Διδάσκοντας Τμηματικό Προγραμματισμό από την πρώτη μέρα

Περικλής Γεωργιάδης

Πειραματικό Γενικό Λύκειο Ηρακλείου, perge@sch.gr

Περίληψη

Περιγράφουμε μια ολιστική σπειροειδή προσέγγιση στη διδασκαλία του Τμηματικού Προγραμματισμού, ενσωματώνοντας τον παράλληλα στη διαπραγμάτευση όλων των εννοιών του δομημένου προγραμματισμού, και όχι απομονωμένο ως αυτοτελές αντικείμενο, προς το τέλος της γραμμικής προσέγγισης ενός τέτοιου μαθήματος, με εργαλείο τη ΓΛΩΣΣΑ, σε μαθητές της Γ΄ Λυκείου, πλαίσιο που ωστόσο δεν είναι μονοσήμαντο. Εκκινώντας από τις ενσωματωμένες συναρτήσεις και το μαθηματικό του υπόβαθρο, ο μαθητής εύκολα δημιουργεί τις πρώτες δικές του μαθηματικές συναρτήσεις, εισάγοντας στη συνέχεια επιπλέον τυπικές παραμέτρους, με αφορμή την εγκυρότητα δεδομένων εισάγεται στις διαδικασίες, και, καθώς κατακτά προγραμματιστικές δομές και δομές δεδομένων, εμπλουτίζει τα υποπρογράμματά του για πιο πολύπλοκα προβλήματα, έμπρακτα κατανοεί τα πλεονεκτήματα του Τμηματικού Προγραμματισμού, και σπειροειδώς χτίζει το θεωρητικό υπόβαθρο πίσω από τον τελευταίο.

Λέξεις κλειδιά: υπολογιστική σκέψη, τμηματικός προγραμματισμός, σπειροειδής μάθηση, ολιστική διδασκαλία

1. Εισαγωγή

Από τα τέλη της δεκαετίας του 1970, οπότε καθιερώθηκε η διδασκαλία του δομημένου και του διαδικαστικού προγραμματισμού, ανακύπτει ο προβληματισμός και η συζήτηση σχετικά με το κατάλληλο χρονικό σημείο για την εισαγωγή και τη διδασκαλία του αρχάριου μαθητή ή φοιτητή στον τμηματικό προγραμματισμό, δηλαδή, στη διάσπαση ενός προβλήματος σε μικρότερα, και στην ανάπτυξη και χρήση υποπρογραμμάτων και βιβλιοθηκών (Pattis, 1993). Η αρχική προσέγγιση υιοθέτησε την αργοπορημένη διδασκαλία για τα παραπάνω, *αφού* έχει προηγηθεί η διδασκαλία όλων των δομών ροής του προγράμματος καθώς και των δομών δεδομένων, θεωρώντας ότι πρόκειται για ένα σχετικά πιο προχωρημένο ζήτημα που απαιτεί τις παραπάνω βασικές γνώσεις και έναν προηγούμενο βαθμό εμπειρίας στην εκάστοτε γλώσσα προγραμματισμού.

Η τάση, ωστόσο, για τη διδασκαλία πλέον του προγραμματισμού *μέσω* μιας γλώσσας προγραμματισμού, αντί για τη διδασκαλία αυτής καθαυτής μιας γλώσσας, μετακίνησε στην δεκαετία του 1980 τον τμηματικό προγραμματισμό προς το κέντρο της διδασκόμενης ύλης, κι αμφισβητήθηκε ακόμη περισσότερο στη συνέχεια με

ISBN: 978-960-578-058-6

προτάσεις για την εισαγωγή σε αυτόν ακόμη νωρίτερα, *παράλληλα* με τις υπόλοιπες βασικές έννοιες και αρχές του προστακτικού προγραμματισμού (Pattis, 1993). Ανασταλτικός παράγοντας παρέμεινε η ανάγκη για τη χρήση παραδειγμάτων - ασκήσεων με προβλήματα που έχουν τον αναγκαίο βαθμό μεγέθους και πολυπλοκότητας που να δικαιολογεί τμηματική προσέγγιση. Το τελευταίο εξαρτάται από το ακροατήριο και το πλαίσιο διεξαγωγής ενός μαθήματος (σχολείο, πανεπιστήμιο, μη παραδοσιακά σχήματα μάθησης), όπως και από τη δυνατότητα της ανάθεσης εκτεταμένων πρότζεκτ, ατομικών ή/και ομαδικών, στους διδασκόμενους.

Με την καθιέρωση του αντικειμενοστρεφούς μοντέλου στην δεκαετία του 1990, αναδείχθηκε παράλληλα ο προβληματισμός και η συζήτηση για το πότε είναι προτιμότερο να εισαχθεί ο μαθητής /φοιτητής στα αντικείμενα, τις κλάσεις, την κληρονομικότητα και τις υπόλοιπες σχέσεις σε αυτά (Pears et al., 2007). Και εδώ η αρχική τάση για την έμφαση της διδασκαλίας απευθείας στις νέες έννοιες του αντικειμενοστρεφούς μοντέλου αμφισβητήθηκε στη συνέχεια με την επιστροφή της προτεραιότητας στις διαδικαστικές δομές (Reges, 2006).

Σύγχρονες, τέλος, γλώσσες προγραμματισμού, όπως η Python, προσφέρουν τη δυνατότητα να ακολουθήσει κανείς παράλληλα, ή διακριτά, την διαδικαστική ή την αντικειμενοστρεφή προσέγγιση, εισάγοντας τον μαθητή με σχεδόν φυσικό τρόπο στον τμηματικό προγραμματισμό.

