

# Πρόταση Διδασκαλίας της Αντικειμενοστραφούς Προσέγγισης σε Αρχάριους Προγραμματιστές με το Περιβάλλον AgentSheets

**Χαράλαμπος Σπυρόπουλος, Βασίλειος Δαγδιλέλης**

Τμήμα Εκπαιδευτικής & Κοινωνικής Πολιτικής, Πανεπιστήμιο Μακεδονίας  
[chspyrop@sch.gr](mailto:chspyrop@sch.gr), [dagdil@uom.gr](mailto:dagdil@uom.gr)

## ΠΕΡΙΛΗΨΗ

Στα εισαγωγικά μαθήματα προγραμματισμού, συχνά χρησιμοποιούνται γλώσσες όπως η Java και η Visual Basic, γλώσσες δηλαδή που έχουν μεγάλη διάδοση ανάμεσα στους επαγγελματίες. Οι γλώσσες αυτές έχουν βέβαια το πλεονέκτημα ότι είναι αυτές που θα χρησιμοποιήσουν οι φοιτητές στην αγορά εργασίας, ωστόσο δημιουργούν στους αρχάριους προγραμματιστές προβλήματα κατανόησης, όπως τουλάχιστον δείχνουν οι σχετικές έρευνες, λόγω της σχετικής τους πολυπλοκότητας και του αντίστοιχου προγραμματιστικού περιβάλλοντος τους. Επιπλέον τα σύγχρονα μοντέλα προγραμματισμού, όπως ο αντικειμενοστραφής, θέτουν και μια σειρά πρόσθετων διανοητικών εμποδίων στους αρχάριους. Πολλοί ερευνητές έχουν προτείνει ιδιαίτερα περιβάλλοντα, για την αντιμετώπιση αυτών των προβλημάτων, ειδικά σχεδιασμένων για την (εισαγωγική κυρίως) διδασκαλία. Στην παρούσα εργασία προτείνεται το προγραμματιστικό περιβάλλον AgentSheets, για μια εισαγωγή στον αντικειμενοστραφή προγραμματισμό. Το προγραμματιστικό περιβάλλον AgentSheets, μπορεί να επιδράσει θετικά στην ανάπτυξη της αντικειμενοστραφούς προγραμματιστικής σκέψης σε αρχάριους προγραμματιστές. Τα οπτικά και απτικά χαρακτηριστικά που ενσωματώνει καθώς και οι δυνατότητες αλληλεπίδρασης και ανατροφοδότησης σε όλες τις φάσεις ανάπτυξης και εκτέλεσης των προγραμμάτων, σε συνδυασμό με τη δυνατότητα γρήγορης υλοποίησης παραδειγμάτων μπορούν να επιδράσουν θετικά, στην ανάπτυξη της αλγοριθμικής σκέψης και της αντικειμενοστραφούς λογικής και επομένως το περιβάλλον AgentSheets φαίνεται να έχει ευρείες εκπαιδευτικές εφαρμογές.

**ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ:** *Objects-First, Αντικειμενοστραφής προγραμματισμός, αρχάριοι προγραμματιστές*

## ΕΙΣΑΓΩΓΗ

Η στροφή που πραγματοποιείται την τελευταία δεκαετία στην ανάπτυξη λογισμικού με την χρήση αντικειμενοστραφών γλωσσών προγραμματισμού, έχει οδηγήσει πολλά ακαδημαϊκά ιδρύματα στην υιοθέτηση της προσέγγισης 'Objects-First' στην εισαγωγική διδασκαλία του προγραμματισμού (ACM 2001). Μάλιστα σε πολλές χώρες το

αντικειμενοστραφές μοντέλο προγραμματισμού έχει υιοθετηθεί και στη Δευτεροβάθμια Εκπαίδευση (π.χ. Ελλάδα) σε κάποιο βαθμό. Ωστόσο, φαίνεται ότι η υιοθέτηση αυτού του μοντέλου και η χρήση αντικειμενοστραφών γλωσσών δημιουργεί στους αρχάριους προγραμματιστές σημαντικά προβλήματα κατανόησης και μάθησης όπως αδυναμία χρήσης των εννοιών πριν την υλοποίησή τους, η ταύτιση κλάσης και αντικειμένων, πέρασμα μηνυμάτων, σύγχυση ταυτότητας/χαρακτηριστικών κ.α. (Holland et al. 1997). Η χρήση επαγγελματικών γλωσσών προγραμματισμού (Java, Visual Basic, C++, C#), αν και έχουν το πλεονέκτημα ότι είναι αυτές που θα χρησιμοποιήσουν οι φοιτητές στην αγορά εργασίας (Allen 1996), δημιουργούν στους αρχάριους προγραμματιστές, εκτός των κλασικών προβλημάτων ως επαγγελματικά περιβάλλοντα, όπως, η εκτεταμένη γλώσσα, ο ανύπαρκτος διδακτικός προσανατολισμός καθώς και φύση των εφαρμογών που μπορεί κανείς να υλοποιήσει στα πρώτα του βήματα (Brusilovsky et al. 1997) και μια σειρά από διδακτικά εμπόδια, που έχουν να κάνουν με την αντικειμενοστρέφεια, όταν αυτή αποτελεί το μοντέλο της πρώτης προσέγγισης στον προγραμματισμό. «Μικρο-Γλώσσες» (Brusilovsky et al. 1997) όπως Logo (Papert 1980), Karel (Pattis 1981), Marta (Calabrese 1989), υποσύνολα γλωσσών όπως η Blue J (Kölling & Rosenberg 1996) καθώς και εκπαιδευτικά περιβάλλοντα όπως το *MicroWorlds Pro*® έχουν προταθεί από πολλούς ερευνητές στην αντιμετώπιση αυτών των προβλημάτων.

