

Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ JAVA ΣΤΗΝ ΤΕΕ: ΜΙΑ ΠΡΟΤΑΣΗ ΓΙΑ ΤΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ

Βραχνός Ευριπίδης
Καθηγητής Πληροφορικής
ΤΕΕ Μήλου

evrachnos@gmail.com
<http://www.db-net.aueb.gr/evry/>

ΠΕΡΙΛΗΨΗ

Η εισαγωγή των υπολογιστών όλο και περισσότερο στην καθημερινή ζωή, μας δημιουργεί νέες προσδοκίες για τα μαθήματα του προγραμματισμού υπολογιστών, οι οποίες μας οδηγούν στην κάλυψη θεμάτων όπως οι γραφικές διεπαφές χρήστη και ο δικτυακός προγραμματισμός. Η διδασκαλία αυτών των εννοιών σε ένα μάθημα που είναι εισαγωγικό στον προγραμματισμό χρειάζεται μια γλώσσα προγραμματισμού που να έχει την κατάλληλη υποδομή για τέτοια θέματα. Πιστεύουμε ότι η Java είναι μια τέτοια γλώσσα προγραμματισμού αφού οι βιβλιοθήκες συνεχώς επεκτείνονται και έχουν φτάσει να υποστηρίζουν την ανάπτυξη εφαρμογών από τα κινητά τηλέφωνα μέχρι και την πλατφόρμα της ψηφιακής τηλεόρασης. Θα δείξουμε επίσης πως η Java μπορεί να γίνει ένα ισχυρό εργαλείο στα χέρια του καθηγητή της ειδικότητας της πληροφορικής στην ΤΕΕ. Η γλώσσα αυτή μπορεί να χρησιμοποιηθεί για τη διδασκαλία πολλών μαθημάτων της ειδικότητας και όχι μόνο του προγραμματισμού. Στο τέλος δίνουμε και κάποια παραδείγματα διδασκαλίας.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: ΤΕΕ, java, αντικειμενοστρεφής γλώσσα προγραμματισμού

ΕΙΣΑΓΩΓΗ

Η εισαγωγή των υπολογιστών όλο και περισσότερο στην καθημερινή ζωή, μας δημιουργεί νέες προσδοκίες για τα μαθήματα του προγραμματισμού υπολογιστών, οι οποίες μας οδηγούν στην κάλυψη θεμάτων όπως οι γραφικές διεπαφές χρήστη και ο δικτυακός προγραμματισμός. Επίσης, όλες σχεδόν οι σύγχρονες γλώσσες προγραμματισμού ακολουθούν την αντικειμενοστρεφή αρχιτεκτονική. Σε πολλά τμήματα πληροφορικής η πρώτη γλώσσα προγραμματισμού που διδάσκεται έχει αντικειμενοστρεφή χαρακτηριστικά. Η διδασκαλία αυτών των εννοιών σε ένα μάθημα που είναι εισαγωγικό στον προγραμματισμό χρειάζεται μια γλώσσα προγραμματισμού που να έχει την κατάλληλη υποστήριξη για τέτοια θέματα. Μια τέτοια γλώσσα είναι η Java, η οποία συνεχώς επεκτείνει τις βιβλιοθήκες της υποστηρίζοντας νέες τεχνολογίες. Το σημαντικό είναι ότι αυτό δε γίνεται εις βάρος της πολυπλοκότητας της γλώσσας. Η γλώσσα δεν παρουσιάζει σημαντικές αλλαγές, απλά προστίθενται νέες λειτουργίες τις οποίες κάποιος μπορεί να επιλέξει να χρησιμοποιήσει ή όχι.

Από το 1995 που εμφανίστηκε η Java δημιούργησε μεγάλη αίσθηση στην ακαδημαϊκή εκπαιδευτική κοινότητα. Αν κοιτάξει κανείς, μόνο τα ονόματα των εργασιών που γίνονται δεκτές στο ετήσιο συνέδριο SIGCSE της ACM τα τελευταία χρόνια, θα διαπιστώσει ότι οι αναφορές στη Java είναι περισσότερες από τις αναφορές σε όλες τις άλλες γλώσσες προγραμματισμού. Αυτό σημαίνει ότι πολλοί επιστήμονες θεωρούν ότι αυτή η γλώσσα μπορεί να χρησιμοποιηθεί και στην εκπαίδευση.

