ePublishing
ΕΘΝΙΚΟ ΚΕΝΤΡΟ ΤΕΚΜΗΡΙΩΣΗΣ
NATIONAL DOCUMENTATION CENTRE

Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση

Τόμ. 1 (2008)

4ο Συνέδριο Διδακτική Πληροφορικής

Συγκρίνοντας και Αξιολογώντας τα Εκπαιδευτικά Εργαλεία και Προγραμματιστικά Περιβάλλοντα για τον Αντικειμενοστρεφή Προγραμματισμό

Σ. Γεωργαντάκη, Σ. Ρετάλης, Ι. Ψαρομήλιγκος, Χ. Κυτάγιας

Βιβλιογραφική αναφορά:

Γεωργαντάκη Σ., Ρετάλης Σ., Ψαρομήλιγκος Ι., & Κυτάγιας Χ. (2023). Συγκρίνοντας και Αξιολογώντας τα Εκπαιδευτικά Εργαλεία και Προγραμματιστικά Περιβάλλοντα για τον Αντικειμενοστρεφή Προγραμματισμό. *Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση*, *1*, 101–110. ανακτήθηκε από https://eproceedings.epublishing.ekt.gr/index.php/cetpe/article/view/5855

Συγκρίνοντας και Αξιολογώντας τα Εκπαιδευτικά Εργαλεία και Προγραμματιστικά Περιβάλλοντα για τον Αντικειμενοστρεφή Προγραμματισμό

Σ. Γεωργαντάκη[1], Σ. Ρετάλης[1], Ι. Ψαρομήλιγκος[2], Χ. Κυτάγιας[2]

[1]Τμήμα Διδακτικής της Τεχνολογίας και Ψηφιακών Συστημάτων, Πανεπιστήμιο Πειραιά
Εργαστήριο Προηγμένων Τεχνολογιών Μάθησης και Πολιτισμού - CoSy-LLab
{rgeo; retal}@unipi.gr
[2]ΤΕΙ Πειραιά
{jpsa;ckyt}@teipir.gr)

Περίληψη

Στη βιβλιογραφία ανακύπτει η ανάγκη για την ανάπτυξη εξειδικευμένων εκπαιδευτικών εργαλείων για την υποστήριξη της διδασκαλίας του Αντικειμενοστρεφούς Προγραμματισμού (ΑΠ). Τα εκπαιδευτικά εργαλεία και προγραμματιστικά περιβάλλοντα αυτά είναι αναμφίβολα λιγότερα και λιγότερο ώριμα από τα αντίστοιχα για το διαδικασιακό παράδειγμα προγραμματισμού (procedural programming paradigm). Η παρούσα εργασία αποσκοπεί στην παράθεση των πιο διαδεδομένων εργαλείων που έχουν αναπτυχθεί για την υποστήριξη της μαθησιακής διαδικασίας του ΑΠ. Τα εργαλεία και περιβάλλοντα αυτά συγκρίνονται σύμφωνα με εξειδικευμένα κριτήρια όπως οπτικοποίηση (visualization) των κλάσεων και των αντικειμένων, οπτικοποίηση της δομής του προγράμματος με UML, κ.α. Από τη συγκριτική μελέτη βρέθηκε ότι το εργαλείο BlueJ υπερτερεί έναντι άλλων. Προς επίρρωση αυτού του πορίσματος, διεξήχθη μία πιλοτική μελέτη αξιολόγησης του εργαλείου από 18 φοιτητές σε αυθεντικό περιβάλλον χρήσης στο πλαίσιο διαδικτυακού μαθήματος (elearning course) στο πρόγραμμα μεταπτυχιακών σπουδών του Τ.Ε.Ι. Πειραιά.

Λέξεις κλειδιά: *Αντικειμενοστρεφής Προγραμματισμός. εκπαιδευτικά εργαλεία, προγραμματιστικά περιβάλλοντα*

Abstract

Throughout bibliography it is noticeable that there is a lack of appropriate tools for the support of teaching object-orientated programming. This paper aims to present the most popular of these tools, and to evaluate them according to specialized criteria, such as: class and object visualization, visualization of the program structure etc. Bluej was found to be most usable one. To confirm these findings, 18 postgraduate students from the technological institution of Piraeus evaluated BlueJ during an e-learning course. We demonstrate the most important findings.

Keywords: *Object oriented programming, educational tools, programming environments*

1. Η διδασκαλία του Αντικειμενοστρεφούς Προγραμματισμού

Στη βιβλιογραφία εμφανίζονται ερευνητικές μελέτες για τη δυσκολία εκμάθησης του Αντικειμενοστρεφούς Προγραμματισμού (ΑΠ) (Holland et al., 1997; Milne and Rowe, 2002; Teif and Hazzan, 2004; Ragonis and Ben-Ari, 2005; Lahtinen et al., 2005). Σε κάποιες από τις έρευνες διατυπώνονται συστάσεις διδασκαλίας για την αποφυγή των δυσκολιών και παρανοήσεων (Fleury, 2001) ή οδηγίες (Kölling and Rosenberg, 2001). Επίσης έχουν διατυπωθεί μια σειρά από παιδαγωγικά σχεδιαστικά πρότυπα (pedagogical patterns) που αφορούν στη διδασκαλία (Bergin, 2006).