Στη σχετική με τη διδακτική του προγραμματισμού ή/και της υπολογιστικής σκέψης ελληνική και διεθνή βιβλιογραφία, και με εξαίρεση ίσως την αναδρομή, σποραδικά μόνο φαίνεται να μελετώνται ζητήματα όπως το παραπάνω που να εστιάζουν στον τμηματικό προγραμματισμό και τα υποπρογράμματα, και ειδικότερα στη διδασκαλία τους στην σχολική εκπαίδευση (Γεωργιάδης, 2014). Εν μέρει αυτό οφείλεται στο γεγονός ότι οι αλγοριθμικές δομές που χρησιμοποιούνται σε αυτά δεν είναι διαφορετικές από τις κλασικές. Ζητήματα, ωστόσο, όπως η προστιθέμενη αξία παρά τη φαινομενική επιβάρυνση (overhead) σε κώδικα και χρόνο εκτέλεσης, η εμβέλεια των μεταβλητών, η διαχείριση τοπικών μεταβλητών, το πέρασμα παραμέτρων, καθώς και τα κριτήρια και οι τεχνικές διάσπασης ενός σύνθετου προβλήματος, έχουν μείνει χωρίς να μελετηθούν σχετικά. Κι αυτό παρόλο που πρόκειται για δεξιότητες απόλυτα συνυφασμένες με την υπολογιστική σκέψη γενικότερα (Wing, 2006). Τέλος, αξίζει να αναφερθεί και το οξύμωρο για εκείνον τον μαθητή που έχει ξεκινήσει απευθείας με συναρτησιακό προγραμματισμό μέσω γλωσσών τύπου Logo, για να απομακρυνθεί στη συνέχεια από τον τμηματικό προγραμματισμό.

2. Το πλαίσιο της διδακτικής προσέγγισης

Στην διδακτέα και εξεταστέα στις Πανελλαδικές Εξετάσεις για την Ομάδα Προσανατολισμού Οικονομίας και Πληροφορικής ύλη του μαθήματος της Ανάπτυξης Εφαρμογών σε Προγραμματιστικό Περιβάλλον, τελευταία ενότητα και αντίστοιχο κεφάλαιο του διδακτικού πακέτου αποτελεί ο Τμηματικός

ISBN: 978-960-578-058-6

Προγραμματισμός (Ψηφιακό Σχολείο, 2018). Από το σχολικό έτος 2019-2020, οπότε το μάθημα μετονομάζεται σε Πληροφορική, θα τον ακολουθεί η διαπραγμάτευση των εννοιών του Αντικειμενοστρεφούς Προγραμματισμού και της εκσφαλμάτωσης προγράμματος, αμφότερων θεμάτων μόνο σε θεωρητικό επίπεδο.

Στην ενότητα του τμηματικού Προγραμματισμού οι μαθητές καλούνται να κατακτήσουν τις βασικές έννοιες και αρχές για τα υποπρογράμματα στον προστακτικό προγραμματισμό, υπό την οπτική Γλωσσών Προγραμματισμού τύπου Algol/Pascal:

Τα υποπρογράμματα διακρίνονται σε δύο κατηγορίες, συναρτήσεις και διαδικασίες.

Οι συναρτήσεις δεν έχουν παρενέργειες (side effects) -επιτελούν μόνο υπολογισμούς χωρίς να ασχολούνται με είσοδο ή έξοδο, επιστρέφουν μόνο μία απλή τιμή στο ίδιο το όνομά τους, στο καλούν (υπο)πρόγραμμα, οι τυπικές παράμετροι αντικαθίστανται μόνο με τιμή (by value) από πραγματικές, η δε κλήση τους γίνεται με αναφορά του ονόματός τους και των πραγματικών παραμέτρων μέσα σε εκφράσεις (expressions).

Οι διαδικασίες, από την άλλη μεριά, μπορούν να έχουν παρενέργειες -εκτός από υπολογισμούς να επιτελούν είσοδο ή/και έξοδο, οι τυπικές τους παράμετροι μπορούν να αντικαθίστανται με τιμή (by value) ή όνομα (by name) από πραγματικές, οπότε να χρησιμεύουν για αμφίδρομη επικοινωνία με το καλούν (υπο)πρόγραμμα, επιστρέφοντας πολλαπλές τιμές, η δε κλήση τους γίνεται προστακτικά με κατάλληλη λέξη-κλειδί.

Παράλληλα με τις θεωρητικές αρχές, οι μαθητές καλούνται να κατακτήσουν και τη δεξιότητα κατασκευής και χρήσης υποπρογραμμάτων με εργαλείο την ΓΛΩΣΣΑ, μια λιτή -με τα απολύτως απαραίτητα- γλώσσα δομημένου προγραμματισμού τύπου Algol, που εστιάζει μόνο σε αριθμητικούς τύπους δεδομένων και πίνακες και τον Διερμηνευτή της (Γεωργόπουλος, 2001), ένα ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών που περιλαμβάνει συντάκτη με επισήμανση των γλωσσικών στοιχείων, εκτέλεση βήμα-βήμα, παρακολούθηση τιμών των μεταβλητών και των παραμέτρων, μηνύματα εκσφαλμάτωσης, και μεταγλωττιστή.

Καθώς τα προγράμματα και οι εφαρμογές που υλοποιεί ένας μαθητής δεν έχουν μεγάλη πολυπλοκότητα και το μέγεθός τους είναι περιορισμένο, του προσφέρονται ελάχιστες ευκαιρίες να διαπιστώσει βιωματικά τα πλεονεκτήματα του Τμηματικού Προγραμματισμού:

- την ευκολότερη επίλυση προβλημάτων με τη διάσπασή τους σε μικρότερα,
- την παράλληλη ή ανεξάρτητη κωδικοποίηση,
- την ευκολότερη εκσφαλμάτωση και
- την επανάχρηση κώδικα.

Το γεγονός αυτό, μαζί και η τοποθέτηση του Τμηματικού Προγραμματισμού στο «τέλος της ύλης», που πολλές φορές περιορίζει το απαιτούμενο στη διδασκαλία

ISBN: 978-960-578-058-6

βάθος, όσον αφορά τις έννοιες και τους μηχανισμούς, και *πλάτος*, όσον αφορά το πλήθος κατάλληλων παραδειγμάτων και ασκήσεων, καθιστούν την ενότητα αυτή επιβάτη δεύτερης κατηγορίας στο αντικείμενο του μαθήματος. Διαπίστωση που αντικατοπτρίζεται κατά κανόνα και στα θέματα των πανελλαδικών εξετάσεων.