Στην πρώτη ενότητα, αναφέρονται τα προβλήματα που συνήθως αντιμετωπίζονται κατά τη διδασκαλία του αντικειμενοστραφούς προγραμματισμού και στην επόμενη ενότητα, παρουσιάζεται η φιλοσοφία των «Μικρο-γλωσσών» προγραμματισμού σε σχέση με την αντικειμενοστραφή προσέγγιση. Στη συνέχεια γίνεται εκτενής αναφορά στο περιβάλλον AgentSheets και στα αντικειμενοστραφή χαρακτηριστικά του όπως επίσης και παρουσίαση της διδακτικής πρότασης με τη χρήση του AgentSheets για την εισαγωγή στον αντικειμενοστραφή προγραμματισμό. Τέλος συνοψίζοντας περιγράφονται τα μελλοντικά ερευνητικά μας σχέδια.

## **ΠΡΟΒΛΗΜΑΤΑ ΔΙΔΑΣΚΑΛΙΑΣ ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΟΥΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ**

Πολλά από τα προαναφερθέντα προβλήματα κατανόησης και μάθησης που έχουν αναφερθεί, προέρχονται από κάποια γενικότερα εμπόδια που προκύπτουν, όταν ο αντικειμενοστραφής προγραμματισμός αποτελεί το μοντέλο της πρώτης προσέγγισης στον προγραμματισμό.

- Στα πρώτα μαθήματα που είναι και αυτά που εισάγουν τους σπουδαστές στις βασικές έννοιες του προγραμματισμού, χρησιμοποιούνται παραδείγματα περιορισμένης πολυπλοκότητας που δεν αναδεικνύουν τα οφέλη της αντικειμενοστραφούς σχεδίασης και ανάπτυξης προγραμμάτων όπως επίσης δεν σχετίζονται άμεσα με τα γενικότερα ενδιαφέροντα τους.
- Προϋποθέτουν την εκμάθηση βασικών προγραμματιστικών δομών και μια πληθώρα χαρακτηριστικών και δυνατοτήτων της γλώσσας πριν την επιτυχημένη ανάπτυξη αντικειμένων (Cooper, Dann & Pausch 2003).

- Η Οπτική αναπαράσταση και παρακολούθηση των αντικειμένων είναι μια αρκετά σύνθετη διαδικασία που απαιτεί προηγούμενη γνώση προγραμματισμού και διαχείρισης του περιβάλλοντος επαφής (Proulx, Raab & Rasala 2002).

Η προσπάθεια λοιπόν κατανόησης και σωστής έκφρασης με μια επαγγελματική γλώσσα προγραμματισμού στην περίπτωση των αρχάριων προγραμματιστών, οδηγεί στο να εστιάζουν οι φοιτητές περισσότερο στην σύνταξη της γλώσσας και πολύ λιγότερο στην σχεδίαση και την ανάπτυξη δεξιοτήτων αντικειμενοστραφούς σχεδίασης και αλγοριθμικής σκέψης.

### **«ΜΙΚΡΟ-ΓΛΩΣΣΕΣ» ΚΑΙ ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ**

Οι περισσότερες από τις μικρο-γλώσσες που έχουν κατασκευαστεί και έχουν εφαρμοστεί μέχρι σήμερα εστιάζουν στον Διαδικαστικό ή Συναρτησιακό προγραμματισμό. Ορισμένες από αυτές στις νέες τους εκδόσεις υποστηρίζουν χαρακτηριστικά και δομές του αντικειμενοστραφούς προγραμματισμού όπως η StarLogo (Resnick 1994), ο Karel++ (Bergin et al. 1997) και το MicroWorlds Pro®, χωρίς όμως να είναι αμιγώς αντικειμενοστραφή. Μια νέα γενιά, αμιγώς αντικειμενοστραφών περιβαλλόντων, έχουν αναπτυχθεί τα τελευταία χρόνια και βρίσκονται στο στάδιο της ερευνητικής τους εφαρμογής και χρήσης ως εργαλεία ανάπτυξης της προγραμματιστικής σκέψης και της εισαγωγής στον προγραμματισμό. Χαρακτηριστικά παραδείγματα τέτοιων περιβαλλόντων είναι ο Objectkarel (Xinogalos 2003), η ALICE (Pausch et al. 1995) που χρησιμοποιώντας μια πολύ πλούσια σε δομές και εντολές γλώσσα αντικειμενοστραφούς προγραμματισμού και ένα τρισδιάστατο εικονικό κόσμο που απευθύνεται κυρίως σε αρχάριους προγραμματιστές της τριτοβάθμιας εκπαίδευσης και το AgentSheets (Repenning 1995). Η δημιουργία προσομοιώσεων στο περιβάλλον AgentSheets έχει ως κύριο στόχο την ανάπτυξη της αντικειμενοστραφούς προγραμματιστικής σκέψης και λιγότερο την εξοικείωση με τις εμπορικές γλώσσες προγραμματισμού. Αν και οι γλώσσες προγραμματισμού που χρησιμοποιούν τα περιβάλλοντα αυτά, αποτελούν συνήθως υποσύνολο μιας εμπορικής γλώσσας προγραμματισμού και δεν υποστηρίζουν πάντα το σύνολο των προγραμματιστικών δομών, εντούτοις θα πρέπει να διαθέτουν χαρακτηριστικά τέτοια που να πλησιάζουν όσο το δυνατόν περισσότερο τα θεμελιώδη χαρακτηριστικά του προγραμματιστικού μοντέλου που στοχεύουν. Τα θεμελιώδη χαρακτηριστικά του αντικειμενοστραφούς προγραμματισμού όπως αυτά διατυπώθηκαν από τον Kay (1993) είναι:

- Τα πάντα είναι αντικείμενα.
- Κάθε είδους υπολογισμοί διεκπεραιώνονται από τα αντικείμενα τα οποία επικοινωνούν μεταξύ τους διατυπώνοντας αιτήματα μέσω μηνυμάτων.
- Κάθε αντικείμενο έχει την δική του μνήμη, η οποία αποτελείται από άλλα αντικείμενα.

- Κάθε αντικείμενο είναι ένα παράδειγμα (instance) μίας κλάσης αντικειμένων. Η κλάση αυτή προσδιορίζει τα χαρακτηριστικά ενός συνόλου ομοειδών αντικειμένων.
- Στις κλάσεις περιγράφονται οι συμπεριφορές των αντικειμένων που ανήκουν σε αυτή την κλάση. Έτσι κάθε αντικείμενο που ανήκει σε μία κλάση μπορεί να εκτελέσει αυτή τη συμπεριφορά.
- Οι κλάσεις είναι οργανωμένες σε μια δενδροειδή ιεραρχική δομή που καθορίζει και την κληρονομικότητα των χαρακτηριστικών από τις γονικές κλάσεις.

Αυτά τα χαρακτηριστικά σκιαγραφούν τα χαρακτηριστικά μιας αντικειμενοστραφούς γλώσσας προγραμματισμού, που βέβαια, επεκτείνονται και αναλύονται, σε μια πληθώρα επιπλέον χαρακτηριστικών, τα οποία επίσης διαφοροποιούνται ανάλογα, με την γλώσσα προγραμματισμού που χρησιμοποιείται.

Μια μικρο-γλώσσα θα πρέπει να ενσωματώνει και να αναδεικνύει με όσο το δυνατόν ποιο ξεκάθαρο τρόπο τα παραπάνω χαρακτηριστικά για να μπορεί να λειτουργήσει ως εργαλείο ανάπτυξης της αντικειμενοστραφούς σκέψης και να επιτρέψει την ομαλή και χωρίς παρανοήσεις μετάβαση σε μια εμπορική γλώσσα προγραμματισμού.

## **ΤΟ ΠΕΡΙΒΑΛΛΟΝ AGENTSHEETS ΚΑΙ ΤΑ ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ**

Το περιβάλλον AgentSheets είναι ένα εργαλείο δημιουργίας προσομοιώσεων πολλαπλών εφαρμογών, που επιτρέπει σε μη προγραμματιστές να δημιουργήσουν τις δικές τους προσομοιώσεις και παιχνίδια. Είναι ουσιαστικά ένα περιβάλλον οπτικού προγραμματισμού το οποίο συνδυάζει πράκτορες (agents), υπολογιστικά φύλλα (spreadsheets) και εργαλεία συγγραφής Java σε ένα ενιαίο περιβάλλον. Οι Πράκτορες παρουσιάζονται ως μικρές κινούμενες εικόνες στην οθόνη του υπολογιστή και έχουν συσχετιζόμενες συμπεριφορές που καθορίζουν την αλληλεπίδρασή τους με το περιβάλλον τους. Οι συμπεριφορές αυτές ορίζονται από τον τελικό χρήστη, επιτρέποντας έτσι σε χρήστες χωρίς προγραμματιστικό υπόβαθρο να δημιουργήσουν πράκτορες με σύνθετες συμπεριφορές (Ioanidou, Spyropoulos & Viglas 2005).

Ο προγραμματισμός στο AgentSheets πραγματοποιείται με την χρήση της γλώσσας προγραμματισμού Visual AgenTalk (VAT). Τα δομικά στοιχεία της γλώσσας καθώς και ο τρόπος που αναπτύσσονται οι προσομοιώσεις από την οπτική γωνία της υιοθέτησης των βασικών χαρακτηριστικών της αντικειμενοστρέφειας είναι:

### **ΔΟΜΙΚΑ ΣΤΟΙΧΕΙΑ ΤΗΣ ΓΛΩΣΣΑΣ VAT**

Σύμφωνα με την διατύπωση του Alan C. Kay (Kay 1993), τα δομικά στοιχεία της γλώσσας VAT που ακολουθούν, αντίστοιχα αυτούς τους κανόνες είναι:

**Εικονοθήκη Πρακτόρων:** Η συλλογή των κλάσεων (πρακτόρων) που δημιουργούνται από τον προγραμματιστή, είναι οι μοναδικές οντότητες της γλώσσας, βάση των οποίων δημιουργούνται και όλα τα αντικείμενα που μπορούν να υπάρξουν και να προγραμματιστούν στις προσομοιώσεις που δημιουργούνται. Δεν υπάρχει λοιπόν τίποτα

άλλο εκτός από αντικείμενα που δημιουργήθηκαν σύμφωνα με την κλάση που έχει περιγράψει ο προγραμματιστής.