Συνάγεται, επομένως, το πόρισμα ότι για την αποτελεσματική διδασκαλία του ΑΠ απαιτείται η ανάπτυξη συγκεκριμένων διδακτικών προσεγγίσεων με εκπαιδευτικές αρχές και φιλοσοφία που θα αξιοποιούν τα αποτελέσματα των ερευνητικών μελετών που έχουν ήδη πραγματοποιηθεί στο χώρο αυτό. Επίσης, ανακύπτει η ανάγκη για την ανάπτυξη εξειδικευμένων εκπαιδευτικών εργαλείων.

Όμως τα εκπαιδευτικά εργαλεία και προγραμματιστικά περιβάλλοντα αυτά είναι αναμφίβολα λιγότερα και λιγότερο ώριμα από τα αντίστοιχα για το διαδικασιακό παράδειγμα προγραμματισμού (procedural programming paradigm) (Kölling and Rosenberg, 2001). Η παρούσα εργασία αποσκοπεί στην παράθεση των πιο διαδεδομένων εκπαιδευτικών εργαλείων και εκπαιδευτικών προγραμματιστικών περιβαλλόντων που έχουν αναπτυχθεί για την υποστήριξη της μαθησιακής διαδικασίας του ΑΠ. Τα εργαλεία και περιβάλλοντα αυτά συγκρίνονται σύμφωνα με εξειδικευμένα κριτήρια που έχουν προταθεί στο (McIver, 2002), όπως οπτικοποίηση (visualization) των κλάσεων και των αντικειμένων, υποστήριξη της μοντελοποίησης (modeling), οπτικοποίηση της δομής του προγράμματος με UML ή UML-like διαγράμματα, οπτικοποίηση της δυναμικής εκτέλεσης του προγράμματος, δυνατότητα αποσφαλμάτωσης (debugging), κ.α. Από τη συγκριτική μελέτη βρέθηκε ότι το εργαλείο BlueJ υπερτερεί έναντι άλλων. Προς επίρρωση αυτού του πορίσματος, διεξήχθη μία πιλοτική μελέτη αξιολόγησης του εργαλείου από 18 φοιτητές σε αυθεντικό περιβάλλον χρήσης στο πλαίσιο διαδικτυακού μαθήματος (elearning course) στο πρόγραμμα μεταπτυχιακών σπουδών του Τ.Ε.Ι. Πειραιά. Στην εργασία παρουσιάζονται τα κύρια ευρήματα αυτής της μελέτης.

2. Προγραμματιστικοί μικρόκοσμοι

Οι προγραμματιστικοί μικρόκοσμοι (programming microworlds) αναπτύσσονται για καθαρά εκπαιδευτικούς σκοπούς, βασίζονται σε μια φυσική μεταφορά (metaphor) (χελώνα, ρομπότ, καγκουρό, τρισδιάστατα αντικείμενα σε εικονικούς κόσμους) – δηλαδή πρωταγωνιστές που "ζουν" μέσα στο περιβάλλον του μικρόκοσμου–, είναι εύχρηστοι, ενσωματώνουν δυνατότητες οπτικοποίησης και κίνησης (animation), συνήθως χρησιμοποιούν μια εκπαιδευτική γλώσσα προγραμματισμού και παρέχουν τη δυνατότητα της δυναμικής προσομοίωσης της εκτέλεσης των προγραμμάτων,

δηλαδή της βήμα προς βήμα εκτέλεσης του και συγχρόνως απεικόνιση του αποτελέσματος της εκτέλεσης στην κατάσταση του μικρόκοσμου. Τα πιο γνωστά περιβάλλοντα είναι:

Ο **KarelJRobot** αποτελεί τη μετάφραση που ο Bergin έκανε στο μικρόκοσμο Karel++ σε αμιγώς γλώσσα Java, δηλαδή ένα αντικειμενοστρεφές Java περιβάλλον (Bergin et al., 2003). Πρόσφατα εκδόθηκε βιβλίο με παραδείγματα και ασκήσεις γι' αυτόν το μικρόκοσμο (Bergin et al., 2005) (http://csis.pace.edu/~bergin/KarelJava2ed/karelexperimental.html).