3. Η διδακτική προσέγγιση του Τμηματικού Προγραμματισμού από τα πρώτα μαθήματα

Στη συνέχεια περιγράφουμε και προτείνουμε την προσέγγιση της σπειροειδούς μάθησης στη διδασκαλία του Τμηματικού Προγραμματισμού, τόσο στη θεωρητική του διαπραγμάτευση, όσο και στην εφαρμογή, με την εισαγωγή των μαθητών σε αυτόν *παράλληλα* με τις υπόλοιπες έννοιες του προγραμματισμού και της ΓΛΩΣΣΑΣ, και την ανάδειξή του σε βασικό εργαλείο, *από τις πρώτες κιόλας διδακτικές ώρες*.

Με αφορμή τις ενσωματωμένες συναρτήσεις της ΓΛΩΣΣΑΣ και το μαθηματικό του υπόβαθρο, ο μαθητής εύκολα δημιουργεί τις πρώτες δικές του μαθηματικές συναρτήσεις εισάγοντας στη συνέχεια και επιπλέον τυπικές παραμέτρους, με αφορμή την εγκυρότητα δεδομένων έρχεται σε επαφή με διαδικασίες, και καθώς προσεγγίζει πλουσιότερες προγραμματιστικές δομές και δομές δεδομένων εμπλουτίζει τα υποπρογράμματά του, έμπρακτα κατανοεί τα πλεονεκτήματα του Τμηματικού Προγραμματισμού, και σπειροειδώς χτίζει το θεωρητικό υπόβαθρο πίσω από τον τελευταίο. Ταυτόχρονα καλλιεργεί εγγενώς μερικές σπουδαίες πτυχές της Υπολογιστικής Σκέψης, όπως η αφαίρεση, η αναγνώριση προτύπων και η γενίκευση.

3.1 Με αφορμή τις ενσωματωμένες συναρτήσεις της ΓΛΩΣΣΑΣ

Η ΓΛΩΣΣΑ περιλαμβάνει 8 ενσωματωμένες αριθμητικές συναρτήσεις με ένα όρισμα χ που είναι πραγματική έκφραση (ή ακέραια που μπορεί να αντικατασταθεί από την ισοδύναμη πραγματική). Πρόκειται για τις:

- Α_Μ(χ), Α_Τ(χ) και Τ_Ρ(χ) για το ακέραιο μέρος, την απόλυτη τιμή και την τετραγωνική ρίζα, αντίστοιχα. Από αυτές, μόνο η πρώτη επιστρέφει τιμή με κατάλληλο τύπο δεδομένων που δεν μπορεί να υπολογιστεί διαφορετικά.
- Ε(χ) και ΛΟΓ(χ) για την εκθετική συνάρτηση e^x και το φυσικό λογάριθμο *lnx*
- ΗΜ(χ), ΣΥΝ(χ) και ΕΦ(χ) για τις 3 βασικές τριγωνομετρικές συναρτήσεις, όπου το όρισμα είναι σε μοίρες. Από τις τρεις συναρτήσεις θα αρκούσε η μία, αφού οι υπόλοιπες μπορούν να παραχθούν μέσω αυτής.

Ο μαθητής εισάγεται στις συναρτήσεις αυτές στις πρώτες διδακτικές ώρες στη ΓΛΩΣΣΑ, οπότε δίδεται η ευκαιρία της χρήσης του μοντέλου του μαύρου κουτιού για τη συνάρτηση, όπου η μοναδική είσοδος χρησιμεύει για να παραχθεί μία μοναδική έξοδος, χωρίς παρενέργειες. Ο άμεσος παραλληλισμός με τη μαθηματική έννοια της συνάρτησης βοηθάει σε αυτό και προσφέρει την ευκαιρία στον

ISBN: 978-960-578-058-6

διδάσκοντα να αναφερθεί με σχεδόν φυσικό τρόπο στη δυνατότητα του προγραμματιστή να κατασκευάσει επιπλέον δικές του συναρτήσεις.

Με μια πρώτη περιστροφή της σπείρας μάθησης, θα λέγαμε μεταφορικά, ο διδάσκων έχει την ευκαιρία να παρουσιάσει τη συντακτική δομή μιας συνάρτησης στη ΓΛΩΣΣΑ, σε ευθεία αντιπαράθεση με τη συντακτική δομή ενός προγράμματος, και να δείξει τον τρόπο κλήσης της, μέσα σε εκφράσεις, σχεδόν φυσικό για τον μαθητή που έχει στο μυαλό του την αναλογία με τις μαθηματικές συναρτήσεις μίας μεταβλητής.

Καθώς μάλιστα, είναι πιθανόν να μην έχει γίνει ακόμη διαπραγμάτευση των δομών ελέγχου της ροής σε ένα πρόγραμμα (δομές επιλογής και επανάληψης), παρά μόνο της ακολουθίας, μπορούν να κατασκευαστούν ως παραδείγματα και ασκήσεις συναρτήσεις μίας γραμμής (one-liners). Μάλιστα, οι ίδιοι οι μαθητές, με την τεχνική του καταιγισμού ιδεών, μπορεί να προτείνουν την κατασκευή συνάρτησης για τον δεκαδικό λογάριθμό (είναι συχνή η αφορμή από τη σχετική παρατήρηση: *με ΛΟΓ συμβολίζουμε τον δεκαδικό λογάριθμο!*), τις υπόλοιπες τριγωνομετρικές, ή όλες τη τριγωνομετρικές, βάσει του ημιτόνου και μόνο.

```
λ10  <- ΛΟΓ(χ)/ΛΟΓ(10)      ! (1) Δεκαδικός Λογάριθμος
λ10  <- ΛΟΓ(χ)/2.3026        ! (2) Δεκαδικός Λογάριθμος
σφ   <- 1/ΕΦ(θ)              ! (3) Συνεφαπτομένη βάσει εφαπτομένης
συνη <- ΗΜ(90 - θ)           ! (4) Συνημίτονο βάσει ημιτόνου
εφη  <- ΗΜ(θ)/ΗΜ(90 - θ)     ! (5) Εφαπτομένη βάσει ημιτόνου
εφη  <- ΗΜ(θ)/συνη(θ)        ! (6) Εφαπτομένη βάσει ημιτόνου με χρήση της (4)
```

