ότι από την αρχή μέχρι το τέλος, είχε μια συνοχή που εμένα με βοήθησε. Όταν διαβάζεις ένα βιβλίο δεν συμβαίνει κάτι τέτοιο». Ένας Υποψήφιος Διδάκτορας που παρακολουθούσε γιατί στο πλαίσιο του Διδακτορικού του χρειαζόταν να χρησιμοποιήσει Java, αρκετά αργότερα από το τέλος του σεμιναρίου είπε ότι δεν θα μπορούσε να είχε κάνει αυτό που του χρειαζόταν αν δεν είχε παρακολουθήσει και αν δεν είχε χρησιμοποιήσει το υλικό του σεμιναρίου. Επίσης αρκετοί φοιτητές και αφού παρήλθε διάστημα από την λήξη του σεμιναρίου, εκτίμησαν ότι βοηθήθηκαν ιδιαίτερα παρακολουθώντας το. Αυτό συνέβη και για φοιτητές οι οποίοι δεν διέθεταν καμία απολύτως προηγούμενη γνώση Προγραμματισμού.

Η γνώση και η εμπειρία που αποκτήθηκε από την διεξαγωγή του σεμιναρίου αυτού, χρησιμοποιήθηκε για την περαιτέρω συμπλήρωση και βελτίωση του μαθησιακού υλικού, το οποίο ήδη χρησιμοποιείται σε μακράς διάρκειας σεμιναριακά μαθήματα που διεξάγονται για του προπτυχιακούς φοιτητές του Τμήματος ΔΤΨΣ του Πανεπιστημίου Πειραιά. Η μικρή διάρκεια του σεμιναρίου και η παράδοση των εργασιών εντός σύντομου χρονικού διαστήματος, επιβάλλει να γίνει παρακολούθηση των φοιτητών που συμμετείχαν και σε μεταγενέστερο χρόνο, πράγμα που αποτελεί μελλοντικό σχέδιό μας.

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Ευχαριστούμε τους Μεταπτυχιακούς φοιτητές του Τμήματος ΔΤΨΣ του Πανεπιστημίου Πειραιά, για την συμμετοχή τους στο σεμινάριο, την ανατροφοδότηση που μας προσέφεραν, τα εποικοδομητικά τους σχόλια και την κριτική τους.

## **ΒΙΒΛΙΟΓΡΑΦΙΑ**

- ACM (2001), *Computing Curricula*, <http://www.computer.org/education/cc2001/final>
- Bennedsen J. & Caspersen M. (2004), Teaching object-oriented programming. Towards teaching a systematic programming process, *18<sup>th</sup> European Conference on Object-Oriented Programming*, Oslo, Norway
- Fleury A. E. (2001), Encapsulation and reuse as viewed by java students, *ACM SIGCSE Bulletin*, 33(1), 189 – 193
- Fleury A. (2000), Programming in Java: Student-constructed rules, *SIGCSE Bulletin*, 32(1), 197-201
- Holland S., Griffiths R. & Woodman M. (1997), Avoiding object misconceptions, *Proceedings of the 28th SIGCSE*, 131-134
- Kölling M. & Rosenberg J. (2001), Guidelines for teaching object orientation with Java, *Proceedings of 6th IT iCSE*, 33-36 Canterbury
- Kölling M., Quig B, Patterson A. & Rosenberg J. (2003), The BlueJ system and its pedagogy, *Journal of Computer Science Education*, 13(4), 249-268
- Nygaard K. (2005), *A sufficiently complex example*, <http://www.intermedia.uio.no/cool/complex.htm>
- Ragonis N. & Ben-Ari M. (2002), Teaching constructors: A difficult multiple choice, *16<sup>th</sup> European Conference on Object-Oriented Programming*, Malaga, Spain
- Sanders D. & Dorn B. (2003), Jeroo: a tool for introducing object-oriented programming, *ACM SIGCSE Bulletin*, 35(1), 201-204
- Satratzemi M., Xinogalos S. & Dagdilelis V. (2003), An environment for teaching object-oriented programming: ObjectKarel, *Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies (ICALT'03)*, 342-343
- Teif M. & Hazzan O., (2004), Junior high school students' perception of object oriented concepts, *18<sup>th</sup> European Conference on Object-Oriented Programming*, Oslo, Norway