Ο **JKarelRobot** (http://math.otterbein.edu/JKarelRobot/) είναι ένας μικρόκοσμος γραμμένος σε Java, ο οποίος υποστηρίζει διαγράμματα ροής (flow charts) και τη δυνατότητα διαγραμματικής παρουσίασης του προγράμματος με τη χρήση διαγραμμάτων δομής (control structure diagrams, CSD) (Buck and Stucki, 2001).

Ο προγραμματιστικός μικρόκοσμος **Jeroo** (http://www.jeroo.org/) (Sanders and Dorn, 2003) χρησιμοποιεί ως μεταφορέα ένα σπάνιο είδος θηλαστικού παρόμοιο με τα μικρόσωμα καγκουρό της Αυστραλίας. Από κατασκευής το περιβάλλον εστιάζει στις έννοιες "αντικείμενο" και "μέθοδος" και στις δομές ελέγχου (control structures). Υποστηρίζει δύο στυλ γλωσσών προγραμματισμού, ένα Java/C++/C# και ένα VB.NET στυλ. Επίσης διαθέτει οπτικοποίηση της κίνησης των αντικειμένων, οπτικοποίηση της εκτέλεσης του κώδικα, παρέχει τη δυνατότητα εκτέλεσης βήμα προς βήμα και συνεχόμενα και επιλογής ταχύτητας εκτέλεσης.

Το εργαλείο λογισμικού **Alice** (http://www.alice.org) είναι ένα τρισδιάστατο διαδραστικό περιβάλλον απεικόνισης (animation) στο οποίο οπτικοποιούνται τα αντικείμενα και οι συμπεριφορές τους (Cooper et al., 2003). Η τριών διαστάσεων απεικόνιση "ενισχύει" την οπτικοποίηση των αντικειμένων παρέχοντας μια αίσθηση πραγματικότητας γι' αυτά και παρέχει ένα ευέλικτο και εποικοδομητικό πλαίσιο για την κατανόηση των αντικειμενοστρεφών εννοιών. Ουσιαστικά ακολουθεί την παράδοση των μικρόκοσμων Karel παρέχοντας μια επιπλέον διάσταση. Οι τριών διαστάσεων κόσμοι είναι περισσότερο ρεαλιστικοί από τους αντίστοιχους δισδιάστατους.

Ο προγραμματιστικός μικρόκοσμος **objectKarel**, αποτελεί ένα αξιόλογο εκπαιδευτικό περιβάλλον το οποίο βασίζεται στον Karel++ (Xinogalos et al., 2006,) (http://csis.pace.edu/~bergin/temp/findkarel.html). Σημαντικές του πρωτοτυπίες αποτελούν, η ύπαρξη του εκδότη δομής (structure editor) για τη δημιουργία των προγραμμάτων, η δυνατότητα της επεξηγηματικής οπτικοποίησης (δηλαδή της εμφάνισης επεξηγήσεων για την τρέχουσα εντολή), η σειρά μαθημάτων (e-lessons) που διαθέτει στα ελληνικά και αγγλικά και των δραστηριοτήτων για την εξοικείωση των σπουδαστών με τις βασικές αντικειμενοστρεφείς έννοιες και τις δομές ελέγχου και η δυνατότητα καταγραφής των ενεργειών των σπουδαστών καθώς δημιουργούν προγράμματα (καταγραφή των προγραμμάτων και όλων των μεταγλωττίσεών τους) για τον προσδιορισμό δυσκολιών και παρανοήσεών τους.

3. Εκπαιδευτικά προγραμματιστικά περιβάλλοντα

Τα πιο διαδεδομένα εκπαιδευτικά προγραμματιστικά περιβάλλοντα για τη διδασκαλία του Αντικειμενοστρεφούς παραδείγματος Προγραμματισμού που χρησιμοποιούν τη γλώσσα Java είναι:

Το ***BlueJ*** (http://www.bluej.org/) (Kölling et al., 2003) αποτελεί ένα πολύ αξιόλογο εκπαιδευτικό προγραμματιστικό περιβάλλον για την υιοθέτηση της "object first" προσέγγισης στη διδασκαλία. Σημαντικά χαρακτηριστικά του είναι: α) η ευχρηστία του σε σχέση με άλλα ολοκληρωμένα προγραμματιστικά περιβάλλοντα (IDEs), β) η δυνατότητα να αντιληφθείς ότι μια εφαρμογή είναι ένα σύνολο από συνεργαζόμενες κλάσεις, γ) η οπτική υπόσταση των κλάσεις και των αντικειμένων, δ) η αλληλεπίδραση του εκπαιδευόμενου με τις κλάσεις δημιουργώντας αντικείμενα (μπορεί να "δει" τα αντικείμενα, να ελέγξει (inspect) τις τιμές των ιδιοχαρακτηριστικών τους (attributes), να αλληλεπιδράσει με αυτά καλώντας και εκτελώντας μεθόδους και να παρατηρήσει και ελέγξει τα αποτελέσματα της εκτέλεσής τους), ε) η ευκολία στο πέρασμα από το οπτικό περιβάλλον στο περιβάλλον κώδικα καθώς και η εύκολη μεταγλώττιση με επισήμανση των λαθών κατευθείαν στο συντάκτη του κώδικα.