ΓΙΑΤΙ Η JAVA ΤΑΙΡΙΑΖΕΙ ΣΤΟ ΤΕΕ

Επιλέξαμε τη γλώσσα προγραμματισμού Java εξαιτίας δυο βασικών λόγων: του τρόπου με τον οποίο είναι σχεδιασμένη η γλώσσα και της δημοτικότητάς της τα τελευταία χρόνια. Το τελευταίο δεν είναι εκπαιδευτικό κριτήριο, ας μην ξεχνάμε όμως ότι σκοπός των ΤΕΕ δεν είναι μόνο η εκπαίδευση αλλά και η κατάρτιση. Η Java είναι μια γλώσσα που χρησιμοποιείται ευρέως στη βιομηχανία και η γνώση της είναι σίγουρα μια καλή επένδυση για το επαγγελματικό μέλλον του απόφοιτου της ειδικότητας της πληροφορικής του ΤΕΕ. Επίσης η εκμάθηση και άλλων γλωσσών γίνεται πιο εύκολη αφού οι γλώσσες που χρησιμοποιούνται αυτή τη στιγμή στη βιομηχανία έχουν πολλά κοινά με τη Java. Οι C και C++ έχουν την ίδια σχεδόν σύνταξη με την Java (αν εξαιρέσουμε τη χρήση δεικτών) και η νέα γλώσσα της Microsoft η C# (*C sharp*) που έρχεται μαζί με την τεχνολογία .NET έχει την ίδια ακριβώς φιλοσοφία με τη Java.

Επίσης το γεγονός ότι χρησιμοποιείται από πολλά τμήματα πληροφορικής σε όλο τον κόσμο ως η πρώτη γλώσσα προγραμματισμού για τους πρωτοετείς φοιτητές σημαίνει ότι θα μπορούμε να βρούμε αρκετό υλικό και πληροφορίες στο διαδίκτυο. Είναι πολύ σημαντικό για τον καθηγητή να μπορεί να βρει εύκολα στο Διαδίκτυο δωρεάν: βιβλία, διδακτικά παραδείγματα (tutorials), προγραμματιστικά περιβάλλοντα, συντάκτες, μεταγλωττιστές, ασκήσεις και αρκετό ακόμα υλικό. Τέλος είναι καλό τα παιδιά να γνωρίζουν μια γλώσσα η οποία θα υπάρχει στην αγορά για πολλά χρόνια ακόμα, αφού πολλές νέες εφαρμογές γίνονται σε Java.

Κάποιοι από τους λόγους για τους οποίους η Java είναι μια πολύ καλή επιλογή για την τεχνική επαγγελματική εκπαίδευση είναι οι παρακάτω:

- Αν και αντικειμενοστρεφής, είναι λιτή και περιεκτική.
- Δεν έχει άλλες δομές δεδομένων εκτός του πίνακα αλλά οι βιβλιοθήκες της, δίνουν πρόσβαση σε μεγάλη ποικιλία από τέτοιες δομές.
- Παρέχει εργαλεία για την υλοποίηση γραφικής διεπαφής χρήστη.
- Παρέχει εργαλεία που διευκολύνουν τον δικτυακό προγραμματισμό
- Έννοιες όπως Συμβολοσειρά (String), Λίστα, Στοιβά και Ουρά υπάρχουν έτοιμες στη βιβλιοθήκη της γλώσσας. Ο μαθητής μπορεί να τις χρησιμοποιεί χωρίς να αναγκαστεί να μπει στις λεπτομέρειες υλοποίησής τους.
- Η διαχείριση μνήμης είναι έμφυτη στη γλώσσα και ο μαθητής δεν χρειάζεται να ασχοληθεί με αυτήν.
- Είναι δωρεάν και γενικά υποστηρίζει τη φιλοσοφία του λογισμικού ανοικτού κώδικα

Τέλος ένα άλλο πολύ σημαντικό πλεονέκτημα που έχει η Java σε σχέση με άλλες γλώσσες είναι η επεκτασιμότητά της. Για παράδειγμα όταν κάποια στιγμή στο σύντομο μέλλον αλλάξουν οι διευθύνσεις IP από 32 bit (IPv4) που είναι σήμερα σε 128 bit (IPv6) οι περισσότερες δικτυακές εφαρμογές που είναι γραμμένες σε Java δεν θα αντιμετωπίσουν σχεδόν κανένα πρόβλημα εκτός ίσως από κάποιες πολύ εξεζητημένες περιπτώσεις.

Η JAVA ΣΤΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ ΤΟΥ ΤΕΕ

Η προσέγγιση που θα πρέπει να ακολουθηθεί, κατά τη γνώμη μας στην ΤΕΕ, σχετικά με τον προγραμματισμό, είναι η διδασκαλία μιας και μόνο γλώσσας προγραμματισμού, η οποία θα χρησιμοποιείται σε όσα μαθήματα έχουν ασκήσεις προγραμματισμού. Έτσι θα δημιουργηθεί ένα ενιαίο πλαίσιο γύρω από τη διδασκαλία του προγραμματισμού με μια σύγχρονη γλώσσα. Δεν είναι δυνατόν να μαθαίνει κάποιος προγραμματισμό εν έτει 2005 με την PASCAL ή τη Visual Basic. Θα μπορούσε να υπάρχει ένα τουλάχιστον μάθημα προγραμματισμού στον Α' κύκλο. Στη συνέχεια δείχνουμε τη συνάφεια αυτού του μαθήματος με άλλα μαθήματα της πληροφορικής του ΤΕΕ.

