

Προσομοίωση με απλά μέσα: μια εισαγωγή στην Ανάπτυξη Εφαρμογών

Περικλής Γεωργιάδης

Πειραματικό Γενικό Λύκειο Ηρακλείου, perge@sch.gr

Περίληψη

Περιγράφουμε μια διδακτική παρέμβαση με την οποία προσεγγίζονται έμπρακτα οι έννοιες της Ανάπτυξης μιας Εφαρμογής (ΑΕ) και του Προγραμματιστικού Περιβάλλοντος (ΠΠ), για την εισαγωγή στο ομώνυμο μάθημα (ΑΕΠΠ), με την κατασκευή μιας ελάχιστης αλλά πλήρους εφαρμογής που λύνει ένα πρόβλημα. Στο τέλος ενός διδακτικού δίωρου, ο μαθητής θα έχει εισαχθεί στη βασική δομή προγράμματος στη ΓΛΩΣΣΑ, θα έχει λύσει ένα πρόβλημα - μη τετριμμένης απάντησης- με Προσομοίωση, μάλιστα αποκλειστικά με στοιχειώδεις πράξεις χωρίς καμία σύνθετη δομή και θα έχει αντιληφθεί ότι η Διεπαφή Χρήστη - Μηχανής είναι παράλληλο, ανεξάρτητο πεδίο στην ανάπτυξη μιας εφαρμογής. Ως πρόβλημα δίδεται ένας μαθηματικός γρίφος του 1934. Η παρέμβαση, που εφαρμόστηκε με επιτυχία σε 3 τμήματα, χρησιμοποιεί τεχνικές μετωπικής διδασκαλίας και επίδειξης σε υπολογιστή με προβολικό σύστημα, καταϊγισμού ιδεών και έμμεσης καθοδήγησης στην προσχεδιασμένη λύση, και έχει πλήθος επεκτάσεις.

Λέξεις κλειδιά: λύκειο, εφαρμογή, γρίφος, υπολογιστική σκέψη, προσομοίωση

1. Εισαγωγή

Το μάθημα Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον διδάσκεται στους τελειόφοιτους μαθητές Λυκείου του Προσανατολισμού Σπουδών Οικονομίας και Πληροφορικής και αυτού των Θετικών Σπουδών. Οι πρώτοι το εξετάζονται στις Πανελλαδικές Εξετάσεις, ενώ η αξιολόγηση για τους δεύτερους περιορίζεται στη βαθμολογία των 2 τετραμήνων, χωρίς τελική γραπτή εξέταση. Για τους μαθητές της πρώτης κατηγορίας κύριος στόχος είναι η επιτυχία στην πανελλήνια εξέταση, και συνακόλουθα η προετοιμασία στις απαιτήσεις και τη φόρμα της εξέτασης αυτής. Με τους μαθητές της δεύτερης, ο διδάσκων έχει μεγαλύτερο βαθμό ελευθερίας και την ευκαιρία να τους καλλιεργήσει την Υπολογιστική Σκέψη (ΥΣ) (Wing, 2006), χωρίς τη μονομέρεια των πανελλαδικών εξετάσεων. Βέβαια, μαζί με τις αντίστοιχες προκλήσεις (Γεωργιάδης, 2016), πιθανόν να αντιμετωπίσει σε μερίδα μαθητών επιπλέον το μειωμένο κίνητρο ενασχόλησης με το αντικείμενο, ελλείψει τελικής εξέτασης, ή/και κάποια δυσπιστία, βάσει της φτωχής εμπειρίας του προηγούμενου μαθήματος στην Β΄ Τάξη (Γεωργιάδης, 2017).

Υπό τις παραπάνω συνθήκες, είναι σημαντικό για τον διδάσκοντα να καταφέρει να

δημιουργήσει γρήγορα στο ακροατήριό του τις παραστάσεις και τις προσδοκίες εκείνες, που θα συμβάλουν στη δημιουργία και τη διατήρηση ενός γνήσιου ενδιαφέροντος για τον προγραμματισμό και την ΥΣ, που αποτελούν τους ευρείς άξονες του μαθήματος για όλους τους μαθητές.

Στα επόμενα, παρουσιάζουμε, μια διδακτική παρέμβαση, ακριβώς, για το πρώτο δίωρο του μαθήματος, όπου επιχειρείται η έμπρακτη εξήγηση των συστατικών του τίτλου του μαθήματος, και η εισαγωγή του μαθητή στην ΥΣ. Με το πέρασμα του δίωρου, στοχεύουμε να κατακτήσει έμπρακτα ο μαθητής τις έννοιες της Ανάπτυξης μιας Εφαρμογής και του Προγραμματιστικού Περιβάλλοντος, με τη χρήση της ΓΛΩΣΣΑΣ (Ψηφιακό Σχολείο, 2018) και του Διερμηνευτή της (Γεωργόπουλος, 2001). Παράλληλα ο μαθητής θα έχει πειστεί ότι ακόμη και μια μινιμαλιστική -σε μέγεθος «μικρόκοσμου»- Γλώσσα Προγραμματισμού και ο Διερμηνευτής της αρκεί για την αντιμετώπιση και επίλυση προβλημάτων με ποικίλη πολυπλοκότητα και μέγεθος, καθώς η ΥΣ αναπτύσσεται και καλλιεργείται ανεξαρτήτως μεγέθους του μέσου. Ακόμη, ο μαθητής θα διαπιστώσει ότι η λύση ενός προβλήματος, η κατασκευή μιας εφαρμογής, είναι ανεξάρτητη από την κατασκευή της απαραίτητης διεπαφής με το χρήστη που έχει τις δικές της απαιτήσεις, και δεν επηρεάζει άμεσα την προσέγγιση ενός προβλήματος.

Η παρέμβαση βασίζεται σε έναν παλιό μαθηματικό γρίφο, ο οποίος, αν προσεγγιστεί υπολογιστικά, λύνεται εύκολα, με βασικές αριθμητικές πράξεις· ενώ, αντίθετα, εμφανίζεται μια σχετική πολυπλοκότητα, όταν επιχειρήσει κάποιος να περιγράψει το πρόβλημα με ένα σύστημα εξισώσεων.