**Συμπεριφορά Αντικειμένων:** Οι οποιοιδήποτε υπολογισμοί, υλοποιούνται από τα αντικείμενα και δεν είναι δυνατόν να υπάρξει κώδικας που να μην ανήκει στη συμπεριφορά της τάξης ενός αντικείμενου. Επίσης τα αντικείμενα, έχουν τη δυνατότητα, με ένα σύνολο εντολών όπως ΕΚΠΕΜΠΩ, ΚΑΝΩ, ΑΚΟΥΣΩ, ΣΤΟ ΑΚΟΥΣΜΑ, να διατυπώνουν αιτήματα μέσω μηνυμάτων. Όλα τα αντικείμενα που έχουν δημιουργηθεί από την ίδια κλάση μπορούν να εκτελέσουν την ίδια συμπεριφορά.

**Μνήμη αντικειμένων:** Εκτός των κλασικών μεταβλητών, τοπικού και καθολικού χαρακτήρα που μπορούν να αναφέρονται τα αντικείμενα, μπορούν μέσω οπτικής διαδικασίας, κατά το προγραμματιστικό στάδιο, να αναφέρονται και σε αντικείμενα ή κλάσεις αντικειμένων.

**Δημιουργία αντικειμένων:** Τα αντικείμενα δημιουργούνται είτε δυναμικά – κατά τη διάρκεια εκτέλεσης του κώδικα μέσω της εντολής ΔΗΜΙΟΥΡΓΩ από κάποιο άλλο αντικείμενο – ή από τον χρήστη, πριν ή και κατά τη διάρκεια εκτέλεσης του προγράμματος. Και στις δύο περιπτώσεις πρέπει να είναι προσδιορισμένη η κλάση βάσει της οποίας θα δημιουργηθεί το αντικείμενο.

**Ιεραρχική Δομή:** Η γλώσσα επιτρέπει την ιεραρχική δομή ενός επιπέδου, για την κληρονομήση των γονικών χαρακτηριστικών, και η μόνη διαφοροποίηση που επιτρέπει είναι η γραφική απεικόνιση της νέας κλάσης.

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

Εκτός από τα χαρακτηριστικά της γλώσσας, σημαντικό είναι το περιβάλλον διεπαφής και τα εργαλεία που προσφέρει για την ανάπτυξη των εφαρμογών να είναι προσαρμοσμένα στην ίδια αντικειμενοστραφή λογική.

**Αντικειμενοστραφής Ανάπτυξη:** Στο περιβάλλον ανάπτυξης προγραμμάτων του AgentSheets τα πάντα είναι αντικείμενα. Οι κλάσεις των αντικειμένων δεν είναι προκαθορισμένες, ορίζονται από τον τελικό χρήστη, ο οποίος είναι υπεύθυνος τόσο για την απεικόνισή τους, όσο και για τον καθορισμό της συμπεριφοράς τους. Οι διάφορες μορφές που δύναται να πάρει ένα αντικείμενο της κλάσης αυτής σχεδιάζεται σε ένα μικρο-σχεδιαστικό πρόγραμμα ή αντλούνται από εξωτερικές πηγές. Η συμπεριφορά της κλάσης οικοδομείται με την χρήση της γλώσσας προγραμματισμού VAT. Κάθε κλάση αντικείμενου έχει τις δικές της συμπεριφορές που είναι οργανωμένες σε μεθόδους, χαρακτηρίζονται από έναν Ενεργοποιητή και αποτελούνται από κανόνες της μορφής «Εάν .. τότε..», στους οποίους τοποθετούνται από τον προγραμματιστή οι κατάλληλες συνθήκες και δράσεις. Τα ίδια τα δομικά στοιχεία με τη σειρά τους, ως αντικείμενα και αυτά, μπορούν να παραμετροποιηθούν με οπτικό τρόπο αλλάζοντας έτσι την συμπεριφορά τους.

**Αντικειμενοστραφής Δομή γλώσσας:** Τα δομικά στοιχεία είναι Ενεργοποιητές, Συνθήκες και Δράσεις που αντλούνται από έτοιμες συλλογές. Ακόμα και τα δομικά στοιχεία της γλώσσας είναι αντικείμενα στην επιφάνεια διεπαφής. Η παροχή των αντικειμένων της γλώσσας (συνθήκες και δράσεις) στην επιφάνεια διεπαφής του