Στο μάθημα *Προγραμματισμός Η/Υ* θα διδάσκονται οι βασικές έννοιες του αντικειμενοστρεφούς προγραμματισμού και οι γνωστές αλγοριθμικές δομές, μαζί με κάποια βασικά στοιχεία γραφικής διεπαφής χρήστη και δομών δεδομένων.

Στο μάθημα *Λειτουργικά Συστήματα* μπορεί να γίνει χρήση της βιβλιοθήκης της γλώσσας για πολυνηματικό προγραμματισμό. Η κλάση *Thread* είναι σχεδιασμένη με απλό τρόπο και μπορεί να χρησιμοποιηθεί για απλές εφαρμογές από τους μαθητές.

Στο μάθημα *Μετάδοση Δεδομένων και Δίκτυα II* μπορεί να γίνει χρήση της βιβλιοθήκης *NET* της γλώσσας, για την υλοποίηση των πρωτοκόλλων του βιβλίου στο περιβάλλον του εργαστηρίου. (Υπάρχει παράδειγμα σε άλλη εργασία μας).

Στο μάθημα *Βάσεις Δεδομένων* θα μπορούσαμε να χρησιμοποιήσουμε τη βιβλιοθήκη *JDBC* για να δει ο μαθητής πως χτίζεται μια ολοκληρωμένη εφαρμογή βάσεων δεδομένων. Υπάρχει φυσικά και η πιο απλή προσέγγιση με *Visual Basic* και *MS Access*. Σε αυτή την περίπτωση όμως αυτά που θα μάθουν οι μαθητές δεν μπορούν να θεωρηθούν καλά προγραμματιστικά εργαλεία για το διαδίκτυο σε αντίθεση με την *MySQL*. Το περιβάλλον αυτό είναι δωρεάν και ανοιχτού κώδικα. Επίσης συνεργάζεται πολύ καλά με την *Java* και μαζί με τη γλώσσα *PHP* αποτελούν αυτή τη στιγμή τα πιο σύγχρονα εργαλεία για ανάπτυξη δικτυακών τόπων με δυναμικές ιστοσελίδες.

Η γλώσσα *PHP* πάνω από βάση δεδομένων *MySQL* είναι αυτά που θα έπρεπε να προστεθούν κατά τη γνώμη μας στην ύλη του μαθήματος *Προγραμματιστικά Εργαλεία στο Διαδίκτυο*, στο κεφάλαιο για την τεχνική *CGI*, δηλαδή την εκτέλεση σεναρίων στην πλευρά του εξυπηρέτη. Επίσης θα χρησιμοποιηθεί η κλάση *Applet* της *Java* για τη δημιουργία *applets*.

Τέλος στο μάθημα *Σχεδίαση και Αξιολόγηση Εφαρμογών* ο μαθητής θα μπορεί να υλοποιήσει μια μεγάλη εφαρμογή κατά τη διάρκεια της χρονιάς που θα συνδυάζει κάποιες από τις παραπάνω τεχνολογίες. Εδώ μπορεί να χρησιμοποιήσει κλάσεις και βοηθητικά προγράμματα όπως το *javadoc* για την αυτόματη παραγωγή της τεκμηρίωσης της εφαρμογής.

Στο επόμενο σχήμα φαίνονται οι αλληλοεξαρτήσεις των μαθημάτων όπως διαμορφώνονται με βάση αυτά που προτείνουμε.



Σχήμα 1. Το βέλος μεταξύ δυο μαθημάτων δηλώνει εξάρτηση του μαθήματος στο οποίο καταλήγει από το μάθημα από το οποίο ξεκινάει

If you want to teach a new paradigm then you need to use a new pedagogy.

Joe Bergin (ACM Java Task Force)

ΟΙ ΔΙΔΑΚΤΙΚΟΙ ΣΤΟΧΟΙ

Οι διδακτικοί στόχοι που θα πρέπει να έχει ένα μάθημα προγραμματισμού με Java είναι κατά τη γνώμη μας : 1) Να κατανοήσουν οι μαθητές τις βασικές αρχές της αλγοριθμικής επίλυσης προβλημάτων 2) Να μπορούν να λύνουν προβλήματα με αντικειμενοστρεφείς τεχνικές 3) Να μάθουν να χρησιμοποιούν τις βιβλιοθήκες της γλώσσας. Το τελευταίο είναι κάτι πολύ σημαντικό στη σημερινή εποχή. Όταν όο μαθητής έχει να λύσει ένα πρόβλημα, κάποιες από τις λειτουργίες που θέλει να υλοποιήσει, κατά πάσα πιθανότητα θα υπάρχουν έτοιμες στη βιβλιοθήκη της java. Το θέμα είναι να ξέρει πως να ψάξει και να τις βρει μόνος τους. Αυτό είναι και το χαρακτηριστικό της σημερινής εποχής της πληθώρας της γνώσης και της εύκολης πρόσβασής της μέσω του παγκόσμιου ιστού.