2. Οι άξονες της διδακτικής παρέμβασης

Περιγράψουμε στη συνέχεια τους άξονες της διδακτικής παρέμβασης.

2.1 Ανάπτυξη Εφαρμογών και Προγραμματιστικό Περιβάλλον

Με την εξέλιξη των υπολογιστικών μηχανών και την καθιέρωση των ευφώνων τηλεφώνων, η έννοια «εφαρμογή», στο συγκείμενο των υπολογιστικών συσκευών, είναι πλέον διάφανη για τους περισσότερους μαθητές. Είναι έτσι σχετικά εύκολο για τον διδάσκοντα να ξεκαθαρίσει ότι οι όροι Ανάπτυξη Εφαρμογών και Προγραμματισμός είναι ταυτόσημοι και δηλώνουν την κατασκευή εκτελέσιμων από τον απλό χρήστη προγραμμάτων σε κάποια υπολογιστική συσκευή. Για την παραγωγική κατασκευή και συντήρηση προγραμμάτων χρειάζονται μια σειρά εργαλείων που περιλαμβάνουν κατά κανόνα έναν, κατά το μάλλον ή ήττον ευφυή, συντάκτη κειμένου για τη συγγραφή του πηγαίου κώδικα, έναν αποσφαλματωτή, με μηχανισμούς εντοπισμού γραμματικών σφαλμάτων, και αποτύπωσης της μνήμης και της εξόδου κατά την εκτέλεση του κώδικα για τον εντοπισμό λογικών λαθών, εργαλεία κατασκευής γραφικών διασυνδέσεων χρήστη για τις υπό ανάπτυξη εφαρμογές, έναν διερμηνευτή, έναν συνδέτη και μεταγλωττιστή, ή άλλα εργαλεία

αυτόματης παραγωγής κώδικα, καθώς και ένα σύστημα ελέγχου εκδόσεων των παραγόμενων εφαρμογών. Τα παραπάνω εργαλεία είναι χρήσιμο να ενσωματώνονται ενιαία στο ίδιο περιβάλλον εργασίας, το οποίο περιγράφει ο όρος Προγραμματιστικό Περιβάλλον ή Ολοκληρωμένο Περιβάλλον Ανάπτυξης (Integrated Development Environment ή IDE). Ένα συμπυκνωμένο τέτοιο Προγραμματιστικό Περιβάλλον αποτελεί ο Διερμηνευτής της Γλώσσας, που υλοποιεί ακόμη και την παραγωγή εκτελέσιμου προγράμματος που μπορεί να εκτελεστεί χωρίς την παρουσία του Διερμηνευτή.

2.2 Υπολογιστική Σκέψη και προγραμματισμός

Η Υπολογιστική Σκέψη (ΥΣ) αποτελεί μια καθοριστική δεξιότητα για την σύγχρονη αγορά εργασίας, η οποία ανήκει στις βασικές δεξιότητες, όχι μόνο των ειδικών, αλλά κάθε ανθρώπου, και, συνεπώς, πρέπει να διδάσκεται και να καλλιεργείται παντού, ειδικά στη γενική υποχρεωτική εκπαίδευση (Wing, 2006). Ωστόσο, από τη σύλληψή της ακόμη (Papert, 1996), ως την ανάπτυξή της από την Wing και άλλους, η ΥΣ αποτελεί αντικείμενο διεξοδικής συζήτησης στην επιστημονική κοινότητα της Διδακτικής της Πληροφορικής και ευρύτερα, καθώς εγείρονται ερωτήματα που αφορούν, εκτός από τον ίδιο τον ορισμό της, την αξία της, αλλά και το πώς μπορεί να διδαχθεί αποτελεσματικά (Aho, 2012; Denning, 2017). Ο αναθεωρημένος ορισμός της ΥΣ την ορίζει ως *τις νοητικές διαδικασίες που εμπλέκονται στην τυποποίηση ενός προβλήματος και την έκφραση των λύσεων του με τέτοιο τρόπο που ένας άνθρωπος ή μία υπολογιστική μηχανή μπορούν να την εφαρμόσουν αποτελεσματικά* (Wing, 2014).

Ειδικά σε σχέση με την εισαγωγή της ΥΣ στα αναλυτικά προγράμματα της Υποχρεωτικής Εκπαίδευσης από τις μικρές τάξεις η επιστημονική συζήτηση δεν έχει ακόμη καταλήξει σε ομοφωνία, ούτε για το περιεχόμενο, ούτε για τη σημασία της (Shuchi & Roy, 2018) και κυρίως γύρω από τις ενδεδειγμένες πρακτικές διδασκαλίας. Τούτο διότι, παρά τη στενή σχέση ΥΣ και προγραμματισμού, δεν φαίνεται να έχουν επιτυχία μέθοδοι έμμεσης διδασκαλίας της πρώτης μέσω του δεύτερου (Duncan & Bell, 2015), κάνοντας φανερή την ανάγκη δημιουργίας ιδιαίτερων εργαλείων (Repenning, Basawapatna & Escherle, 2016), ενώ, από παιδαγωγική άποψη, τα ερωτήματα σχετίζονται με το απαραίτητο μίγμα επιστημονικής -στο αντικείμενο της Επιστήμης Υπολογιστών- και παιδαγωγικής επάρκειας για προσαρμοσμένη στις αντίστοιχες ηλικίες των μαθητών διδασκαλία.

Σχετική ομοφωνία υπάρχει για τις βασικές πρακτικές της ΥΣ που αποτελούν η υπολογιστική διασύνδεση και σύνθεση, η δημιουργία υπολογιστικών αντικειμένων, η αφαίρεση, η αναγνώριση προτύπων (Selby, 2013), η ανάλυση προβλημάτων και αντικειμένων (Bocconi, et al., 2016), η επικοινωνία και συνεργασία (Seehorn, et al., 2011), όπως και στους γενικούς στόχους της διδασκαλίας ΥΣ που συνοψίζονται στις επτά μεγάλες ιδέες της Επιστήμης Υπολογιστών (College Board, 2016): τη δημιουργικότητα, την αφαίρεση, τα Δεδομένα και την Πληροφορία, τους

Αλγόριθμους, τον Προγραμματισμό, το Διαδίκτυο και τον καθολικό αντίκτυπο (Γεωργιάδης, 2016).