Το **DrJava** (http://drjava.sourceforge.net/) αποτελεί ένα παιδαγωγικό προγραμματιστικό περιβάλλον Java το οποίο εισάγει ομαλά τους σπουδαστές στη συγγραφή προγραμμάτων (Allen et al., 2002). Επιτρέπει την ανάπτυξη, τον έλεγχο και την αποσφαλμάτωση προγραμμάτων Java με διαδραστικό και επαυξητικό τρόπο. Υποστηρίζει τον κύκλο "read-eval-print loop" (read, evaluate, print, REPL) ο οποίος και διευκολύνει την επαυξητική ανάπτυξη των προγραμμάτων. Ουσιαστικά αποτελεί έναν διερμηνευτή Java (Java interpreter) που επιτρέπει διαδραστικό έλεγχο της εκτέλεσης εντολών και αλληλεπίδραση σε επίπεδο αντικειμένου, μέσω ενός διαδραστικού παραθύρου στο οποίο οι σπουδαστές μπορούν να εισάγουν εκφράσεις και εντολές Java και να δουν άμεσα τα αποτελέσματά τους χωρίς την ανάγκη μεθόδου main() και της χρήσης ενός εξωτερικού μεταγλωττιστή. Παρόμοιο περιβάλλον, είναι και το **BeanShell** (http://www.beanshell.org/).

Το **ProfessorJ** (Proulx and Gray, 2006) (http://www.professorj.org/), αποτελεί ένα προγραμματιστικό περιβάλλον για την Java, που διαθέτει επίπεδα παρουσίασης των δομών της γλώσσας (Beginner, Intermediate, Advanced), παρέχοντας ένα απλοποιημένο ενδιάμεσο χρήστη στο μεταφραστή Java και το σύστημα εκτέλεσης Java. Είναι επέκταση του προγραμματιστικού περιβάλλοντος DrScheme της γλώσσας Scheme, αλλά αποτελεί περιβάλλον Java. Κάθε επίπεδο υποστηρίζει ένα υποσύνολο της γλώσσας. Με αυτό τον τρόπο οι σπουδαστές εισάγονται σταδιακά στις συντακτικές και σημασιολογικές λεπτομέρειες της γλώσσας, έχοντας στη διάθεσή τους ένα περιβάλλον το οποίο ακολουθεί την πρόοδο του μαθήματος που

παρακολουθούν, που είναι μια βαθμιαία έκθεση σε αυξανόμενης πολυπλοκότητας δομές.

Το περιβάλλον **JPie** (Java Programmer's Interactive Environment) (http://jpie.cse.wustl.edu/) είναι ένα εκπαιδευτικό ολοκληρωμένο προγραμματιστικό περιβάλλον σχεδιασμένο να κάνει προσιτή σε ευρύτερο "κοινό" τη δύναμη της αντικειμενοστρεφούς ανάπτυξης λογισμικού. Είναι υλοποιημένο σε Java και είναι συμβατό με μια ποικιλία από πλατφόρμες. Αναπαριστά τα προγράμματα γραφικά και όχι με κείμενο. Υποστηρίζει τη "ζωντανή" κατασκευή κώδικα σε Java με οπτικό τρόπο, μέσω άμεσου χειρισμού γραφικών αναπαραστάσεων προγραμματιστικών εννοιών (αφαιρέσεων) και δομών (όπως κλάσεις, υποκλάσεις, μεταβλητές στιγμιοτύπου (type και scope), μέθοδοι, event handler μέθοδοι κ.ά.). Πολλές λειτουργίες εκτελούνται με τη μέθοδο "drag and drop". Η δυνατότητα χειρισμού εννοιών γραφικά και όχι ως μια γραμμική σειρά χαρακτήρων, σε συνδυασμό με την υποστήριξη "ζωντανής" μεταβολής του λογισμικού παρέχει έναν φυσικότερο τρόπο ανάπτυξης λογισμικού και αυξάνει το επίπεδο της αφαίρεσης.

Για λόγους καθαρά οικονομίας χώρου αναφέρουμε μόνο ονομαστικά και τα παρακάτω εκπαιδευτικά προγραμματιστικά περιβάλλοντα που είναι τα: **Javiva**, **iContract** και **JML** (Turner and Zachary, 2001), **GiniPad** (http://www.mokabyte.it/ginipad/english.htm),**JGrasp**(http://www.eng.auburn.edu/grasp/), **jCreatorLE**(http://www.jcreator.com/index.htm),**D-ChK** (Depradine and Gay, 2004), **greenfoot** (http://www.greenfoot.org) και **P-coder** (Roy and Armarego, 2004).