Εικόνα 1. *Παραδείγματα συναρτήσεων (κυρίως μέρους) μίας γραμμής*

Στην εικόνα 1 βλέπουμε τα παραδείγματα που αναφέρθηκαν. Αξίζει να τονιστεί ότι ο μαθητής κατανοεί τη συντακτική δομή της συνάρτησης ομαλά στο σημείο αυτό, καθώς δεν παρουσιάζεται ακόμη όλο το υλικό που αφορά τα υποπρογράμματα και την υλοποίησή τους. Υπάρχει, επίσης, συνάφεια με τη συντακτική δομή του κυρίως προγράμματος, και οι επιπλέον γραμμές κώδικα για την σύνταξη των συναρτήσεων δεν αντιμετωπίζονται ως μειονέκτημα (overhead). Παρατηρήστε ακόμη ότι στην (6) έχουμε κλήση της συνάρτησης *συνη(θ)* από την (4), οπότε γίνεται εισαγωγή και στην κλήση συνάρτησης από συνάρτηση.

Δύο ακόμη χρήσιμα ζητήματα μπορεί να αναδείξει η συζήτηση στην τάξη ή το εργαστήριο μέχρι το σημείο αυτό: Στο διδακτικό εγχειρίδιο οι συναρτήσεις της ΓΛΩΣΣΑΣ παρουσιάζονται γενικά ως αριθμητικές, χωρίς διάκριση στον ακριβή τύπο του αποτελέσματος -φυσικά η Α_Μ(χ) επιστρέφει ακέραιο- καθώς επίσης και το ερώτημα για το πώς προνοεί ο προγραμματιστή για το κριτήριο της καθοριστικότητας στις συναρτήσεις του. Για το τελευταίο, ο μαθητής ενθαρρύνεται να διαπιστώσει ότι η ίδια η ΓΛΩΣΣΑ αφήνει να προκληθεί σφάλμα κατά την εκτέλεση (runtime error), που σημαίνει ότι ο προγραμματιστής πρέπει ο ίδιος να μεριμνήσει να μην συμβεί αυτό. Συνεπώς μπορεί να ακολουθήσει την ίδια στρατηγική και στις δικές του

ISBN: 978-960-578-058-6

συναρτήσεις: να είναι το (υπο)πρόγραμμα που καλεί μια συνάρτηση, εκείνο που πρέπει να μεριμνήσει για την αποφυγή σφάλματος. Για το πρώτο, μπορεί να γίνει αναφορά στην υλοποίηση της μετατροπής τύπου (type casting) στις (πλήρεις) γλώσσες προγραμματισμού.

3.2 Περισσότερη πολυπλοκότητα - συναρτήσεις που περιλαμβάνουν επιλογή

Με την εμπιστοσύνη για την ικανότητά του εδραιωμένη, ο μαθητής καλείται στη συνέχεια, στο πλαίσιο κατάλληλου προβλήματος, να υλοποιήσει μια συνάρτηση στρογγυλοποίησης ενός (θετικού, για λόγους απλότητας) πραγματικού αριθμού στον κοντινότερό του ακέραιο. Αυτό μπορεί να γίνει είτε πριν, είτε αφού έχει εισαχθεί στη δομή επιλογής της ΓΛΩΣΣΑΣ, ωστόσο ενθαρρύνεται να υλοποιήσει και πάλι συνάρτηση μίας γραμμής, όπως στη γραμμή (1) της εικόνας 2:

```
! (1) Στρογγυλοποίηση στον πλησιέστερο ακέραιο
στρ <- Α_Μ(χ + 0.5)
! (2) Στρογγυλοποίηση στο ν-οστό δεκαδικό ψηφίο
στρ <- Α_Μ(χ*10^ν + 0.5)/10^ν
! (3) Στρογγυλοποίηση στο ν-οστό δεκαδικό ψηφίο
στρ <- Α_Μ(χ) + Α_Μ((χ - Α_Μ(χ))*10^ν + 0.5)/10^ν
```

Εικόνα 2. *Συναρτήσεις μίας γραμμής για στρογγυλοποίηση*

Ιδιαίτερη αναφορά μπορεί να γίνει στον τύπο του αποτελέσματος σε συνδυασμό με την μετατροπή τύπων (type casting), καθώς η επιστρεφόμενη από την A_M() ακέραια τιμή θα μετατραπεί σε πραγματική, αν ως τέτοια έχει δηλωθεί η *στρ(χ)*.

Πλέον υπάρχει η αφορμή της επέκτασης για την κατασκευή συνάρτησης *στρ(χ, ν)* με δύο παραμέτρους, αν το κατάλληλο πρόβλημα απαιτεί στρογγυλοποιήσεις στο πρώτο ή επόμενο δεκαδικό ψηφίο (π.χ. για τελικούς σχολικούς βαθμούς), οπότε μια δεύτερη παράμετρος *ν* θα παριστάνει το πλήθος των απαιτούμενων δεκαδικών ψηφίων. Και πάλι η ευθεία οδός της χρήσης μίας έκφρασης είναι απλούστερη από τη χρήση δομής επιλογής. Ανάλογα με το ακροατήριο, μπορεί να γίνει και αναφορά στους κινδύνους μη αναγκαίας υπερχείλισης (overflow), εφόσον η παράμετρος *χ* έχει μεγάλη τάξη μεγέθους, και μπορεί να ζητηθεί η υλοποίηση που φαίνεται στη έκφραση 3 της εικόνας 2. Ωστόσο, μπορεί να επισημανθεί ότι η έκφραση 2 λειτουργεί σωστά και για αρνητικά *ν*, στρογγυλοποιώντας στην πλησιέστερη δεκάδα, εκατοντάδα, κλπ. (π.χ. για στρογγυλοποιήσεις σε πληθυσμούς), ενώ η 3 όχι.