2.3 Γρίφοι, τρικ, προσομοίωση και Υπολογιστική Σκέψη

Ακριβώς επειδή η ΥΣ δεν ταυτίζεται με τον προγραμματισμό (Werner & Denning, 2009), με τον ακριβή της ορισμό να παραμένει ανοιχτό θέμα (Bocconi, et al., 2016), υπάρχει συζήτηση για τη διαφοροποίηση στη διδασκαλία και την αξιολόγηση μεταξύ τους, και για τα κατάλληλα εργαλεία. Έχει προταθεί η χρήση λογικών προβλημάτων, που να μην σχετίζονται άμεσα με τον προγραμματισμό, ή να μην απαιτούν κώδικα (Dagiene & Stururiene, 2016). Παρόμοια, έχουν προταθεί ως κατάλληλο υλικό, που δεν απαιτεί υπολογιστή, αλλά καλλιεργούν την ΥΣ, τα «μαγικά» τρικ (Greenberg, 2017), αλλά και οι γρίφοι με μολύβι (Butler, Bezakova, & Fluet, 2017).

Σε αρκετό από το υλικό αυτό, όπως και στη διδακτική παρέμβαση του παρόντος, η τυποποίηση του προβλήματος περιλαμβάνει κάποιου είδους προσομοίωση. Αυτή δεν αφορά τη ρεαλιστική απεικόνιση κάποιου πραγματικού ή φανταστικού κόσμου, αλλά την αποτύπωση των κατάλληλων μεγεθών ή ποσοτήτων που παριστάνουν την εξέλιξη του συστήματος που περιγράφεται σε ένα πρόβλημα, και των κανόνων με τους οποίους το σύστημα μεταβαίνει από τη μία κατάσταση στην άλλη. Η προσομοίωση αφορά ακριβώς την νοητική μετάβαση από τον άνθρωπο ή τη μηχανή από μια αρχική σε μια τελική κατάσταση, ή το αντίστροφο. Μια εναλλακτική προσέγγιση στην τυποποίηση του προβλήματος μπορεί να αναζητά την καταγραφή ενός συστήματος εξισώσεων που το περιγράφουν, η λύση του οποίου δίνει τους ανοικτούς τύπους για τον υπολογισμό των ζητούμενων. Σε αρκετά προβλήματα, η προσομοίωση μπορεί αποτελεί την απλούστερη προσέγγιση.

2.4 Αναμενόμενες δυσκολίες του μαθητή

Όπως επισημαίνει η ίδια (Wing, 2014), στον αναθεωρημένο ορισμό της ΥΣ καθεμιά λέξη αποτελεί αυστηρό τεχνικό όρο και χρήζει περαιτέρω ανάλυσης. Η *τυποποίηση ενός προβλήματος* δηλώνει την ανάγκη για αυστηρή (ανα)διατύπωση, ενώ η *έκφραση των λύσεων* παραπέμπει στην αναγκαιότητα κατάλληλων γλωσσών, αυστηρότερων στους κανόνες τους από τον καθημερινό λόγο. Στη συνέχεια, οι λέξεις *άνθρωπος*, *υπολογιστική μηχανή*, και *αποτελεσματικά* είναι σημαντικές: τι σημαίνει αποτελεσματικά, και πώς διαφοροποιείται μια μηχανή από τον άνθρωπο; Όπως έχουμε επιχειρηματολογήσει (Γεωργιάδης 2017), αρχικά -εν προκειμένω- ο μαθητής μπορεί να αναζητεί η λύση ενός στιγμιότυπου κάποιου προβλήματος, το οποίο θα πρέπει να γενικεύσει, ως προς τη διατύπωση και στη συνέχεια τις λύσεις του. Αυτό τον οδηγεί στην επιλογή της κατάλληλης γλώσσας, όπου εμφανίζεται ένα μεταπρόβλημα (Μαμονά-Downs & Παπαδόπουλος, 2017): να έχει ξεκαθαρίσει τους όρους δεδομένο, μεταβλητή, ανάθεση τιμής, σε αντιδιαστολή με την εμπειρία του στο συγκεκριμένο άλλων μαθημάτων. Τέλος, τι δηλώνει η *αποτελεσματικότητα* στον

ορισμό, από μηχανή και άνθρωπο; Πρώτα-πρώτα, ανάλογα με την επιλεγμένη γλώσσα, μια λύση μπορεί να λιγότερο ή περισσότερο κομψή, ενώ αποτελεσματική θα είναι αν είναι αποφασίσιμη ή υπολογίσιμη κατά Turing. Αφήνοντας στην άκρη το γεγονός ότι τα μοντέλα υπολογισμού αναθεωρούνται (π.χ. με τους βιολογικούς και κβαντικούς υπολογιστές), εδώ έχουμε άγνωστη ορολογία που δεν κρίνεται σκόπιμο να δώσουμε στο μαθητή. Αυτό που πρέπει να έχει γίνει σαφές από τις προηγούμενες τάξεις, είναι ότι ο περιορισμός των λειτουργιών σε μια λύση μόνο σε είσοδο, έξοδο, υπολογισμό εκφράσεων, αναθέσεις τιμών, επιλογές και επαναλήψεις, προκύπτει από τις αντίστοιχες δυνατότητες του εκτελεστή του: προκειμένου για υπολογιστή, αριθμητικές πράξεις (ακριβέστερα, πράξεις επί δεδομένων από bits), συγκρίσεις, μεταφορές δεδομένων, και, τέλος, αλλαγές στη ροή της εκτέλεσης των εντολών, λειτουργίες, επαρκείς για επίλυση υπολογιστικών προβλημάτων. Το ανοιχτό ερώτημα αν αυτές είναι και οι μόνες λειτουργίες του ανθρώπινου νου στη λύση ενός προβλήματος, προσπερνιέται από τη στιγμή που αναζητούμε λύσεις εφαρμόσιμες και από τον άνθρωπο και από τη μηχανή.