Επίσης οι μαθητές δεν αρκεί να βρουν την κατάλληλη βιβλιοθήκη αλλά θα πρέπει να και μπορούν να τη χρησιμοποιήσουν. Να χρησιμοποιούν δηλαδή, τις μεθόδους μιας κλάσης της βιβλιοθήκης χωρίς να χρειάζεται να ξέρουν τα πάντα για αυτήν. Να τη βλέπουν δηλαδή σαν ένα “μαύρο κουτί”.

Θα πρέπει να μπορούμε να διαχωρίσουμε τα βασικά χαρακτηριστικά της γλώσσας στα οποία θέλουμε να επικεντρώσουμε το ενδιαφέρον των μαθητών και να αποφύγουμε όσο μπορούμε αναφορές στην πλούσια και αχανή βιβλιοθήκη της γλώσσας. Να αποφύγουμε την παγίδα στην οποία θα πέσουμε αν για κάθε μια νέα κλάση που θα χρησιμοποιούμε, χρειάζεται να εξηγούμε άλλες πέντε.

Ο στόχος δεν είναι να μάθουν οι μαθητές τις λειτουργίες της βιβλιοθήκης της γλώσσας αλλά τη φιλοσοφία της. Εδώ είναι πρέπει να παίζει σημαντικό ρόλο ο καθηγητής. Αυτός είναι που θα πρέπει να επιλέξει ένα υποσύνολο από τη βιβλιοθήκη της γλώσσας το οποίο θεωρεί ότι θα τον διευκολύνει και όχι θα τον δυσκολέψει στη διδασκαλία του μαθήματος. Σε πολλές περιπτώσεις θα χρειαστεί να αναπτύξει και αυτός μια κατάλληλη διεπαφή η οποία θα κρύβει από τους μαθητές όλες εκείνες τις λεπτομέρειες που θα δυσκόλευαν την εκπαιδευτική διαδικασία, χωρίς να έχουν καμιά διδακτική αξία. Το ότι εμείς μάθαμε προγραμματισμό με τον δύσκολο τρόπο δεν σημαίνει ότι πρέπει να μάθουν και οι μαθητές μας έτσι. Από ότι φαίνεται ο καθηγητής θα πρέπει να δουλέψει λίγο περισσότερο στην προετοιμασία του μαθήματος σε αυτή την περίπτωση (ίσως να χρειαστεί και κάποια επιμόρφωση) αλλά τα “αγαθά κόποις κτώνται”.

Η ΜΕΘΟΔΟΛΟΓΙΑ

Στη βιβλιογραφία αναφέρονται δυο τρόποι διδασκαλίας (Bruce 2004) της Java ανάλογα με τη σειρά προσέγγισης και το βαθμό έμφασης σε κάποιες έννοιες της γλώσσας. Ο πιο “παραδοσιακός” τρόπος είναι να ξεκινήσεις όπως θα ξεκινούσες με οποιαδήποτε άλλη αλγοριθμική (procedural) γλώσσα και να αφήσεις στην άκρη για λίγο τα αντικειμενοστρεφή χαρακτηριστικά της γλώσσας (objects-late). Πως όμως μπορείς να το καταφέρεις αυτό;

Για να δημιουργήσουμε αντικείμενα θα χρησιμοποιήσουμε υποχρεωτικά τον τελεστή *new* ο οποίος καλεί τον κατασκευαστή (*constructor*) της κλάσης. Η είσοδος και η έξοδος είναι δεν είναι άπλες. Ακόμα και αν χρησιμοποιήσουμε μόνο τη μέθοδο `System.out.println()` θα χρειαστεί να εξηγήσουμε τι είναι το `System`, το `out` και το σημαντικότερο από όλα : η τελεία ‘.’ (dot-notation). Δεν μιλάμε φυσικά για το κύριο πρόγραμμα όπου η εμφάνιση της μεθόδου *public static void main(String args[])* στο πρώτο μάθημα έχει ως βέβαιο αποτέλεσμα τον υποδιπλασιασμό του κοινού (στο πανεπιστήμιο) στο επόμενο διάλειμμα. Τέλος δεν μπορούμε να χρησιμοποιήσουμε σχεδόν τίποτα από τη βιβλιοθήκη της γλώσσας αφού όλα είναι δομημένα και οργανωμένα με την αντικειμενοστρεφή φιλοσοφία

Όλα τα παραπάνω θα επιφέρουν σύγχυση στον μαθητή (και πολλές φορές και στον καθηγητή!). Όπως θα υποψιάζεται ο αναγνώστης είμαστε αντικειμενικοί πολέμιοι του παραδοσιακού τρόπου και υιοθετούμε τον δεύτερο τρόπο που θα προτείνουμε στη συνέχεια (objects first). Δηλαδή την εισαγωγή των εννοιών της κλάσης και του αντικειμένου από την αρχή.