χρήστη, είναι το χαρακτηριστικό που αποδίδει την έννοια του «Απτικού» Προγραμματισμού στη γλώσσα VAT. Η έννοια της "αφής" χρησιμοποιείται εδώ όπως χρησιμοποιείται από τον Papert (Papert 1993) για να εξηγήσει, το βαθμό οικειότητας του προγραμματιστή σε σχέση με τα υπολογιστικά του αντικείμενα. (Repenning & Ambach 1996, Repenning & Ioannidou 1997). Με αφετηρία τον ορισμό του Papert, στη VAT, η έννοια των υπολογιστικών αντικειμένων στην γλώσσα δεν περιορίζεται στα αντικείμενα που προγραμματίζονται, δηλαδή τους πράκτορες, αλλά ισχύει και για τα ίδια τα συστατικά της γλώσσας προγραμματισμού (δράσεις, συνθήκες, κανόνες), τα οποία έχουν προαχθεί σε αντικείμενα ανώτερου επιπέδου, με εύκολο χειρισμό. Ο Απτικός Προγραμματισμός επιχειρεί να κάνει τον προγραμματισμό πιο προσιτό στον τελικό χρήστη, προσθέτοντας την αντίληψη του χειρισμού στην οπτική αντίληψη. Στον Απτικό Προγραμματισμό, τα προγράμματα δεν είναι στατικές αναπαραστάσεις, ούτε δίνουν την αίσθηση του απλού χειρισμού και της διόρθωσης του κώδικα. Οι αρχές και τα προγράμματα του Απτικού Προγραμματισμού δεν προάγουν μόνο τις οπτικές αναπαραστάσεις, που βοηθούν στην αναγνωσιμότητα και στην κατανόηση των προγραμμάτων, αλλά είναι δυναμικά και περιλαμβάνουν χειρισμούς, όπως ο καθορισμός παραμέτρων με τη χρήση οπτικών πεδίων (Σχήμα 1).



**Σχήμα 1:** Καθορισμός παραμέτρων με τη χρήση οπτικών πεδίων

**Αντικειμενοστραφής Εκσφαλμάτωση:** Στο περιβάλλον του AgentSheets αποκλείονται τα συντακτικά λάθη, αφού δεν λαμβάνει χώρα η διαδικασία της πληκτρολόγησης κώδικα και έτσι ο προγραμματιστής έχει να ανησυχεί μόνο για τα λογικά λάθη. Τα εργαλεία εκσφαλμάτωσης που παρέχει το περιβάλλον είναι, ο έλεγχος μιας δράσης, μιας συνθήκης, ενός κανόνα ακόμα και μιας ολόκληρης μεθόδου απευθείας πάνω σε ένα αντικείμενο με άμεση οπτική και ηχητική ανατροφοδότηση στην πορεία εκτέλεσης του κώδικα. Επίσης η βηματική εκτέλεση μιας προσομοίωσης είναι άλλο ένα εργαλείο που βοηθά στην εκσφαλμάτωση.

**Αντικειμενοστραφής Τεκμηρίωση:** Σε οποιοδήποτε στάδιο ανάπτυξης της προσομοίωσης, το περιβάλλον δίνει τη δυνατότητα στον προγραμματιστή για αυτόματη παραγωγή της πλήρους τεκμηρίωσης του κώδικα σε HTML μορφή, όπου περιγράφονται όλες οι κλάσεις με τις απεικονίσεις τους και οι αντίστοιχες συμπεριφορές που έχουν καθοριστεί σε μορφή ψευδοκώδικα.

## ΑΠΟ ΤΗ ΣΧΕΔΙΑΣΗ ΣΤΗ ΔΙΔΑΚΤΙΚΗ ΠΡΑΞΗ

Στη διδακτική προσέγγιση «Objects-First», υπάρχει μια αντικειμενική διδακτική δυσκολία στον ορισμό και την περιγραφή της αντικειμενοστραφούς λογικής ως μοντέλου σκέψης και επίλυσης προβλημάτων. Σε αυτή την προσπάθεια πολλά από τα βιβλία προσπαθούν να εισάγουν τον φοιτητή στις θεμελιώδεις έννοιες της αντικειμενοστραφούς σκέψης, χρησιμοποιώντας παραδείγματα από τον πραγματικό κόσμο, πολύ κοντά στην καθημερινότητα. Με τη χρήση τέτοιων παραδειγμάτων που εκμεταλλεύονται την προηγούμενη γνώση και εμπειρία των φοιτητών γίνεται πολύ πιο εύκολα η οικοδόμηση νέων δομών αναπαραστάσεων και η πραγματοποίηση εννοιολογικών αλλαγών (Κόμης 2005). Πολλές φορές όμως τα παραδείγματα αυτά είναι αδύνατον να υλοποιηθούν προγραμματιστικά, τουλάχιστον στα πρώτα προγραμματιστικά βήματα, αφού η υλοποίησή τους προϋποθέτει εκτεταμένη γνώση και εμπειρία σε μια γλώσσα προγραμματισμού. Ένα τέτοιο παράδειγμα που έχει χρησιμοποιηθεί από βιβλία εισαγωγής στον προγραμματισμό είναι το παράδειγμα της παρακολούθησης τηλεόρασης. Προερχόμενο από την καθημερινότητα των φοιτητών, μπορεί να αναδείξει με τον καλύτερο τρόπο τις βασικές αρχές του αντικειμενοστραφούς μοντέλου. Η τηλεόραση αποτελεί ένα αντικείμενο που ενώ δεν μας ενδιαφέρουν τα εσωτερικά της χαρακτηριστικά γνωρίζουμε ότι υποστηρίζει ένα σύνολο από προκαθορισμένες συμπεριφορές. Το ίδιο συμβαίνει και με το χειριστήριο της τηλεόρασης που αποτελεί επίσης ένα αντικείμενο με προκαθορισμένες συμπεριφορές. Τα βασικά σημεία που μπορούν να τονιστούν και να αναπτυχθούν σε ένα τέτοιο παράδειγμα είναι:

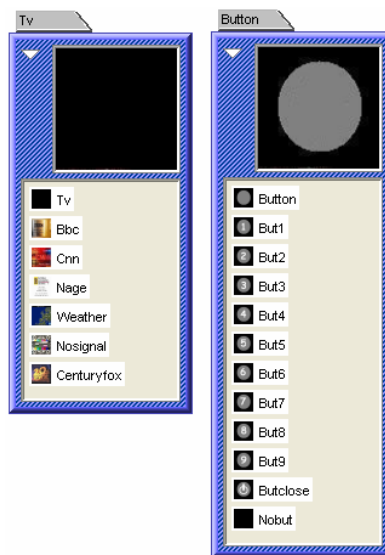
- Η επιλογή του καναλιού είναι κάτι που υποστηρίζει εσωτερικά η τηλεόραση εφόσον όμως δεχθεί το κατάλληλο αίτημα από κάποιο άλλο αντικείμενο που είναι το χειριστήριο. Το αίτημα εκτελείται λαμβάνοντας υπόψη και την πιθανή του παράμετρο που μπορεί να είναι ο αριθμός του καναλιού ή απλά το προηγούμενο-επόμενο κανάλι.
- Το ίδιο συμβαίνει και με το ήχο, τα αιτήματα στα οποία αποκρίνεται η τηλεόραση είναι η αυξομείωση της έντασης, η ενεργοποίηση ή απενεργοποίηση του ήχου, η αλλαγή του ηχητικού φίλτρου. Όλα αυτά είναι προκαθορισμένες συμπεριφορές της συγκεκριμένης τηλεόρασης.

Επεκτείνοντας και αναλύοντας τέτοιου είδους παραδείγματα, έχουμε την δυνατότητα να περιγράψουμε αρκετά από τα βασικά χαρακτηριστικά του αντικειμενοστραφούς μοντέλου. Η υλοποίηση όμως του παραδείγματος και η ανάπτυξη του προγράμματος προσομοίωσης του στον υπολογιστή, δίδει την ιδανική και ολοκληρωμένη αίσθηση και αντίληψη στους φοιτητές, του τι στην πραγματικότητα σημαίνει, αναπτύσσω αντικειμενοστραφή προγράμματα. Με τη γλώσσα VAT του AgentSheets δίδεται η δυνατότητα στους φοιτητές, να υλοποιήσουν και να δουν να δουλεύει, ένα τέτοιου είδους παράδειγμα, σε πολύ σύντομο χρονικό διάστημα. Στη συνέχεια η αλλαγή και η διατύπωση διαφορετικών σεναρίων τους εμπλέκει σε μια διαδικασία μετατροπής και επαναπροσδιορισμού, μέσα από την οποία και τις έννοιες της αντικειμενοστρέφειας κατανοούν, αλλά και βασικές προγραμματιστικές έννοιες, όπως ο κύκλος ζωής των

προγραμμάτων, μεταβλητές τοπικές και καθολικές, σταθερές, εκτέλεση κατά συνθήκη κ.α.

Η υλοποίηση στο AgentSheets ακολουθεί τα παρακάτω στάδια:

**Καθορισμός των αντικειμένων και της απεικόνισής τους:** Στο παραπάνω παράδειγμα, τα αντικείμενα είναι η τηλεόραση και το χειριστήριο. Οι διαφορετικές τους απεικονίσεις είναι η εικόνα των διαφορετικών καναλιών που προβάλει η οθόνη για το αντικείμενο της τηλεόρασης, και για αντικείμενο του χειριστηρίου, τα διαφορετικά κουμπιά που θα υπάρχουν πάνω σε αυτό (Σχήμα 2). Οι απεικονίσεις αυτές είτε σχεδιάζονται σε μια μικρο-σχεδιαστική εφαρμογή μέσα στο ίδιο το περιβάλλον ή αντλούνται από εικόνες που βρίσκονται στον υπολογιστή ή ακόμα και στο Διαδίκτυο.



**Σχήμα 2:** Καθορισμός των αντικειμένων και της απεικόνισής τους

**Καθορισμός της συμπεριφοράς των αντικειμένων:** Η συμπεριφορά του κάθε αντικειμένου καθορίζεται κάνοντας χρήση των δομών υψηλού επιπέδου που παρέχει η γλώσσα VAT, όπως ΣΤΟ ΑΚΟΥΣΜΑ (μήνυμα), ΕΞΕΠΕΜΨΕ (μήνυμα), ΑΛΛΑΖΩ (μορφή), ΗΧΩ (ήχος) και όλα αυτά μέσα σε προτάσεις ΑΝ...ΤΟΤΕ... (Σχήμα 3).

Σε κάθε στάδιο της ανάπτυξης, ο σπουδαστής, έχει τη δυνατότητα να εξετάσει την επίδραση του κάθε κανόνα πάνω σε αντικείμενα που τοποθετεί πάνω στο πεδίο εργασίας, ώστε να έχει άμεση ανατροφοδότηση, ελέγχοντας και επαναπροσδιορίζοντας τον τρόπο που αναπτύσσει την κάθε συμπεριφορά. Τέλος με την εκτέλεση της



προσομοίωσης (Σχήμα 4) ελέγχει το συνολικό αποτέλεσμα και πως τα διάφορα αντικείμενα αλληλεπιδρούν μεταξύ τους αλλά και με τον χρήστη της προσομοίωσης.