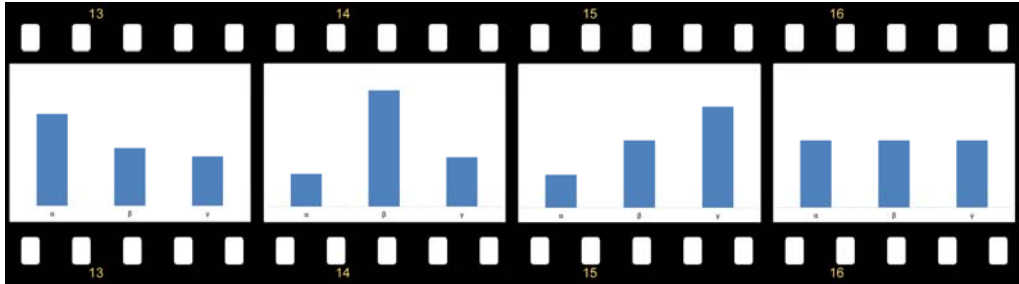
3. Ο γρίφος του Πέρελμαν με τους σωρούς σπίρτα

Στο βιβλίο Ζωντανά Μαθηματικά (Живая математика) που πρωτοεκδόθηκε το 1934 (Перельман, 1934), ο Ρώσος δασολόγος Γιάκοβ Πέρελμαν παραθέτει με χιούμορ μια σειρά από γρίφους και επιστημονικές ιδέες. Ο γρίφος 9 στη σελ. 13 λέει:

Ένας άλλος ομιλητής άδειασε όλα τα σπύρτα από ένα κουτί και άρχισε να τα χωρίζει σε τρεις σωρούς. Θα ανάψεις φωτιά, αστείεύτηκαν οι υπόλοιποι. Ένας γρίφος με σπύρτα, εξήγησε ο πρώτος. Ορίστε τρεις άνισοι σωροί, συνολικά 48 σπύρτα. Δεν σας λέω πόσα στον καθένα. Όμως, σημειώστε τα εξής: αν από τον πρώτο σωρό μεταφέρω στον δεύτερο όσα σπύρτα έχει ο δεύτερος, και μετά από τον δεύτερο μεταφέρω στον τρίτο όσα έχει ο τρίτος, και, τέλος, απ' τον τρίτο βάλω στον πρώτο όσα θα έχει ο πρώτος, αν, λέω, κάνω όλ' αυτά, τότε όλοι οι σωροί θα έχουν το ίδιο πλήθος σπύρτα. Πόσα είχε ο κάθε σωρός αρχικά;

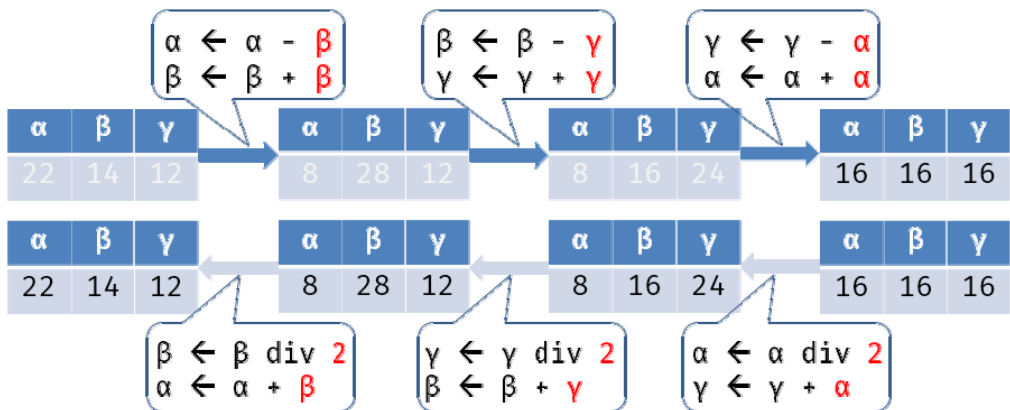
Ο αλγεβρικός αυτός γρίφος λύνεται εύκολα, αν από την τελική κατάσταση κάνουμε τα βήματα προς τις ενδιάμεσες δύο, και τέλος στην αρχική. Αυτό είναι το ανάλογο του να τρέξουμε αντίστροφα τα 4 βασικά «καρέ» ενός ιδεατού φιλμ που απεικονίζει το πρόβλημα (εικόνα 1). Στο τελευταίο (4ο) καρέ οι 3 σωροί έχουν από 16 σπύρτα. Αυτό προέκυψε από το 3ο καρέ με τη μεταφορά από τον 3ο σωρό στον 1ο, τόσων σπύρτων όσων είχε στο καρέ αυτό, με αποτέλεσμα τον διπλασιασμό τους. Άρα για να γυρίσουμε από το 4ο καρέ στο 3ο, αρκεί να πάρουμε τα μισά από τα σπύρτα του 1ου σωρού στο 1ο καρέ και να τα επιστρέψουμε στον 3ο σωρό. Ας δούμε τώρα πώς είχε προκύψει η κατάσταση στο 3ο καρέ από το 2ο. Από το 2ο σωρό του 2ου καρέ μετακινήθηκαν τόσα σπύρτα, όσα είχε στο 2ο καρέ ο 3ος σωρός, με συνέπεια να έχουν διπλασιαστεί στο 3ο καρέ. Άρα για να γυρίσουμε από το 3ο καρέ στο 2ο, αρκεί να πάρουμε τα μισά από τα σπύρτα του 3ου σωρού στο 3ο καρέ και να τα

επιστρέψουμε στον 2ο σωρό. Ας δούμε, τέλος, πώς είχε προκύψει η κατάσταση στο 2ο καρέ από το 1ο. Από το 1ο σωρό του 1ου καρέ μετακινήθηκαν τόσα σπίρτα, όσα είχε στο 1ο καρέ ο 2ος σωρός, με συνέπεια να έχουν διπλασιαστεί στο 2ο καρέ. Άρα για να γυρίσουμε από το 2ο καρέ στο 1ο, αρκεί να πάρουμε τα μισά από τα σπύρτα του 2ου σωρού στο 2ο καρέ και να τα επιστρέψουμε στον 1ο σωρό.