Υπάρχουν εκπαιδευτικά προγραμματιστικά περιβάλλοντα τα οποία έχουν αναπτυχθεί για αυτόν ακριβώς τον σκοπό. Την ομαλή εισαγωγή του μαθητή στην αντικειμενοστρεφή φιλοσοφία ή σε έναν αντικειμενοστρεφή κόσμο αν θέλετε. Μερικά από αυτά είναι το DrJava και το BlueJ που είναι το εκπαιδευτικό εργαλείο της επιλογής μας. Να σημειώσουμε ότι και τα δυο διατίθενται δωρεάν στις ομώνυμες διευθύνσεις.

Μια άλλη κατηγορία εργαλείων που έχουν αναπτυχθεί για τη διδασκαλία της αντικειμενοστρεφούς φιλοσοφίας είναι οι μικρόκοσμοι αντικειμένων: Ο πιο γνωστός είναι ο Καρέλ το ρομπότ και πιο συγκεκριμένα η αντικειμενοστρεφής του έκδοση Karel++. Η έκδοση αυτή παρέχει ένα περιβάλλον όπου οι μαθητές μπορούν να αναπτύξουν σύντομα και απλά προγράμματα μέσω των οποίων χειρίζονται αντικείμενα-ρομπότ. Μπορούν να δημιουργούν αντικείμενα, να τα ομαδοποιούν σε κλάσεις και να καλούν τις λειτουργίες τους μέσω μηνυμάτων. Είναι γνωστό ότι πολλοί άνθρωποι μαθαίνουν από παραδείγματα και ύστερα γενικεύουν, όπως μαθαίνουν και οι υπολογιστές δηλαδή (Mitchell 1997). Η επαφή ενός μαθητή με έναν μικρόκοσμο αντικειμένων έχει σαν σκοπό να διακρίνει ο μαθητής τις ομοιότητες μεταξύ κάποιων αντικειμένων και στη συνέχεια να τα κατατάξει σε τάξεις. Για κάθε τάξη θα ορίσει μια κλάση της Java. Θα συνεχίσει να διακρίνει τις ιδιότητές τους αλλά και τις λειτουργίες που μπορούν να επιτελέσουν.

Πέρα όμως από τη χρήση κάποιου εκπαιδευτικού λογισμικού ας δούμε μερικές από τις έννοιες που θα πρέπει να εξηγήσουμε στα πρώτα μαθήματα:

- Τα πάντα είναι αντικείμενα
- Τα αντικείμενα δημιουργούνται από τις κλάσεις
- Πολλά αντικείμενα μπορούν να δημιουργηθούν από μια κλάση
- Ένα πρόγραμμα είναι ένα σύνολο από αντικείμενα που λέει το ένα στο άλλο τι να κάνει μέσω μηνυμάτων
- Κάθε αντικείμενο έχει τον δικό του χώρο στη μνήμη
- Κάθε αντικείμενο ανήκει σε μια κλάση
- Όλα τα αντικείμενα του ίδιου τύπου μπορούν να λάβουν τα ίδια μηνύματα.
- Όλα τα αντικείμενα της ίδιας κλάσης έχουν την ίδια δομή
- Η κατάσταση κάθε αντικειμένου είναι διαφορετική και καθορίζεται από τις ιδιότητές του.
- Τα αντικείμενα έχουν λειτουργίες τις οποίες μπορούμε να καλέσουμε.
- Οι μέθοδοι μπορούν να έχουν παραμέτρους και αποτελέσματα

Από τα παραπάνω είναι φανερό ότι δε μπορούμε να ξεκινήσουμε από την αρχή με ένα πλήρες πρόγραμμα αλλά μόνο με τμήματά του. Ο καθηγητής θα πρέπει να έχει ήδη ετοιμάσει ασκήσεις στις οποίες ο μαθητής θα πρέπει να συμπληρώνει τον κώδικά του ώστε να επιτελεί συγκεκριμένες λειτουργίες. Δηλαδή κάτι σαν τις ερωτήσεις συμπλήρωσης κενών ή σαν να συμπληρώνεις τα κομμάτια σε ένα μισοτελειωμένο παζλ. Μόνο που εδώ ο καθηγητής έχει συμπληρώσει πρώτος τα δύσκολα κομμάτια του παζλ και περιμένει και ο μαθητής να κάνει το ίδιο με αυτά που απομένουν.

Η ΒΙΒΛΙΟΘΗΚΗ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΤΟΥ ΤΕΕ

Εκτός από τις ασκήσεις που θα πρέπει να ετοιμάζει για κάθε μάθημα ο καθηγητής υπάρχει και ένα ακόμη σημαντικό θέμα που πρέπει να αντιμετωπίσει. Την τεράστια έκταση και το μεγάλο βαθμό πολυπλοκότητας της βιβλιοθήκης λογισμικού που φέρνει μαζί της η Java.