4. Συγκριτική παρουσίαση των κύριων εκπαιδευτικών εργαλείων και προγραμματιστικών περιβαλλόντων

Στον Πίνακα 1 δίνεται μια συνολική συγκριτική παρουσίαση των κύριων εκπαιδευτικών εργαλείων και προγραμματιστικών περιβαλλόντων που προαναφέρθηκαν με βάση τα σημαντικά χαρακτηριστικά που διαθέτουν ή όχι και την ωριμότητα χρήσης τους. Ως βασικά χαρακτηριστικά που θα πρέπει να ληφθούν υπόψη στη συγκριτική παρουσίαση των εργαλείων-περιβαλλόντων θεωρήθηκαν τα: απλότητα (simplicity) του περιβάλλοντος, δυνατότητα αλληλεπίδρασης (interaction) (δημιουργία αντικειμένων και κλήση μεθόδων επί αυτών), οπτικοποίηση (visualization) των κλάσεων και των αντικειμένων, υποστήριξη της μοντελοποίησης (modeling), οπτικοποίηση της δομής του προγράμματος με UML ή UML-like διαγράμματα, οπτικοποίηση της δυναμικής εκτέλεσης του προγράμματος, δυνατότητα δημιουργίας skeletons κώδικα από τις γραφικές αναπαραστάσεις των κλάσεων, ύπαρξη γραφικού ενδιαμέσου, πλήρης υποστήριξη της γλώσσας προγραμματισμού Java, εύκολη μεταγλώττιση (compilation), επισήμανση της γραμμής κώδικα με συντακτικό λάθος, δυνατότητα αποσφαλμάτωσης (debugging). Τα περισσότερα από τα παραπάνω αναφέρονται ως βασικές κατηγορίες χαρακτηριστικών οι οποίες θα πρέπει να λαμβάνονται υπόψη κατά τη συγκριτική

ανάλυση εκπαιδευτικών προγραμματιστικών περιβαλλόντων στο McIver (2002). Από τα περιβάλλοντα-εκπροσώπους του προγραμματιστικού μικρόκοσμου Karel the Robot, στον πίνακα έχει περιληφθεί η υλοποίηση του JKarelRobot.

Από τη συγκριτική παρουσίαση, προκύπτει η υπεροχή του περιβάλλοντος BlueJ, σε χαρακτηριστικά και ωριμότητα χρήσης (Hagan and Markham 2000; Van Haaster and Hagan, 2004). Αναδεικνύεται η έλλειψη του χαρακτηριστικού της δυναμικής προσομοίωσης εκτέλεσης προγραμμάτων από τα εκπαιδευτικά προγραμματιστικά περιβάλλοντα. Το περιβάλλον greenfoot που ενσωματώνει επιπλέον αυτό το χαρακτηριστικό αποτελεί ένα περιβάλλον σχεδιασμένο για εκπαιδευόμενους σχολικής ηλικίας, που σε άλλη βαθμίδα εκπαίδευσης η χρήση του ίσως χαρακτηριζόταν "παιδική" και επίσης δεν υπάρχουν γι' αυτό αποτελέσματα χρήσης στη διδακτική πράξη. Του περιβάλλοντος BlueJ ακολουθεί το jGrasp, αλλά δεν έχει δοκιμαστεί και δεν υποστηρίζει πλήρως την οπτικοποίηση ούτε τη σχεδίαση μοντέλων.

Από τους προγραμματιστικούς μικρόκοσμους ο Jeroo διαθέτει αρκετά σημαντικά χαρακτηριστικά, αλλά δεν προσφέρει τη δυνατότητα της πλήρους παρουσίασης της φιλοσοφίας του ΑΠ, αφού από κατασκευής δεν υποστηρίζει κληρονομικότητα και οι Jeroo γλώσσες υποστηρίζουν μόνο μια κλάση και τη δημιουργία μέχρι τεσσάρων αντικειμένων από αυτήν. Σημαντικό χαρακτηριστικό αποτελεί η συνάφεια της γλώσσας τύπου JAVA/C++/C# του περιβάλλοντος, με τη Java. Οι μικρόκοσμοι Alice και objectKarel ακολουθούν σε χαρακτηριστικά. Σχετικά με τον objectKarel, αποτελεί σαφώς ένα πληρέστατο τέτοιου είδους περιβάλλον (προγραμματιστικός μικρόκοσμος) για την παρουσίαση της αντικειμενοστρέφειας και των βασικών εννοιών-αρχών της, με τα πολύ σημαντικά χαρακτηριστικά του εκδότη δομής (structure editor) τον οποίο διαθέτει και ο οποίος ελαχιστοποιεί την πιθανότητα συντακτικών λαθών και των μαθημάτων που ενσωματώνει κατά ενότητες.