Εικόνα 1. Το φιλμ με τα 4 καρέ που περιγράφουν τον κόσμο του γρίφου

Είναι φανερό ότι τα 3 μεγέθη των σωρών, οι μεταβλητές α , β και γ , είναι αναγκαία και αρκετά για την περιγραφή των 4 καταστάσεων. Στο πάνω μισό της, η εικόνα 2 δείχνει τις τιμές των μεταβλητών αυτών, τα πλήθη των σπύρτων δηλαδή, στις 4 διαδοχικές καταστάσεις: στις 3 πρώτες είναι θολές καθώς δεν τις γνωρίζουμε κατά τη διατύπωση. Γνωρίζουμε, όμως, για κάθε βέλος μετάβασης, τις μεταβολές των τιμών των 2 μεταβλητών που εμπλέκονται αντίστοιχα (εκφρασμένες στη ΓΛΩΣΣΑ).



Εικόνα 2. Οι παράμετροι που περιγράφουν τα 4 καρέ

Από την τελική κατάσταση, στο κάτω μισό της εικόνας 2, μπορούμε να καθορίσουμε τις αντίστοιχες μεταβολές, ώστε να διατρέξουμε αντίστροφα το πρόβλημα προς τη ζητούμενη αρχική κατάσταση, με αρχικές τιμές 16, ή, αν γενικεύσουμε την τιμή 48 σε μια ποσότητα ω που μπορεί να διαβάσει η λύση, $\omega/3$.

Η προσέγγιση που περιγράψαμε αποτελεί ακριβώς την εφαρμογή της Υπολογιστικής Σκέψης· χρησιμοποιεί μόνο προσθέσεις και ακέριους υποδιπλασιασμούς και οι 3 μεταβλητές δεν παριστάνουν τους «αγνώστους του προβλήματος», όπως λανθασμένα θα υποψιαζόταν ένας μαθητής, ορμώμενος από την μαθηματική του εμπειρία, αλλά μεταβάλλονται -όπως δηλώνει το όνομά τους- κατά την προσομοίωση του κόσμου του προβλήματος. Η ανοικτή αλγεβρική προσέγγιση ακολουθεί.

3.2 Ανοικτή αλγεβρική προσέγγιση

Αν ονομάσουμε x_1 , x_2 και x_3 (με $x_1+x_2+x_3=48$) τις ζητούμενες αρχικές ποσότητες, για τους 3 σωρούς, τότε στη 2η κατάσταση οι 3 σωροί αντιστοιχούν στις ποσότητες x_1-x_2 , $2x_2$ και x_3 , στην 3η στις x_1-x_2 , $2x_2-x_3$ και $2x_3$ και στην τελική στις x_1-x_2 , $2x_2-x_3$ και $2x_3-x_1$. Σε αυτήν ισχύουν οι ισότητες $x_1-x_2=48/3$, $2x_2-x_3=48/3$ και $2x_3-x_1=48/3$. Λύνοντας το σύστημα εξισώσεων, με διαδοχικές αντικαταστάσεις, καταλήγουμε στη ζητούμενη λύση $x_1=22$, $x_2=14$ και $x_3=12$. Ή, αν γενικεύσουμε το δεδομένο 48 με μια ποσότητα ω , βρίσκουμε $x_1=11\omega/24$, $x_2=7\omega/24$ και $x_3=6\omega/24$. Είναι προφανές ότι η τιμή του ω πρέπει να είναι πολλαπλάσια του 24, και τέτοια είναι το 48. Στη διατύπωση μιας λύσης με αυτήν την προσέγγιση, μπορούμε να εφαρμόσουμε απευθείας τους ανοικτούς τύπους, βάσει του ω , για τον υπολογισμό των ζητούμενων αρχικών τιμών.

3.3 Γενίκευση του προβλήματος

Η τιμή 24 που βρήκαμε προηγουμένως, προκύπτει ως το γινόμενο $3 \cdot 2^3$. Το 3 είναι το πλήθος των σωρών (και μεταβάσεων), ενώ από τους (υπο)διπλασιασμούς προκύπτει η τιμή 2. Γενικεύοντας το πρόβλημα από τους 3 σωρούς στους v ($v \geq 2$), από το σύστημα εξισώσεων που προκύπτει με αγνώστους x_1, x_2, \dots, x_v , παίρνουμε τη γενική λύση που φαίνεται στον τύπο 1:

$$x_i = \begin{cases} \frac{2^v + 2^{v-1} - 1}{2^v} \cdot \frac{\omega}{v} & \text{για } i = 1 \text{ και } \omega = \kappa \cdot v \cdot 2^v, \kappa \geq 1 \\ \frac{2^{2+v-i} - 1}{2^{2+v-i}} \cdot \frac{\omega}{v} & \text{για } i > 1 \text{ και } \omega = \kappa \cdot v \cdot 2^v, \kappa \geq 1 \end{cases} \quad (1)$$

Στον πίνακα 1 γίνεται φανερή η εκθετική αύξηση του απαραίτητου πλήθους σπίρτων:

Πίνακας 1. Ο γρίφος με διαφορετικά πλήθη σωρών σπίρτων

v	Συν. Πλήθος	Σωροί
2	8κ	$5\kappa, 3\kappa$
3	24κ	$11\kappa, 7\kappa, 6\kappa$
4	64κ	$23\kappa, 15\kappa, 14\kappa, 12\kappa$
5	160κ	$47\kappa, 31\kappa, 30\kappa, 28\kappa, 24\kappa$
6	384κ	$95\kappa, 63\kappa, 62\kappa, 60\kappa, 56\kappa, 48\kappa$

Αντίστοιχα, μπορούμε να εφαρμόσουμε την προσέγγιση της προσομοίωσης και να βρούμε τη λύση με n μεταβλητές που θα παριστάνουν το πλήθος των σπύρων για n σωρούς από $k \cdot n \cdot 2^n$, συνολικά, σπύρα. Φυσικά, αντί για απλές μεταβλητές, θα προτιμήσουμε πίνακα με n στοιχεία και θα εφαρμόσουμε επαναληπτική διαδικασία. Αυτό, φυσικά, ξεφεύγει από τη διδακτική παρέμβαση που περιγράφουμε εδώ.