Ο καθηγητής δεν μπορεί να έχει την απαίτηση από έναν μαθητή, να μπορεί να ελίσσεται μέσα σε αυτόν τον τεράστιο όγκο πληροφοριών. Τουλάχιστον όχι τους πρώτους μήνες. Η λύση σε αυτό το πρόβλημα είναι η ανάπτυξη μιας βιβλιοθήκης για καθαρά εκπαιδευτικούς σκοπούς. Η βιβλιοθήκη αυτή θα είναι υποσύνολο της βιβλιοθήκης της γλώσσας και θα περιλαμβάνει μόνο εκείνες τις λειτουργίες που θεωρούνται απαραίτητες για ένα εισαγωγικό μάθημα στον προγραμματισμό. Επίσης θα είναι αρκετά απλή και κατανοητή για τους μαθητές.

Μια τέτοια βιβλιοθήκη θα μπορούσε να αναπτυχθεί από μια ομάδα εργασίας καθηγητών πληροφορικής των ΤΕΕ έτσι ώστε να είναι κοινή για όλους. Στη συνέχεια ο κάθε καθηγητής θα μπορεί να προσθέτει νέες λειτουργίες. Αυτή άλλωστε είναι και η ουσία του λογισμικού ανοικτού κώδικα.

Ένα παράδειγμα είναι η χρήση γραφικών διεπαφών (όπως παράθυρα διαλόγων, μενού κλπ) η οποία μπορεί πλέον με τη Java, να χρησιμοποιηθεί σε ένα μάθημα εισαγωγής στον προγραμματισμό. Συνήθως προγράμματα με γραφική έξοδο εντυπωσιάζουν τους μαθητές. Οι βιβλιοθήκες *awt* και *swing* της *java* είναι όμως δύσχρηστες για έναν μαθητή Λυκείου που δεν έχει άλλη εμπειρία από προγραμματισμό. Η λύση θα μπορούσε να ήταν ο σχεδιασμός μιας δικής μας βιβλιοθήκης-εργαλειοθήκης με λειτουργίες για τον σχεδιασμό της διεπαφής της εφαρμογής. Οι μέθοδοι στην βιβλιοθήκη αυτή θα είναι έτσι σχεδιασμένες ώστε να αποκρύπτουν όσο το δυνατόν περισσότερες λεπτομέρειες από τους μαθητές, οι οποίες θα τους αποσπούσαν την προσοχή από τους διδακτικούς στόχους που έχουμε θέσει.

Αντίστοιχες βιβλιοθήκες έχουν αναπτυχθεί από πανεπιστήμια στο εξωτερικό. Χαρακτηριστικά παραδείγματα είναι η βιβλιοθήκη προγραμματισμού *Java Power Tools*, η βιβλιοθήκη *Widget* για σχεδιασμό γραφικών όπως επίσης και το γνωστό βιβλίο της Judy Bishop *Java Gently*, στον δικτυακό τόπο του οποίου προσφέρονται δωρεάν όλα τα παράδειγμα του βιβλίου.

Στη συνέχεια δίνουμε ένα απλό παράδειγμα απλοποίησης των λειτουργιών εισόδου-εξόδου που έχουμε δοκιμάσει μέσα στην τάξη.

ΕΙΣΟΔΟΣ/ΕΞΟΔΟΣ ΣΤΗ JAVA: ΕΝΑ ΑΠΛΟ ΠΑΡΑΔΕΙΓΜΑ

Ετοιμάσαμε για τους μαθητές την κλάση με τη μέθοδο `main(String [ ])`, η οποία καλεί τη μέθοδο `soSimple()`. Αυτή είναι και η μέθοδος που έπρεπε να υλοποιήσουν οι μαθητές. Η μέθοδος αυτή διαβάσει τους βαθμούς 6 μαθημάτων και υπολογίζει και εμφανίζει το μέσο όρο του μαθητή στα μαθήματα αυτά.

Αρχικά τοποθετήσαμε το αρχείο `soSimple.java` στη διαμοιραζόμενη περιοχή του εξυπηρετή του εργαστήριου και οι μαθητές το πήραν από εκεί. Ο στόχος ήταν να υλοποιήσουν τη μέθοδο `soSimple()` κάνοντας χρήση των μεθόδων εισόδου/εξόδου `read()` και `print()`, που έχουμε ορίσει εμείς για να διευκολύνουμε τους μαθητές. Το τελικό πρόγραμμα φαίνεται παρακάτω. Να σημειώσουμε ότι το προγραμματιστικό περιβάλλον που χρησιμοποιήθηκε είναι το JCreator το οποίο διατίθεται δωρεάν από την ομώνυμη διεύθυνση.