5. Πιλοτική μελέτη αξιολόγησης του εργαλείου BlueJ

Έχοντας δεδομένο ότι το εργαλείο BlueJ χαρακτηρίζεται ως το πλέον κατάλληλο για την υποστήριξη της διδασκαλίας του ΑΠ, μια μελέτη προς επίρρωση αυτού του δεδομένου διεξήχθη κατά το εαρινό εξάμηνο του 2006 στο πλαίσιο διαδικτυακού (elearning) μεταπτυχιακού μαθήματος το οποίο διήρκεσε δυόμισι μήνες στο Τ.Ε.Ι. Πειραιά. Λεπτομέρειες για τη μελέτη και τη διδακτική προσέγγιση υπάρχουν στο (Georgantaki et al., 2007). Συμμετείχαν δέκα οκτώ (18) μεταπτυχιακοί φοιτητές, 17 άνδρες και 1 γυναίκα, ηλικίας από 23 έως 52 ετών, οι οποίοι προέρχονταν από τμήματα Τ.Ε.Ι. (14 άτομα) και τμήματα Α.Ε.Ι. (4 άτομα). Το κίνητρο των φοιτητών για τη συμμετοχή τους στο μάθημα, ήταν κυρίως το ενδιαφέρον τους για το αντικείμενο του ΑΠ, η βελτίωση των γνώσεων και των δεξιοτήτων τους στον ΑΠ και έπειτα το ότι τους χρειαζόταν στο πλαίσιο των μεταπτυχιακών τους σπουδών. Κατά τη διάρκεια του μαθήματος πραγματοποιήθηκαν τρεις (3) συνεδρίες συντονισμού και

επίλυσης αποριών. Οι συνεδρίες αυτές έγιναν σε εργαστήριο και δόθηκε η δυνατότητα στους φοιτητές να εξασκηθούν και να εξοικειωθούν με τα προγραμματιστικά περιβάλλοντα BlueJ και SUN One Studio που κλήθηκαν να χρησιμοποιήσουν για τις ανάγκες του μαθήματος.

Σύμφωνα με το εκπαιδευτικό σενάριο, οι εκπαιδευόμενοι αφού παρατήρησαν (μέσω ηλεκτρονικών σημειώσεων) πώς ο εκπαιδευτής –ο οποίος έπαιζε το ρόλο ενός ειδικού στην ανάπτυξη αντικειμενοστρεφούς λογισμικού– ανέλυσε, σχεδίασε, υλοποίησε, αποσφαλμάτωσε και εκτέλεσε μια απλή αντικειμενοστρεφή εφαρμογή με τη βοήθεια BlueJ, κλήθηκαν να κάνουν το ίδιο για τις ανάγκες μίας ατομικής εργασίας. Εργασία έπρεπε να παραδοθεί σε επιμέρους παραδοτέα τα οποία είναι τα αποτελέσματα ενεργειών ανά φάση ανάπτυξης (ανάλυση-σχεδίαση-υλοποίηση).

Ανάμεσα στα θέματα που ερευνήθηκαν στην πιλοτική αυτή μελέτη ήταν η αξιολόγηση της ευχρηστίας του εκπαιδευτικού προγραμματιστικού περιβάλλοντος BlueJ. Οι απαντήσεις των φοιτητών σε ένα ερωτηματολόγιο σχετικά με τα σημαντικά χαρακτηριστικά του περιβάλλοντος BlueJ, φαίνονται στον Πίνακα 2.

Όπως αναμένονταν, τα "δυνατά" χαρακτηριστικά του BlueJ, ήταν η οπτικοποίηση των κλάσεων και των αντικειμένων, η απλότητά του και η δυνατότητα αλληλεπίδρασης που προσφέρει (Kölling et al., 2003).

Πίνακας 2: *Αξιολόγηση των σημαντικών χαρακτηριστικών του BlueJ*

Ποια από τα παρακάτω χαρακτηριστικά του BlueJ σας βοήθησαν στη μαθησιακή διαδικασία; Επιλέξτε αυτά που νομίζετε.	**Αριθμός φοιτητών (max 18)**
Οι κλάσεις και τα αντικείμενα αποκτούν οπτική υπόσταση.	12
Εύκολη δημιουργία αντικειμένων και οπτικοποίησης τους.	12
Ευκολία χρήσης του περιβάλλοντος.	11
Εύκολος έλεγχος των τιμών των ιδιοχαρακτηριστικών των αντικειμένων.	9
Επιτρέπεται να αλληλεπιδράς απευθείας με τις κλάσεις και τα αντικείμενα.	9
Εύκολος πειραματισμός με άμεσο αποτέλεσμα (feedback).	8
Εύκολη εκτέλεση των μεθόδων και έλεγχός τους (εύκολο πέρασμα παραμέτρων και επιστροφή αποτελέσματος), χωρίς να χρειάζεται να γράψεις κώδικα γιαυτό το σκοπό	8
Εύκολο compilation με επισήμανση των συντακτικών λαθών απευθείας στο παράθυρο του editor με μήνυμα και επισήμανση της γραμμής κώδικα με το λάθος.	8
Διαθέτει όψη "implementation" και όψη "interface" για τις κλάσεις.	4
Διαθέτει debugger εύκολο στη χρήση.	5