**Σχήμα 3:** Καθορισμός της συμπεριφοράς των αντικειμένων



**Σχήμα 4:** Εκτέλεση της προσομοίωσης

Με την ολοκλήρωση της προσομοίωσης μπορούν να προταθούν διάφορα σενάρια μεταβολής του πεδίου εργασίας όπως η προσθήκη ενός ακόμα αντικειμένου της ίδιας κλάσης, για παράδειγμα άλλη μια τηλεόραση που αντιδρά ταυτόχρονα στα αιτήματα του ίδιου χειριστήριου, αυτόματη αναζήτηση και προγραμματισμός καναλιών κ.α. αναδεικνύοντας έτσι μέσα από πειραματικές διαδικασίες που υλοποιούν οι ίδιοι οι μαθητές θεμελιώδεις προγραμματιστικές έννοιες του αντικειμενοστραφούς προγραμματισμού.

Τα πλεονεκτήματα της χρήσης του AgentSheets στην υλοποίηση ενός τέτοιου παραδείγματος είναι:

- Οπτικοποίηση του παραδείγματος
- Δυνατότητα στους μαθητές να υλοποιήσουν ιδέες από τα πρώτα κιάλας μαθήματα και να νιώσουν την χαρά της δημιουργίας.
- Δυνατότητα πειραματισμού και βιωματικής εμπειρίας.

## **ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ**

Εκτιμάται ότι το προγραμματιστικό περιβάλλον του AgentSheets, μπορεί να επιδράσει θετικά στην ανάπτυξη της αντικειμενοστραφούς προγραμματιστικής σκέψης σε αρχάριους προγραμματιστές. Πέρα από τα οφέλη που παρέχει η διαδικασία της μοντελοποίησης και των προσομοιώσεων, σε διάφορες γνωστικές περιοχές, μπορεί να βοηθήσει στα εισαγωγικά προγραμματιστικά μαθήματα και να λειτουργήσει ως μοχλός ανάπτυξης προγραμματιστικών δεξιοτήτων, ιδιαίτερα σε αρχάριους προγραμματιστές. Με την οπτικοποίηση σε όλα τα επίπεδα, την άμεση ανατροφοδότηση κατά την διαδικασία ανάπτυξης προγραμμάτων καθώς και με τη δυνατότητα επίλυσης και προσομοίωσης προβλημάτων υψηλής πολυπλοκότητας, μπορεί να ενθαρρύνει και να ενθουσιάζει τους νέους προγραμματιστές ταυτόχρονα με την ανάπτυξη των βασικών προγραμματιστικών τους δεξιοτήτων. Το πεδίο διερεύνησης σχετίζεται με το, αν το AgentSheets μπορεί να λειτουργήσει ως περιβάλλον για τη διδασκαλία του προγραμματισμού και να αποτελέσει την «Αλγοριθμοχώρα» κατά αντιστοιχία με την «Μαθηματοχώρα» που προτείνει ο Papert στο βιβλίο του “Mindstorms”. Τα οπτικά και απτικά χαρακτηριστικά που ενσωματώνει το περιβάλλον στο επίπεδο της γλώσσας προγραμματισμού καθώς και οι δυνατότητες αλληλεπίδρασης και ανατροφοδότησης σε όλες τις φάσεις ανάπτυξης και εκτέλεσης των προγραμμάτων, είναι πιθανόν να επιδρούν θετικά στην βιωματική ανάπτυξη της αλγοριθμικής σκέψης και της αντικειμενοστραφούς λογικής. Η ανάπτυξη τόσο εισαγωγικών, όσο και μιας σειράς στοχευμένων μαθημάτων είναι τα επόμενα βήματά μας, τα οποία θα ακολουθήσει η πειραματική εφαρμογή τους, στη Τριτοβάθμια εκπαίδευση, προκειμένου να καταγραφούν και να αναλυθούν οι διαφοροποιήσεις και τα οφέλη που πιθανόν προκύπτουν τόσο σε εννοιολογικό επίπεδο όσο και σε επίπεδο βασικών προγραμματιστικών δεξιοτήτων. Συγκεκριμένα, θα διεξαχθούν μια σειρά μαθημάτων σε φοιτητές χωρίς προηγούμενη προγραμματιστική εμπειρία και παράλληλα με τα επίσημά τους μαθήματα εισαγωγής στον προγραμματισμό. Η έρευνα θα πραγματοποιηθεί σε δύο ανεξάρτητα τμήματα φοιτητών του Πανεπιστημίου Μακεδονίας, το ένα από φοιτητές τμήματος Πληροφορικής και το άλλο από φοιτητές του τμήματος Εκπαιδευτικής και Κοινωνικής Πολιτικής. Με ερωματολογία, πριν, μετά και κατά την διάρκεια των διδασκαλιών θα καταγραφούν οι μεταβολές σε εννοιολογικό επίπεδο περί της αντίληψης για τον αντικειμενοστραφή προγραμματισμό γενικά αλλά και στις προγραμματιστικές τους δεξιότητες. Αντικείμενο επίσης της μελέτης θα αποτελέσει και η σύγκριση των αποτελεσμάτων στα δύο διαφορετικής προέλευσης και προσανατολισμού δείγματα των φοιτητών. Στη χρήση βέβαια τέτοιου είδους περιβαλλόντων προγραμματισμού, ως