4. Η διδακτική παρέμβαση

Με αφορμή τον τίτλο του μαθήματος, αρχικά γίνεται μια συζήτηση, βάσει και της εμπειρίας των μαθητών, για τους όρους Εφαρμογή, Ανάπτυξη Εφαρμογών, Προγραμματισμός και Προγραμματιστικό Περιβάλλον. Τη συζήτηση διευκολύνει η χρήση, με προβολικό σύστημα, του Διερμηνευτή της Γλώσσας, ήδη γνώριμου στη λειτουργία της Ψευδογλώσσας. Με αυτόν αποκαλύπτεται κατόπιν το λεξιλόγιο της ΓΛΩΣΣΑΣ και γίνεται μια γρήγορη εισαγωγή στη δομή προγράμματος σε αυτήν, στην αναγκαιότητα δήλωσης των μεταβλητών, και στις εντολές εισόδου - εξόδου. Στην συνέχεια τίθεται ο γρίφος του Πέρελμαν, με χρήση κινούμενης εικόνας.

Με συζήτηση και καταγισμό ιδεών, αλλά και έμμεση καθοδήγηση στην προσχεδιασμένη λύση της χρήσης τριών μεταβλητών και την προσομοίωση της αντιστροφής των τριών μεταβάσεων από την τελική κατάσταση προς την αρχική, οι μαθητές εκτίθενται σε τεχνικές βιωματικής και ανακαλυπτικής μάθησης. Η δομή της παρέμβασης είναι μικτή συμπεριλαμβάνοντας μερικώς συμπεριφοριστικές διδακτικές στρατηγικές, καθώς εισάγονται νέες έννοιες και υπάρχει καθοδήγηση προς τη λύση. Καθώς χρησιμοποιείται προσέγγιση μέσα από ολοκληρωμένο παράδειγμα (Atkinson, et al., 2000, Morrison, Margulieux, & Guzdial, 2015), επιχειρείται η παρουσίαση μιας όψης της συνολικής εικόνας, χωρίς τεχνικές λεπτομέρειες της γλώσσας. Στοχεύοντας στην κινητοποίηση της Υπολογιστικής Σκέψης, ο διδάσκων δημιουργεί και παρουσιάζει «ζωντανά», με τον υπολογιστή και το προβολικό σύστημα, τη λύση στη ΓΛΩΣΣΑ, χρησιμοποιώντας μια μεταβλητή ω και τιμή 48 για το συνολικό πλήθος των σπύρων και τις εντολές ανάθεσης της εικόνας 2 (κάτω μέρος), με αρχικές τιμές $\omega/3$ για τις α , β και γ . Η ολομέλεια συζητά για τη σημασία της σειράς των εντολών μέσα σε κάθε ζευγάρι που καθορίζει τη μετάβαση: πρώτα υποδιπλασιάζουμε /αφαιρούμε το μισό και κατόπιν προσθέτουμε. Παρεμβάλλονται ανάμεσα στα ζεύγη των μεταβάσεων τεχνητά breakpoint, χωρίς αναφορά στον όρο: εντολές εξόδου με τις τρέχουσες τιμές των μεταβλητών, και εντολές εισόδου χωρίς ορίσματα για συνέχεια με enter. Γίνεται, τέλος, συζήτηση για την τιμή του συνολικού πλήθους των σπύρων. Οι μαθητές βλέπουν εύκολα την απαίτηση να είναι πολλαπλάσιο του 3 και άρτιος, όμως μια δοκιμή για $\omega=36$, τους δείχνει ότι αυτό δεν φτάνει. Το σημείο αυτό, δίνει το ερέθισμα να ανατεθεί στους μαθητές η αλγεβρική διερεύνηση και λύση μέσω εξισώσεων, για παρουσίαση σε επόμενο μάθημα, και αξιολόγηση από τους ίδιους της δυσκολίας των δύο προσεγγίσεων.

Ο διδάσκων ολοκληρώνει το πρόγραμμα στη ΓΛΩΣΣΑ, με προσθήκη εντολής

εισόδου που ζητά το συνολικό πλήθος σπύρων, συνέχεια μόνο αν είναι αυτό είναι πολλαπλάσιο του 24, έξοδο με τις τιμές των μεταβλητών σε όλα τα βήματα προς τα πίσω, αλλά, για επιβεβαίωση, και στα βήματα προς εμπρός, με διπλή, δηλαδή, προσομοίωση, πίσω-μπρος. Γίνεται τέλος μια ανασκόπηση για το πόσο απλές εντολές χρειάστηκαν για τη λύση του προβλήματος, που δεν φαινόταν εξ αρχής απλό. Τέλος, η «μεγάλη εικόνα» ολοκληρώνεται με την παραγωγή της ολοκληρωμένης εφαρμογής, που εκτελείται πλέον σε δικό της παράθυρο, και όχι μέσα από τον Διερμηνευτή. Για τον ανήσυχο μαθητή δίνεται η δυνατότητα να αλλάξει ακόμη και το εικονίδιο της με εργαλεία τύπου Resource Hacker.