Άλλες συναρτήσεις που μπορούν να υλοποιηθούν σε αυτή τη στροφή της σπείρας μάθησης είναι μία Α_Τ(χ) του ίδιου του μαθητή, και συναρτήσεις που επιστρέφουν την ελάχιστη ή μέγιστη τιμή από σε 2 ή 3 παραμέτρους, π.χ. *ελάχ(χ,ψ)* ή *μεγ(χ, ψ, ω)*.

ISBN: 978-960-578-058-6

3.3 Συναρτήσεις που επιστρέφουν λογική τιμή - παίζοντας με ημερομηνίες

Καθώς οι μαθητές αποκτούν εμπειρία στη δομή επιλογής και τους τύπους δεδομένων, είναι χρήσιμο να προχωρήσουν και στην υλοποίηση μη αριθμητικών συναρτήσεων. Αφορμή μπορούν να δώσουν προβλήματα με ημερομηνίες.

Στην εικόνα 3 φαίνεται η έκφραση που επιστρέφει Αληθή ή Ψευδή τιμή για τη συνάρτηση *δίσεκτο(εε)*, που εξετάζει αν είναι δίσεκτο το έτος *εε*, και οι εκφράσεις για τις συναρτήσεις *ααΟΚ(εε)* και *μμΟΚ(μμ)* που επιστρέφουν την εγκυρότητα (Αληθή ή Ψευδή) ενός έτους *εε*, και ενός μήνα *μμ*, ως ακεραίων αριθμών:

```
δίσεκτο <- (εε mod 4 = 0 ΚΑΙ εε mod 100 <> 0) Η εε mod 400 = 0
εεΟΚ <- εε > 0
μμΟΚ <- μμ >= 1 ΚΑΙ μμ <= 12
```

Εικόνα 3. *Λογικές συναρτήσεις (εκφράσεις) για δίσεκτο έτος, εγκυρότητα έτους/μήνα*

Οι 3 παραπάνω συναρτήσεις μπορεί να καλούνται σε μια γενικότερη συνάρτηση που επιστρέφει το αποτέλεσμα εγκυρότητας, Αληθές ή Ψευδές, μιας ημερομηνίας που εκφράζεται με τρεις αριθμητικές παραμέτρους για το έτος, τον μήνα και την ημέρα του μήνα *ημερομηνίαΟΚ(εε,μμ,ηη)*. Μια πιο απαιτητική εφαρμογή που έκανε χρήση δομών επανάληψης, θα μπορούσε μαζί με την εγκυρότητα μιας ημερομηνίας να υπολογίζει και την αντίστοιχη ημέρα της εβδομάδας, με ορισμένους περιορισμούς και επιπλέον δεδομένα· για παράδειγμα στο Γρηγοριανό ημερολόγιο, από τον 17ο αιώνα και μετά, με δεδομένο ότι η πρωτοχρονιά του 1601 ήταν Δευτέρα.

3.4 Διαδικασίες για έλεγχο εγκυρότητας εισόδου

```
ΔΙΑΔΙΚΑΣΙΑ είσοδος(τ, πμ, μσ, α, δ)
ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: τ, α, δ
  ΧΑΡΑΚΤΗΡΕΣ: πμ, μσ
ΑΡΧΗ
  ΓΡΑΨΕ πμ
  ΔΙΑΒΑΣΕ τ
  ΟΣΟ τ < α Η τ > δ ΕΠΑΝΑΛΑΒΕ
    ΓΡΑΨΕ μσ
    ΓΡΑΨΕ πμ
    ΔΙΑΒΑΣΕ τ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

ΚΑΛΕΣΕ είσοδος(τ1, "Βαθμός 1ου τετραμήνου(1-20):  ", "Λάθος βαθμός!", 1, 20)
```

Εικόνα 4. *Διαδικασία εισόδου ακεραίου στο κλειστό διάστημα [α, δ] και κλήση της*

Ο έλεγχος εγκυρότητας των δεδομένων που διαβάζει ένα πρόγραμμα αποτελεί κοινό σημείο αρκετών προβλημάτων. Αποτελεί επίσης ένα κλασικό παράδειγμα χρήσης επαναληπτικής δομής που, μαζί με την εντολή εισόδου, περιλαμβάνει, προαιρετικά,

ISBN: 978-960-578-058-6

κάποιο προτρεπτικό μήνυμα ή κάποιο μήνυμα σφάλματος, όσο δεν ικανοποιείται ή μέχρις ότου ικανοποιηθεί ο έλεγχος εγκυρότητας του δεδομένου που διαβάζεται.

Καθώς πρόκειται για επαναλαμβανόμενο μοτίβο, είναι ευκολότερο να αναδειχθεί η αξία της επανάχρησης ενός υποπρογράμματος, όπως και η εισαγωγή του μαθητή στις διαδικασίες -υποπρογράμματα, που δεν επιστρέφουν απλώς μία τιμή, αλλά εκ κατασκευής έχουν παρενέργειες, μπορούν, δηλαδή, εν προκειμένω στη ΓΛΩΣΣΑ, να μεταβάλλουν τις τιμές των παραμέτρων τους στο (υπο)πρόγραμμα που τις καλεί, καθώς και να επιτελούν είσοδο και έξοδο, παρέχοντας αμφίδρομη επικοινωνία με αυτό. Στην εικόνα 4 με τη διαδικασία *είσοδος(τ, πμ, μσ, α, δ)* δίνεται ένα παράδειγμα υλοποίησης της εισόδου ενός ακεραίου *τ*, έγκυρου στο κλειστό διάστημα [α, δ].

Δίδεται ακόμη, ένα παράδειγμα κλήσης της διαδικασίας για να διαβαστεί ένας έγκυρος βαθμός τετραμήνου σε κάποιο σχολικό μάθημα. Μόνο η παράμετρος *τ1* περνά με αναφορά, ενώ οι άλλες τέσσερις είναι σταθερές τιμές. Οι μαθητές μπορούν να ξεκινήσουν με απλούστερες υλοποιήσεις εγκυρότητας, προσθέτοντας διαδοχικά παραμέτρους και γενικεύοντας. Επιπλέον λογικές παράμετροι μπορεί να καθορίζουν αν τα άκρα είναι ανοιχτά ή κλειστά, ή αν απουσιάζει το ένα άκρο.