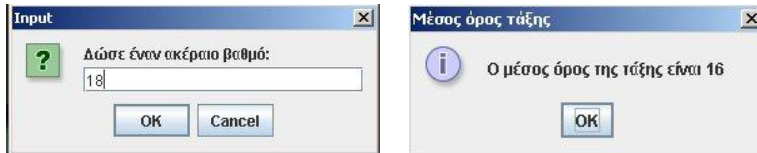
```
public static void soSimple( )
{
    int total, i, grade, average;    // Δηλώσεις των μεταβλητών
    String Message;
    total = 0;                       // Δίνουμε αρχικές τιμές
    i = 1;
```

```

while ( i <= 6 ) {           // Επανάλαβε έξι φορές
    grade = read();          // Διάβασε έναν βαθμό
    total = total + grade;    // Πρόσθεσέ τον στο σύνολο
    i = i + 1;               // Να ξέρω σε ποιον βαθμό είμαι
}

average = total / 6;         // Υπολόγισε τον μέσο όρο
                             // Φτιάξε το μήνυμα που θα εμφανιστεί
Message = "Ο μέσος όρος της τάξης είναι " + average;
print(Message);             // Εμφάνισε το μήνυμα
System.exit( 0 );           // Δώσε εντολή να τερματιστεί το πρόγραμμα
}

```



Σχήμα 2. Η εισόδος και έξοδος για το παραπάνω παράδειγμα. Φυσικά θα εμφανιστούν 6 πλαίσια εισόδου για τους 6 βαθμούς και μια έξοδος στο τέλος για το αποτέλεσμα

Παρακάτω δίνουμε τις μεθόδους που ορίσαμε ώστε να απλοποιήσουμε για τους μαθητές μας τις λειτουργίες εισόδου-εξόδου που είναι άνευ σημασίας. Αυτές αποτελούν ουσιαστικά τη μίνι βιβλιοθήκη μας.

```

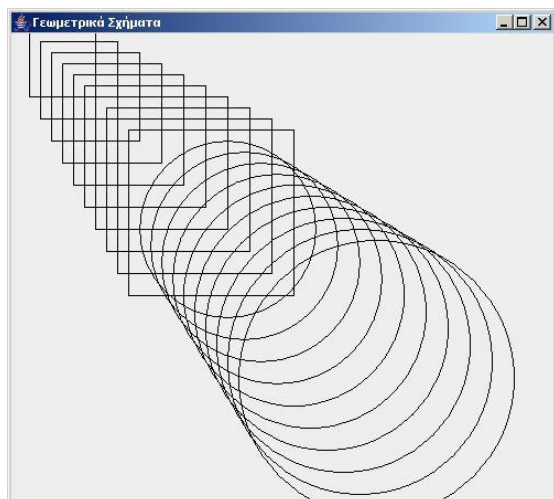
PUBLIC STATIC INT READ()      // ΔΙΑΒΑΖΕΙ ΕΝΑΝ ΑΚΕΡΑΙΟ ΑΡΙΘΜΟ
{
    STRING GRADE = JOPTIONPANE.SHOWINPUTDIALOG(
        "ΔΩΣΕ ΕΝΑΝ ΑΚΕΡΑΙΟ ΒΑΘΜΟ :");
    INT GRADEVALUE = INTEGER.PARSEINT( GRADE );
    RETURN GRADEVALUE;
}

PUBLIC STATIC VOID PRINT(STRING MESSAGE)    // ΕΜΦΑΝΙΖΕΙ ΣΤΗΝ ΟΘΟΝΗ
ΕΝΑ ΜΗΝΥΜΑ
{
    JOPTIONPANE.SHOWMESSAGEDIALOG( NULL, MESSAGE, "ΜΕΣΟΣ ΌΡΟΣ
ΤΑΞΗΣ",
    JOPTIONPANE.INFORMATION_MESSAGE );
}

```

ΥΛΟΠΟΙΗΣΗ ΓΕΩΜΕΤΡΙΚΩΝ ΣΧΗΜΑΤΩΝ

Ένα άλλο ενδιαφέρον παράδειγμα που άρεσε στους μαθητές ήταν η χρήση του αντικειμένου *g* της κλάσης *Graphics* για τη σχεδίαση γεωμετρικών σχημάτων στην οθόνη. Δόθηκε πάλι στους μαθητές ένα αρχείο java σχεδόν έτοιμο, και αυτοί απλά έπρεπε να συμπληρώσουν τις εντολές για την εμφάνιση των σχημάτων στην οθόνη. Αφού εξηγήθηκε η λειτουργία των μεθόδων *drawRect* και *drawOval* του αντικειμένου *g*, αφήσαμε τους μαθητές να πειραματιστούν με αυτούς. Έτσι προέκυψαν τα σχήματα που φαίνονται στο παρακάτω σχήμα. Την ιδέα για αυτή την άσκηση την βρήκαμε στο πολύ γνωστό βιβλίο (Deitel 2002).