Με το τέλος της δώωρης παρέμβασης έχουν προσεγγιστεί από τον μαθητή που ξεκινά μόλις το μάθημα της ΑΕΠΠ οι εξής στόχοι:

- έχει αντιληφθεί έμπρακτα τι σημαίνει Ανάπτυξη μιας Εφαρμογής και τι Προγραμματιστικό Περιβάλλον, με την κατασκευή μιας ελάχιστης αλλά πλήρους εφαρμογής που λύνει ένα πρόβλημα
- έχει γνωρίσει τη βασική δομή ενός προγράμματος στη ΓΛΩΣΣΑ
- έχει λύσει ένα πρόβλημα -μη τετριμμένης απάντησης- με Προσομοίωση
- έχει λύσει το πρόβλημα αυτό αποκλειστικά με στοιχειώδεις πράξεις χωρίς καμία σύνθετη δομή
- έχει αντιληφθεί ότι η Διεπαφή Χρήστη - Μηχανής είναι παράλληλο, ανεξάρτητο πεδίο στην ανάπτυξη μιας εφαρμογής
- έχει αντιληφθεί το ρόλο των μεταβλητών στον προγραμματισμό και τη διαφοροποίησή τους από τα μαθηματικά
- έχει καλλιεργήσει την Υπολογιστική του Σκέψη και θα έχει θετική στάση απέναντι στο μάθημα και το αντικείμενο της Επιστήμης των Υπολογιστών, ανεξαρτήτως προσανατολισμού των σπουδών τους.

5. Συμπεράσματα - Επεκτάσεις

Αξίζει να επαναληφθεί ότι η παρέμβαση που παρουσιάστηκε επιζητά μέσα σε ένα διδακτικό δίωρο να παρουσιάσει στους μαθητές την πλήρη «μεγάλη εικόνα» για το αντικείμενο του μαθήματος, που ξεκινά από την αρχική, και συχνά όχι οριστική, διατύπωση ενός προβλήματος, συνεχίζει με τη διερεύνηση, τη σχεδίαση και την υλοποίηση στη ΓΛΩΣΣΑ λύσεων μιας τουλάχιστον σαφώς αναδιατυπωμένης εκδοχής του, και ολοκληρώνεται με την παραγωγή μιας εκτελέσιμης εφαρμογής που εκτελείται απευθείας από το Λειτουργικό Σύστημα. Έτσι επιζητείται να κεντριστεί το ενδιαφέρον της τάξης και να διατηρηθεί στη συνέχεια της χρονιάς.

Η διδακτική παρέμβαση εφαρμόστηκε με επιτυχία τον Σεπτέμβριο 2017 σε 3 τμήματα και, όπως όλο το μάθημα, υποστηρίχτηκε και από Ηλεκτρονική Τάξη στο Πανελλήνιο Σχολικό Δίκτυο. Το ένα από τα τρία τμήματα ήταν προσανατολισμού Σπουδών Οικονομίας και Πληροφορικής, με εξεταζόμενο, δηλαδή, το μάθημα στις

πανελλαδικές εξετάσεις, και αυτό χρησίμευσε για την ανίχνευση της γνησιότητας του ενδιαφέροντος των μαθητών και της -όποιας- αποτελεσματικότητας της παρέμβασης. Το χαρακτηριστικό σε τέτοια τμήματα είναι ότι πάνω από τους μισούς μαθητές έχουν ήδη ξεκινήσει από το καλοκαίρι εξωσχολική προετοιμασία για τα μαθήματα που θα εξεταστούν πανελλαδικά, με συνέπεια την εδραιωμένη εντύπωση ότι η σχολική διδασκαλία δεν έχει να προσφέρει κάτι ιδιαίτερο. Η παρέμβαση στο τμήμα αυτό αφιενδίασε δημιουργικά τους μαθητές, μάλιστα διαπιστώθηκε ότι απαιτήθηκε μεγαλύτερη προσπάθεια για την αντιμετώπιση του προβλήματος, καθώς οι μαθητές ήταν εγκλωβισμένοι σε μοτίβα προσανατολισμένα αποκλειστικά προς το υποτιθέμενο ύψος των ασκήσεων των πανελληνίων εξετάσεων. Εν τέλει, φάνηκε να δημιουργείται γνήσιο ενδιαφέρον στους περισσότερους μαθητές και των τριών τμημάτων, και να υπάρχει κινητοποίηση και στη συνέχεια του μαθήματος μέσα στη σχολική χρονιά.

Σε ένα διαφορετικό πλαίσιο, χωρίς το στόχο της παρουσίασης της «μεγάλης εικόνας», ή σε μια τάξη διαφορετική από τελειόφοιτους, η παρέμβαση θα μπορούσε να έχει λιγότερη μετωπική διδασκαλία και τεχνικές καθοδηγούμενης μάθησης, και να δίδει μεγαλύτερη σημασία στη βιωματική και ανακαλυπτική μάθηση. Σε μικρότερες ηλικίες μια κονστρουβιστική προσέγγιση που θα ενίσχυε το βιωματικό χαρακτήρα θα μπορούσε να κάνει χρήση πραγματικών σπίρτων και να αναπτύσσει ένα πλαίσιο παιχνιδιού ανάμεσα σε ομάδες μαθητών, όπου θα υπήρχαν διαδοχικές προσπάθειες επίλυσης.

Ακόμη, και ανάλογα με το διαθέσιμο χρόνο, διαφορετικές επεκτάσεις αφορούν την αναγκαία και ικανή συνθήκη στο πλήθος των σπίρτων, περισσότερους από 3 σωρούς (μπορεί να ζητούνται απευθείας ως παράμετροι εισόδου το n και το k της εξίσωσης 1), τη μεταφορά διαφορετικού πλήθους σπίρτων μεταξύ των σωρών, κ.ο.κ.,. Ιδιαίτερη κινητοποίηση προσφέρει, επίσης, στους μαθητές η αλγεβρική αντιμετώπιση του προβλήματος με σύστημα εξισώσεων.

Από τα προηγούμενα φάνηκε ότι η παρέμβαση αυτή είναι ιδιαίτερα επεκτάσιμη, και μάλιστα, συνολικά: τόσο ως προς τα ερωτήματα, όσο και ως προς το ακροατήριο, τη γλώσσα προγραμματισμού ή άλλα ενδεικνύμενα εργαλεία. Δεν υπάρχει μονοσήμαντη σχέση, ούτε με το συγκεκριμένο μάθημα, ούτε με τη ΓΛΩΣΣΑ. Ακόμη, η επέκταση σε μεταβλητό πλήθος σωρών, επιτρέπει τη χρήση δεικτοδοτημένων δομών δεδομένων, πινάκων ή λιστών, και αντίστοιχων επαναληπτικών εντολών.