3.5 Διαιρετότητα, div, mod και επαναληπτικές δομές

Ο αλγόριθμος του Ευκλείδη για την εύρεση του Μέγιστου Κοινού Διαιρέτη δύο φυσικών αριθμών (Γεωργιάδης, 2017) αποτελεί ένα καλό παράδειγμα δημιουργίας συνάρτησης που απαιτεί επανάληψη (τμήμα 1 στην εικόνα 5). Μάλιστα δίνει την ευκαιρία στο μαθητή να διαπιστώσει την τοπικότητα των παραμέτρων-μεταβλητών, καθώς ενώ μεταβάλλονται μέσα στη συνάρτηση-αλγόριθμο, οι αντίστοιχες στο κύριο πρόγραμμα παραμένουν αμετάβλητες.

```
! (1) Αλγόριθμος του Ευκλείδη για εύρεση ΜΚΔ
ΟΣΟ ν <> 0 ΕΠΑΝΑΛΑΒΕ
  τ <- μ
  μ <- ν
  ν <- τ mod ν
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
μκδ <- μ
! (2) Υπολογισμός ΕΚΠ μέσω ΜΚΔ
εκπ <- μ * ν div μκδ(μ, ν)
! (3) Με αποφυγή μη αναγκαίας υπερχείλισης
εκπ <- μ div μκδ(μ, ν) * ν
```

Εικόνα 5. *Υπολογισμός ΜΚΔ και ΕΚΠ, μέσω ΜΚΔ*

Οι μαθητές μπορούν επίσης να υλοποιήσουν την εύρεση του ΜΚΔ βάσει του ορισμού του, με επαναληπτικές δοκιμές των πιθανών διαιρετών ενός από τους δύο αριθμούς στον άλλο, δοκιμάζοντας βελτιώσεις, όπως την επιλογή των πιθανών διαιρετών του μικρότερου από τους δύο αριθμού, την δοκιμή από τους μεγαλύτερους

ISBN: 978-960-578-058-6

προς τους μικρότερους πιθανούς διαιρέτες, ή την παράλειψη των δοκιμών στο διάστημα μεταξύ αριθμού και του μισού του. Αξίζει, ωστόσο, να γίνει συζήτηση για την βελτίωση της ταχύτητας αλγορίθμου, μόνο όταν αυτή αλλάζει τάξη/εις μεγέθους.

Με παρόμοια προσέγγιση, δηλαδή με επαναληπτικές δοκιμές, οι μαθητές μπορούν να κατασκευάσουν συνάρτηση για την εύρεση του Ελάχιστου Κοινού Πολλαπλασίου δύο φυσικών αριθμών. Και πάλι, ωστόσο, δίδεται η αφορμή παραδείγματος κλήσης συνάρτησης σε συνάρτηση, μέσω της ταυτότητας $ΜΚΔ(μ,ν)×ΕΚΠ(μ,ν)=μ×ν$, που επιτρέπει την υλοποίηση της συνάρτησης του Ελάχιστου Κοινού Πολλαπλασίου ως $μ×ν/ΜΚΔ(μ,ν)$, με κλήση του Μέγιστου Κοινού Διαιρέτη (τμήμα 2 στην εικόνα 5).

Εδώ δίνεται επίσης η αφορμή να συζητηθεί η πιθανότητα υπερχείλισης του γινομένου $μ×ν$ για επαρκώς μεγάλες τιμές των $μ$ και $ν$ εξαιτίας της προτεραιότητας των πράξεων * και *div* από αριστερά. Η υπερχείλιση αυτή δεν είναι αναγκαστική και μπορεί να αποφευχθεί επιλέγοντας τον υπολογισμό $μ$ *div* $ΜΚΔ(μ,ν)$ * $ν$, ώστε να γίνει η διαίρεση πριν τον επίφοβο πολλαπλασιασμό (τμήμα 3 στην εικόνα 5). Έτσι το ΕΚΠ(μ,ν) θα υπερχειλίσει αναγκαστικά μόνο αν οι επαρκώς μεγάλες τιμές των $μ$ και $ν$ είναι και πρώτες μεταξύ τους, και όχι εξαιτίας της προτεραιότητας των πράξεων. Έτσι, ο διδάσκων έχει την ευκαιρία να επισημάνει ότι ο κόσμος των μαθηματικών έχει τον υψηλότερο βαθμό αφαίρεσης ως απολύτως νοητική, αλλά πάντοτε συνεπής σε αξιώματα και θεωρήματα, κατασκευή, ενώ στον προγραμματισμό, όλα τελικά πρέπει να υλοποιηθούν σε θαυμαστής μεν, πεπερασμένης δε τεχνολογίας, ταχύτητας και χωρητικότητας, απτά ψηφιακά κυκλώματα.

```
! Σώμα της πψ(χ)
ι <- 0
ΟΣΟ χ >= 10^ι ΕΠΑΝΑΛΑΒΕ
  ι <- ι + 1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
πψ <- ι
! Σώμα της δψ(χ)
δψ <- χ div Α_Μ(10^(ν - 1)) mod
! Σώμα της αψ(χ)
αψ <- δψ(χ, πψ(χ) + 1 - ν)
! Σώμα της π(χ)
π <- χ = κ(χ)
```

```
! Σώμα της σψ(χ)
σ <- 0
ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ πψ(χ)
  σ <- σ + δψ(χ, ι)
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
σψ <- σ
! Σώμα της κ(χ)
κα <- 0
ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ πψ(χ)
  κα <- κα*10 + δψ(χ, ι)
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
κ <- κα
```