Σχήμα 3. Το πρόγραμμα εμφανίζει παραλληλόγραμμα και κύκλους τα οποία μετατοπίζονται αφού σε κάθε επανάληψη αλλάζουμε τις συντεταγμένες τους συναρτήσει του μετρητή της επανάληψης.

Ο κώδικας που ζητήθηκε να γράψουν οι μαθητές δίνεται παρακάτω:

```
for ( int i = 1; i <= 10; i++ )
    g.drawRect( 10 + i*10, 10 + i*10, 50 + i*10, 50 + i*10 );
for ( int i=11; i <= 20; i++ )
    g.drawOval( 10 + i*10, 10 + i*10, 50 + i*10, 50 + i*10 );
```

ΣΥΜΠΕΡΑΣΜΑΤΑ

Στην εργασία αυτή παρουσιάσαμε κάποιες σκέψεις σχετικά με το μάθημα του προγραμματισμού στην ΤΕΕ. Διατυπώσαμε την άποψη ότι το μάθημα αυτό θα πρέπει να αποτελεί τον κεντρικό άξονα του προγράμματος σπουδών του ΤΕΕ. Αρκετά από τα άλλα μαθήματα θα πρέπει να περιλαμβάνουν προγραμματιστικές εργασίες. Για να γίνει αυτό χρειάζεται μια γλώσσα προγραμματισμού η οποία να υποστηρίζει λειτουργίες που θα χρειαστούν σε κάθε μάθημα. Για παράδειγμα θα πρέπει να υποστηρίζει δικτυακό προγραμματισμό για το μάθημα δίκτυα υπολογιστών, πολυνηματικό προγραμματισμό για τα λειτουργικά συστήματα, λειτουργίες βάσεων δεδομένων για το ομώνυμο μάθημα και άλλα. Καταλήξαμε στη γλώσσα προγραμματισμού Java η οποία έχει αρχίσει να γίνεται η πρώτη γλώσσα προγραμματισμού που διδάσκονται οι φοιτητές σε αρκετά πανεπιστήμια.

Δείξαμε πως μπορούμε να διδάξουμε προγραμματισμό με απλό τρόπο στους μαθητές με αυτή τη γλώσσα. Προτείνουμε την ανάπτυξη μιας κοινής βιβλιοθήκης σε Java για εκπαιδευτικούς σκοπούς. Αυτό θα μπορούσε να αποτελέσει λογισμικό ανοικτού κώδικα ώστε να συμμετέχουν όλοι οι εκπαιδευτικοί που θέλουν να προσφέρουν και επίσης να είναι διαθέσιμο σε όλους τους εκπαιδευτικούς. Άλλωστε ένας από τους βασικούς λόγους που επιλέξαμε τη Java είναι ότι διατίθεται δωρεάν και η φιλοσοφία της προάγει την ιδέα του λογισμικού ανοικτού κώδικα.

Ο σκοπός της βιβλιοθήκης αυτής, όπως δείξαμε με τα προηγούμενα παραδείγματα, θα είναι η απλοποίηση πολλών λειτουργιών που μπορούν να προκαλέσουν σύγχυση στους μαθητές. Άλλωστε ο σκοπός μας είναι οι μαθητές να επικεντρωθούν στις βασικές αρχές του

προγραμματισμού και της αλγοριθμικής επίλυσης των προβλημάτων και όχι σε λεπτομέρειες όπως οι λειτουργίες εισόδου/εξόδου. Και η Java είναι η μοναδική ίσως γλώσσα που θα μας βοηθήσει να το πετύχουμε.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Bergin J., Stehlik M., Roberts J. and Pattis R., (2004), Karel J. Robot: A gentle introduction to the art of object-oriented programming in Java. Unpublished.
2. Bruce B. K. (2004), Controversy on How to Teach CS 1: A discussion on the SIGCSE-members mailing list, In Proceedings ACM SIGCSE.
3. Deitel H.M. Deitel P.J. (2002), Java How To Program, Prentice Hall.
4. Eckel (1997), Thinking in Java, Thinking in java, <http://www.EckelObjects.com/Eckel>
5. Kolling, M. (1999), Teaching Object Orientation with the Blue Environment. Journal of Object-Oriented Programming, 12(2), 14-23.
6. Mitchell T. (1997), Machine Learning. Mc Graw-Hill.
7. Raab J, Rasala R. and Proulx V.K. (2000), Pedagogical Power Tools for Teaching Java, Proceedings of the 5th Conference on Innovation and Technology in Computer Science Education, Helsinki, finland, pp. 156-159.
8. Roberts E. (2000), The Widget package for Java, Tech. rep., Stanford University, <http://cse.stanford.edu/java/widget/index.html>
9. SIGCSE, ACM Special Interest Group on Computer Science Education, <http://www.sigcse.org>
10. Βακάλη Α. et. al. (1999), Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον,
11. ΥΠΕΠΘ, Παιδαγωγικό Ινστιτούτο, Αθήνα.