Αναφορές

Aho, A.V. (2012), Computation and computational thinking. *The Computer Journal*, 55(7), 832-835.

- Atkinson, R., Derry, S., Renkl, A., & Wortham, D. (2000). Learning from Examples: Instructional Principles from the Worked Examples Research. *Review of Educational Research, Vol. 70, No. 2*, 181-214.
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., Engelliardt, K, Kampylis, P., & Punie, Y. (2016). Developing computational thinking in compulsory education - Implications for policy and practice. *JRC Science for Policy Report. Publications Office of the European Union*.
- Butler, Z., Bezakova, I., & Fluet, K. (2017). Pencil Puzzles for Introductory Computer Science: an Experience- and Gender-Neutral Context. *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17)*. ACM, New York, NY, USA, 93-98.
- College Board (2016). AP Computer Science Principles Ανάκτηση από το <https://secure-media.collegeboard.org/digitalServices/pdf/ap/ap-computer-science-principles-course-and-exam-description.pdf>
- Dagiene, V., & Stupuriene, G. (2016). Bebras - a Sustainable Community Building Model for the Concept Based Learning of Informatics and Computational Thinking. *Informatics in education 15, 1*, 25-44.
- Denning, P.J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM, 60(6)*, 33-39.
- Duncan, C. & Bell, T. (2015). A Pilot Computer Science and Programming Course for Primary School Students. *Proceedings of the Workshop in Primary and Secondary Computing Education*. ACM, London, United Kingdom, 39-48.
- Greenberg, R.I. (2017). Educational Magic Tricks Based on Error-Detection Schemes. *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '17)*. ACM, New York, NY, USA, 170-175
- Morrison, B., Margulieux, L., Guzdial, M. (2015). Subgoals, Context, and Worked Examples in Learning Computing Problem Solving. *ICER '15 Proceedings of the eleventh annual International Conference on International Computing Education Research*, Omaha, Nebraska, USA

- Papert, S. (1996). An Exploration in the Space of Mathematics Educations. *International Journal of Computers for Mathematical Learning*, 1(1): p. 95-123.
- Repenning, A., Basawapatna, A., & Escherle, N. (2016). Computational Thinking Tools. *IEEE Symposium on Visual Languages and Human-Centric Computing*, IEEE Press: Cambridge, UK.
- Seehorn, D., Carey, S., Fuschetto, B., Lee, I., Moix, D., O'Grady-Cunniff, D., Boucher Owens, B., Stephenson, C., & Verno, A. (2011). CSTA K-12 Computer Science Standards: Revised 2011. Technical Report. ACM, New York, NY, USA.
- Selby, C. (2013). Computational Thinking: The Developing Definition. In Proceedings of the 13th ACM IEEE-CS Joint Conference on Digital Libraries. July 22-26, 2013, Indianapolis, IN, USA. ACM, New York.
- Shuchi, G., & Roy, P. (2018). Computational Thinking: A competency whose time has come. In Sentance, S., Carsten, S., & Barendsen, E. (Eds), *Computer Science Education: Perspectives on teaching and learning*. Bloomsbury
- Werner, L., & Denning, J. (2009). Pair Programming in Middle School. *Journal of Research on Technology in Education* 42, 1, 29-49.
- Wing, J. (2006). Computational thinking. *Communications of the ACM*. 49, no 3, 33-35.
- Wing, J. (2014). Computational Thinking Benefits Society. *40th Anniversary Blog of Social Issues in Computing*. Ανάκτηση από το [http://socialissues.cs.toronto.edu/index.html %3Fp=279.html](http://socialissues.cs.toronto.edu/index.html%3Fp=279.html)
- Перельман, Я. И. (1934). Живая математика. Математические рассказы и головоломки. М.-Л., ПТИ, 1934 <https://www.dropbox.com/s/zyjthsac85h7l7q/perelman-ja-i--zhivaja-matematika.pdf>
- Γεωργιάδης, Π. (2016). Επαναληπτικές Δομές με το παράδειγμα της Φαρμακευτικής Αγωγής και της Εκθετικής Απόσβεσης. *8th Conference on Informatics in Education - Η Πληροφορική στην εκπαίδευση (8th CIE 2016)*, Πειραιάς.
- Γεωργιάδης, Π. (2017). Αρχή με επανάληψη – Εισαγωγή σε μια προσέγγιση top - down στη διδασκαλία του προγραμματισμού. *9th Conference on Informatics in Education - Η πληροφορική στην Εκπαίδευση (9th CIE2017)*, Πειραιάς.

Γεωργόπουλος, Α. (2001). *Ο Διερμηνευτής της ΓΛΩΣΣΑΣ για την «Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον» (ΑΕΠΠ)*. Ανάκτηση από το <http://alkisg.mysch.gr/>

Μαμονά-Downs, Γ. & Παπαδόπουλος, Ι. (2017). *Επίλυση προβλήματος στα Μαθηματικά*. Ηράκλειο: Πανεπιστημιακές Εκδόσεις Κρήτης

Ψηφιακό Σχολείο (2018). *Διδακτικό Πακέτο Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον*. Ανάκτηση από το <http://ebooks.edu.gr/modules/document/document.php?course=DSGL-C101&download=/4c65902ff3dk>

Abstract

We describe a 2-hour teaching intervention in an upper K12 class setup, where students are introduced to the development of a ready-to-use small application in the mini programming language GLOSSA, within a micro-IDE, while being exposed to Computational Thinking (CT). An old Russian math puzzle is exploited, whose solution is approached by simulation. Besides their introduction to the language and its IDE, students realize the CT approach and its merits, and comprehend the fact that a simulation is rather about the values of various parameters, and not a GUI. The intervention has been successfully applied to 3 classes and is open to various extensions.

Keywords: upper K12, application, puzzle, computational thinking, simulation.