Εικόνα 6. *Έξι συναρτήσεις για τα ψηφία ενός φυσικού αριθμού*

Με τη χρήση επαναληπτικών δομών και τις πράξεις της ευκλείδειας διαίρεσης και υπολοίπου, ο μαθητής μπορεί να κατασκευάσει ακόμα συναρτήσεις σχετικά με τα ψηφία ενός φυσικού αριθμού $χ$. Στην εικόνα 6 παρουσιάζονται τα σώματα έξι τέτοιων συναρτήσεων: Οι $δψ(χ,ν)$ και $αψ(χ,ν)$ επιστρέφουν το ν-οστό -από δεξιά ή αριστερά- ψηφίο ενός φυσικού αριθμού $χ$. Οι $πψ(χ)$ και $σψ(χ)$ επιστρέφουν το πλήθος και το άθροισμα, αντίστοιχα, των ψηφίων του $χ$. Η $κ(χ)$ επιστρέφει τον κατοπτρικό αριθμό του $χ$, τον αριθμό που σχηματίζουν, δηλαδή, τα ψηφία του, παρατιθέμενα με

ISBN: 978-960-578-058-6

αντίστροφη διάταξη. Τέλος, η *π(χ)* επιστρέφει Αληθή ή Ψευδή τιμή αν το χ είναι καρκινικός, ισούται, δηλαδή, με τον κατοπτρικό του, ή όχι. Χωρίς φυσικά να είναι μονοσήμαντες, επιλέχθηκαν υλοποιήσεις που να αναδεικνύουν την επανάχρηση ως βασικό χαρακτηριστικό του τμηματικού προγραμματισμού.

3.6 Συναρτήσεις μορφοποίησης εξόδου που επιστρέφουν τιμή χαρακτήρων

Βασική αδυναμία της ΓΛΩΣΣΑΣ αποτελεί η απλοϊκή υλοποίηση αλφαριθμητικών, που δεν επιτρέπει εφαρμογές με αυτά. Για να αποκτήσει ωστόσο ο μαθητής μια γεύση για τη χρήση και του τύπου δεδομένων αυτού, μπορούν να υλοποιηθούν συναρτήσεις για την έξοδο στην οθόνη μορφοποιημένων ακέραιων αριθμών με διαχωριστική τελεία ανά τριάδα ψηφίων από το τέλος προς την αρχή, ή, με μεγαλύτερο βαθμό δυσκολίας, πραγματικών αριθμών με ελληνική υποδιαστολή και πλήθος δεκαδικών ψηφίων που ορίζει σχετική παράμετρος. Ο μαθητής θα χρειαστεί τον πολυμορφικό τελεστή της πρόσθεσης που επιτελεί συνένωση σε αλφαριθμητικά.

3.7 Διαδικασίες για έλεγχο εγκυρότητας με πίνακα (μοναδικές τιμές εισόδου)

Με την εισαγωγή του μαθητή στις δομές δεδομένων, πλήθος προβλημάτων του δίνουν την ευκαιρία να υλοποιήσει νέες συναρτήσεις ή διαδικασίες, ή να επεκτείνει τη λειτουργικότητα προηγούμενων. Μία παραλλαγή διαδικασίας που επιτελεί έλεγχο εγκυρότητας αφορά την είσοδο μοναδικών τιμών για το γέμισμα ενός πίνακα -για παράδειγμα αυτές μπορεί να είναι πεδία-πρωτεύοντα κλειδιά (αριθμοί μητρώου ή ταυτότητας) σε ένα πρόβλημα με παράλληλους πίνακες. Στην υλοποίηση μιας τέτοιας διαδικασίας, σε κάθε είσοδο επόμενου στοιχείου, πριν αυτό γίνει αποδεκτό, πρέπει να γίνεται έλεγχος ότι δεν υπάρχει ήδη ανάμεσα σε όλα τα προηγηθέντα -κάτι που μπορεί να υλοποιεί με τη σειρά της μια καλούμενη συνάρτηση.

4. Πρακτική εφαρμογή - Συμπεράσματα - Επεκτάσεις

Η προσέγγιση που παρουσιάσαμε εφαρμόστηκε συστηματικά για δύο σχολικά έτη σε τρία συνολικά τμήματα μαθητών. Μπορέσαμε να κινητοποιήσουμε το σύνολο των μαθητών, οι οποίοι εργάστηκαν με γνήσιο ενδιαφέρον καθ' όλη τη διάρκεια του μαθήματος, έχοντας να αντιμετωπίσουν έναν νέο τόπο, μακριά από τη γραμμική κάλυψη της ύλης στην εξωσχολική φροντιστηριακή διδασκαλία που αποτελεί τον κανόνα σε εξεταζόμενα πανελληνίως μαθήματα. Χρειάστηκε η τήρηση μιας λεπτής ισορροπίας μεταξύ της εργασίας στον υπολογιστή και στο χαρτί, καθώς, από τη μία πλευρά, χωρίς τον υπολογιστή, έστω στη μορφή της αντιγραφής και επικόλλησης, είναι αδύνατον να εκτιμηθούν βιωματικά τα πλεονεκτήματα του τμηματικού προγραμματισμού, και από την άλλη, η τελική και τόσο βαρύνουσα πανελλήνια εξέταση στο χαρτί, επιβάλλει την άσκηση της δεξιότητας του μαθητή στη συγγραφή κώδικα χωρίς υπολογιστή.

ISBN: 978-960-578-058-6

Τονίζουμε ότι περιγράψαμε μια γραμμή ολιστικής προσέγγισης στο επιμέρους αντικείμενο του τμηματικού προγραμματισμού, αντιμετωπίζοντάς το, δηλαδή, όχι ως ξεχωριστό μέρος, αλλά ενσωματώνοντάς το στη γραμμή διδασκαλίας ολόκληρου του μαθήματος. Δεν αναφερθήκαμε στις μεθόδους που μπορεί να ακολουθήσει ο διδάσκων στην προσέγγιση αυτή, οι οποίες είναι *ανεξάρτητες*. Καταιγισμός ιδεών, λιγότερη ή περισσότερη μετωπική διδασκαλία, τεχνικές καθοδηγούμενης μάθησης, βιωματική, ανακαλυπτική και μέσα από προβλήματα μάθηση, ομαδοσυνεργατική και κονστρουκτιβιστική προσέγγιση, αποτελούν μεθόδους που χρησιμεύουν στη προσέγγιση του υλικού αυτού, ενσωματωμένες στην προσέγγιση της ύλης ολόκληρου του μαθήματος.