6. Συμπεράσματα

Είναι γεγονός ότι η εκμάθηση του ΑΠ παρουσιάζει δυσκολίες (Satratzemi et al. 2003). Για την αντιμετώπιση των δυσκολιών έχουν αναπτυχθεί εκπαιδευτικά προγραμματιστικά περιβαλλόντων. Το Bluej αποτελεί ένα πολύ εύχρηστο περιβάλλον το οποίο συμβάλλει αποτελεσματικά στην υποστήριξη της διδασκαλίας ανάπτυξης εφαρμογών σε Java. Ανάμεσα στα σημαντικά χαρακτηριστικά του Bluej,

τα οποία αναφέρονται τόσο στη βιβλιογραφία όσο και στα πορίσματα μίας μελέτης αξιολόγησης που παρουσιάστηκε στην εργασία αυτή, είναι ότι παρέχει τη δυνατότητα στους εκπαιδευόμενους να αντιληφθούν ότι μια εφαρμογή είναι ένα σύνολο από συνεργαζόμενες κλάσεις, ότι είναι εύκολο το πέρασμα από το οπτικό περιβάλλον στο περιβάλλον κώδικα, κ.α. Μεγάλο πλεονέκτημα του εργαλείου είναι ότι είναι επεκτάσιμο και ήδη μία σειρά από επισυναπτόμενες οντότητες (plug-in modules) αναπτύσσονται.

Βιβλιογραφία

Allen E., Cartwright R., and Stoler B. (2002). DrJava: A lightweight pedagogic environment for Java. *ACM SIGCSE Bulletin, 34*(1), 137-141.

Bergin, J. (2006). *Some Pedagogical Patterns*. Διαθέσιμο στην ηλεκτρονική διεύθυνση http://csis.pace.edu/~bergin/patterns/fewpedpats.html.

Bergin, J., Stehlik, M., Roberts, J. and Pattis, R. (2005). *Karel J. Robot: A Gentle Introduction to the Art of Object-Oriented Programming in Java*. USA: Dream Songs Press.

Bergin, J., Stehlik, M., Roberts, J., and Pattis, R. (2003). *Karel J. Robot: A Gentle Introduction to the Art of Object-Oriented Programming in Java.* Published manuscript. Διαθέσιμο στην ηλεκτρονική διεύθυνση http://csis.pace.edu/~bergin/KarelJava2ed/Karel++JavaEdition.html.

Buck, D., and Stucki, D.J. (2001). JKarelRobot: A Case Study in Supporting Levels of Cognitive Development in the Computer Science Curriculum. *ACM SIGCSE Bulletin, 33*(1), 16-20.

Cooper, S., Dann, W., and Pausch, R. (2003). Teaching Objects-first in Introductory Computer Science. *ACM SIGCSE Bulletin 35*(1), 191-195.

Depradine, C., and Gay, G. (2004). Active participation of integrated development environments in the teaching of object-oriented programming. *Computers & Education, 43*(3), 291-298.

Fleury, A. E. (2001). Encapsulation and reuse as viewed by java students. *ACM SIGCSE Bulletin, 33*(1), 189-193.

Georgantaki, S., Psaromiligkos, Y., Retalis, S., Dendrinos, V., and Adamopoulos, D. (2007). Developing a blended learning strategy for teaching Object-Oriented Programming using the "Model First" approach*,, Proceedings of Informatics Education Europe II Conference 2007 (IEEII2007)*, 29-30 November 2007, Thessaloniki, Greece, pp. 87-96.

Hagan, D. and Markham, S. (2000). Teaching Java with the BlueJ Environment. Paper presented at the *Australian Society for Computers in Learning in Tertiary Education (ASCILITE),* Coffs Harbour, Australia.

Holland, S., Griffiths, R., and Woodman, M. (1997). Avoiding object misconceptions. In *Proceedings of the 28th SIGCSE,* 131-134.

Kölling, M., Quig, B., Patterson, A., and Rosenberg, J. (2003). The BlueJ system and its pedagogy. *Journal of Computer Science Education, Special Issue on Learning and Teaching Object Technology, 13*(4), 249-268.

Kölling, M., and Rosenberg, J. (2001). Guidelines for Teaching Object Orientation with Java. *ACM SIGCSE Bulletin 33*(3), 33-36.