εργαλείων εισαγωγής στον προγραμματισμό, δεν υπάρχει μια μοναδική και ιδανική επιλογή. Πιθανόν ο συνδυασμός τους, με άλλου είδους μικρόκοσμους και μικρο-γλώσσες ή ακόμα και η προσθήκη νέων χαρακτηριστικών να οδηγήσει σε θετικότερα αποτελέσματα. Ο στόχος από την χρήση τέτοιων περιβαλλόντων, ειδικά στην τριτοβάθμια εκπαίδευση, είναι να λειτουργήσουν ως γέφυρα για το ομαλό και πετυχημένο πέρασμα σε εμπορικά αντικειμενοστραφή περιβάλλοντα προγραμματισμού.

## **ΒΙΒΛΙΟΓΡΑΦΙΑ**

- ACM (2001), *Computing Curricula 2001*, Final Report, The Joint Task Force on Computing Curricula, <http://www.computer.org/education/cc2001/cc2001.pdf>
- Kay A. C. (1993), The early history of Smalltalk, *The Second ACM SIGPLAN History of Programming Languages Conference (HOLP-II)*, ACM SIGPLAN Notices, 28(3), 69-75
- Allen R. K., Grant D. D. & Smith R. (1996), Using Ada as the first programming language: A retrospective, *Proceedings of Software Engineering: Education & Practice (SE:E&P'96)*, IEEE Computer Society Press
- Bergin J., Stehlik M., Roberts J. & Pattis R. (1997), *Karel++*. A gentle introduction to the art of Object Oriented Programming, Wiley
- Brusilovsky P., Calabrese E., Hvorecky E., Kouchnirenko A. & Miller P. (1997), Mini-languages: A way to learn programming principles, *Education and Information Technologies*, 2(1), 65-83
- Calabrese E. (1989), Marta - the "Intelligent Turtle", in G. Schuyten & M. Valcke (Eds.), *Proceedings of Second European Logo Conference*, EUROLOGO'89, 111-127, Gent, Belgium,
- Cooper S., Dann W. & Pausch R. (2003), Teaching objects-first in introductory computer science, *SIGCSE'03*, Reno, Nevada, USA, February 19-23,
- Holland S., Griffiths R., Woodman M. (1997), Avoiding object misconceptions, *SIGCSE '97 CA*, USA
- Kölling M. & Rosenberg J. (1996), An object-oriented program development environment for the first programming course, *Proceedings of 27th SIGCSE Technical Symposium on Computer Science Education*, 83-87, ACM, Philadelphia, Pennsylvania
- MicroWorlds Pro® (2000), Logo Computer Systems Inc., Highgate Springs  
<http://www.microworlds.com/solutions/mwpro.html>
- Papert S. (1980), *Mindstorms: Children, computers and powerful ideas*, Brighton, Sussex: Harvester Press,
- Papert S. (1993), *The Children's Machine*, New York: Basic Books
- Pattis R. E. (1981), *Karel the robot. A gentle introduction to the art of programming*, London: Wiley

- Pausch R., Burnette T., Capeheart A. C., Conway M., Cosgrove D., DeLine R., Durbin J., Gossweiler, R., Koga S. & White J. (1995), *Alice: Rapid prototyping system for virtual reality*, IEEE Computer Graphics and Applications
- Proulx V., Raab, R. & Rasala R. (2002), Objects from the beginning with GUIs, *Proceedings of the 7<sup>th</sup> annual conference on Innovation and Technology in Computer Science Education*, 65-69, Archus, Denmark
- Repenning A. & Sumner T. (1995), Agentsheets: A medium for creating domain-oriented visual languages, *Computer*, 28, 17-25
- Repenning A. & Ioannidou A. (1997), Behavior Processors: Layers between End-Users and Java Virtual Machines, *Proceedings of the 1997 IEEE Symposium of Visual Languages*, 402-409, Capri, Italy
- Repenning A. & Ambach J. (1996), Tactile programming: A unified manipulation paradigm supporting program comprehension, composition and sharing, *Proceedings of the 1996 IEEE Symposium of Visual Languages*, 102-109, Boulder, CO, Computer Society
- Resnick M. (1994), *Turtles, termites, and traffic jams*, MIT Press
- Xinogalos S. (2003), ObjectKarel: A didactic Microworld for teaching Object-Oriented Programming, *ITiCSE'0*, Thessaloniki, Greece ACM 1-58113-672-2/03/0006
- Ιωαννίδου Α., Σπυρόπουλος Χ. & Βίγκλας Λ. (2005), Η Ελληνική έκδοση του AgentSheets ως περιβάλλον διδασκαλίας προγραμματισμού μέσα από διαθεματικά παραδείγματα», *Πρακτικά 3<sup>ου</sup> Πανελληνίου Συνεδρίου “Αξιοποίηση των Τεχνολογιών της Πληροφορίας και της Επικοινωνίας στη Διδακτική Πράξη»*, Σύρος
- Κόμης Β. (2005), *Εισαγωγή στη Διδακτική της Πληροφορικής*, Αθήνα: Εκδόσεις Κλειδάριθμος