Ιδιαίτερη επίσης σημασία έχει η επιλογή των κατάλληλων προβλημάτων, κατά το δυνατόν ρεαλιστικών. Στη βιβλιογραφία συναντήσαμε κατά κόρον ως παράδειγμα επανάχρησης τη συνάρτηση του παραγοντικού στον υπολογισμό διωνυμικών συντελεστών, διατάξεων, μεταθέσεων, με κλήση της σε αριθμητές και παρονομαστές, κάτι που είναι απολύτως αναποτελεσματικό για μεγάλα μεγέθη· σαφώς προηγούνται οι απλοποιήσεις πριν τους υπολογισμούς. Τα ίδια τα προβλήματα μπορούν ενίοτε να αναδείξουν ιδέες. Για παράδειγμα, στη σύνθεση δύο δυνάμεων, κλασικό απλό πρόβλημα για τη χρήση ενσωματωμένων συναρτήσεων της ΓΛΩΣΣΑΣ, η σχέση $\varepsilon\varphi\theta=F_2 \cdot \eta\mu\varphi/(F_1+F_2 \cdot \sigma\upsilon\nu\varphi)$ για τη διεύθυνση της συνισταμένης δύναμης, δίνει ελάχιστη πληροφορία, σε σχέση με τη συνάρτηση τόξου. Μαθητές που έχουν κατανοήσει τις επαναληπτικές δομές, υλοποίησαν μια τέτοια αντίστροφη συνάρτηση, που προσεγγίζει τη γωνία θ, συγκρίνοντας επαναληπτικά τιμές της συνάρτησης της εφαπτομένης για διαδοχικές δοκιμαστικές τιμές γωνίας έως την επιθυμητή ακρίβεια.

Το προηγούμενο παράδειγμα αναδεικνύει τη δυνατότητα που προσφέρει ο τμηματικός προγραμματισμός, όταν ενσωματώνεται στη διδασκαλία εξαρχής, με φυσικό τρόπο, και δεν απομονώνεται ως υποτιθέμενα προχωρημένο στοιχείο προγραμματισμού, να καλλιεργηθεί η Υπολογιστική Σκέψη (Wing, 2014) στον μαθητή. Η εμπειρία του να ανατρέχει σε έναν πίνακα τιμών της συνάρτησης της εφαπτομένης στο βιβλίο του, μετασχηματίζεται αβίαστα στη συνάρτηση που του λείπει στο πρόβλημα. Αναπτύσσει έτσι τη λειτουργία της αφαίρεσης, και προχωρά και σε αυτήν της γενίκευσης. Η ίδια η συντακτική δομή των υποπρογραμμάτων, ιδίως οι κεφαλίδες τους, βοηθούν στην ανάπτυξη της ικανότητάς του να διακρίνει τα δεδομένα και την ζητούμενη πληροφορία, ενώ η κλήση υποπρογραμμάτων σε υποπρογράμματα και προγράμματα του δείχνουν βιωματικά τα χαρακτηριστικά της διασύνδεσης και της επανάχρησης. Εντέλει ο τμηματικός προγραμματισμός αναδεικνύεται ως ιδανικό μέσο για την ανάπτυξη της Υπολογιστικής Σκέψης.

Η προσέγγιση που παρουσιάσαμε έγινε στο συγκεκριμένο πλαίσιο του μαθήματος της Γ΄ Λυκείου και του προγραμματιστικού εργαλείου της ΓΛΩΣΣΑΣ. Όμως, καθώς δεν υπάρχει μονοσήμαντη σχέση με αυτά, μπορούμε να προσεγγίσουμε με παρόμοιο ολιστικό σπειροειδή τρόπο τη διδασκαλία του τμηματικού προγραμματισμού σε

ISBN: 978-960-578-058-6

οποιαδήποτε γλώσσα δομημένου προγραμματισμού. Με χρήση διαφορετικών προβλημάτων το κάνουμε και στην Α΄ Τάξη σε τμήματα προγραμματισμού σε Python.

Αναφορές

Pattis, R.E. (1993), The “Procedures Early” Approach in CS 1: A Heresy. *Proceedings of the twenty-fourth SIGCSE technical symposium on Computer science education* (SIGCSE '93). ACM, New York, NY, USA, 122-126.

Pears A., Seidman S., Malmi L., Mannila L., Adams E., Bennedsen J., Devlin M., & Paterson J. (2007), A Survey of Literature on the Teaching of Introductory. Programming. *Working group reports on ITiCSE on Innovation and technology in computer science education. Proceedings* (ITiCSE-WGR '07). ACM, New York, NY, USA, 204-223.

Reges, S. (2006), Back to basics in CS1 and CS2. *Proceedings of the 37th SIGCSE technical symposium on Computer science education* (SIGCSE '06). ACM, New York, NY, USA, 293-297.

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

Γεωργιάδης, Π. (2014). Υποπρογράμματα και επαναληπτικές δομές με αφορμή την Εικασία Κόλατς. *6th Conference on Informatics in Education - Η Πληροφορική στην εκπαίδευση (6th CIE 2014),* Κέρκυρα.

Γεωργιάδης, Π. (2017). Αρχή με επανάληψη - Εισαγωγή σε μια προσέγγιση top - down στη διδασκαλία του προγραμματισμού. *9h 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/

ISBN: 978-960-578-058-6

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 holistic spiral-learning approach to the teaching of Structural Programming, embedding it in parallel with all other concepts of structured programming, avoiding isolating it as an independent chapter, towards the end of a linear approach in a relevant course. With the built-in functions and their own mathematical backgrounds to build on, students easily create their first mathematical functions and continue with additional formal parameters, then they build processes to implement input validation, gradually enriching their programming skills and knowledge which allows them to create subprograms for even more complex problems, practically realizing the advantages of Modular Programming, and spirally completing the required theoretical background.

Keywords: computational thinking, modular programming, spiral learning, holistic teaching

ISBN: 978-960-578-058-6