Lahtinen, E., Ala-Mutka, K., and Järvinen, H.-M. (2005). A study of the difficulties of novice programmers**.** In *Proceedings of the 10th Annul SIGCSE conference on Innovation and Technology in Computer Science Education,* 14–18, June 27-29, Monte de Caparica, Portugal, also published in *ACM SIGCSE Bulletin,37*(3), 14-18.

McIver, L. (2002). Evaluating Languages and Environments for Novice Programmers. *14th Annual Workshop of the Psychology of Programming Interest Group (PPIG 2002),* Middlesex, UK, 100-110.

Milne, J., and Rowe, G. (2002). Difficulties in Learning and Teaching Programming Views of Students and Tutors. *Education and Information Technologies,7*(1),55-66.

Proulx, V., and Gray, K. E. (2006). Design of Class Hierarchies: An Introduction to OO Program Design. *ACM SIGCSE Bulletin 38*(1), 288-292.

Ragonis, N., and Ben-Ari, M. (2005). A Long-Term Investigation of the Comprehension of OOP Concepts by Novices. *Computer Science Education, 5*(3), 203-221.

Roy, G.G., and Armarego, J. (2004). Teaching programming with objects. *ASCILITE (Australasian Society for Computers in Learning in Tertiary Education) Conference*, Perth, Western Australia. Διαθέσιμο στην ηλεκτρονική διεύθυνση http://www.ascilite.org.au/conferences/perth04/procs/roy.html.

Sanders, D., and 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*, 342-343.

Teif, M., and Hazzan, O. (2004). Junior High School Students' Perception of Object Oriented Concepts. Paper presented at *18th European Conference on Object-Oriented Programming, 8th Workshop on Pedagogies and Tools for the Teaching and Learning of Object Oriented Concepts*, Oslo, Norway.

Turner, T. A., and Zachary, L. (2001). Javiva: ATool for Visualizing and Validating Student-Written Java Programs. In *Proceedings of the 32nd SIGCSE Technical Symposium on Computer Science Education,* Charlotte, February, 45-49.

Van Haaster, K., and Hagan, D. (2004). Teaching and Learning with BlueJ: an Evaluation of a Pedagogical Tool. *Information Science & Information Technology Education Joint Conference,* Rockhampton, QLD, Australia, 456-470.

Xinogalos, S., Satratzemi, M., and Vassilios Dagdilelis, V. (2006). An introduction to object-oriented programming with a didactic microworld: objectKarel. *Computers & Education, 47*(2), 148-171.

Πίνακας 1: Συγκριτική παρουσίαση Προγραμματιστικών Μικρόκοσμων-Εκπαιδευτικών Προγραμματιστικών περιβαλλόντων για τη διδασκαλία του ΑΠ

		Προγραμματιστικοί Μικρόκοσμοι				**Εκπαιδευτικά Προγραμματιστικά Περιβάλλοντα**										
		JKarelRobot	Jeroo	Alice	objectKarel	BlueJ	DrJava	ProfessorJ	JPie	Javiva	GiniPad	jGrasp	jCreator	D-Chk	greenfoot	P-Coder
Σημαντικά χαρακτη-ιστικά	Απλότητα	✓	✓	✓	✓	✓	✓	✓	✓	✓				✓	✓	
	Δυνατότητα αλληλεπίδρασης					✓	✓		✓			✓			✓	✓
	Οπτικοποίηση		μα[2]	μα[1]	μα[1]	✓			γαε[3]	αοα[4]	δδκιμ[5]	αμμκ[6]	δδκιμ[4]		✓	
	Μοντελοποίηση					✓									✓	
	Οπτικοποίηση δομής προγράμματος					✓						✓			✓	
	Οπτικοποίηση δυναμικής εκτέλεσης προγράμματος	✓	✓	✓	✓										✓	
	Δημιουργία skeletons κώδικα					✓									✓	
	γραφικό ενδιάμεσο	✓	✓	✓	✓	✓	✓				✓	✓	✓	✓	✓	
	Πλήρης υποστήριξη Java					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	Εύκολο compilation		✓		✓	✓	✓	✓			✓	✓	✓	✓	✓	
	Επισήμανση γραμμής με συντακτικό λάθος		✓		✓	✓	✓				✓	✓	✓	✓	✓	
	αποσφαλματωτής (debugger)	✓				✓	✓					✓			✓	
Ωριμότητα χρήσης	Μικρή		✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Αρκετή - Μεγάλη	✓				✓										

[2] μα = μόνο οπτικοποίηση αντικειμένων
[3] γαε = γραφικές αναπαραστάσεις εννοιών
[4] αοα = αφηρημένες οπτικοποιήσεις αντικειμένων
[5] δδκιμ = δενδρική δομή κλάσεων – ιδιοτήτων - μεθόδων
[6] αμμκ = αντικείμενα με μορφή κειμένου