

# Ο ρόλος των δεδομένων στη διδακτική του προγραμματισμού υπολογιστικών συσκευών: μία βιβλιογραφική επισκόπηση

Ρόζου Μαρία<sup>1</sup>, Λαδιάς Αναστάσιος<sup>2</sup>, Παπαδάκης Σπυρίδων<sup>2,3</sup>

[mrozou04@gmail.com](mailto:mrozou04@gmail.com), [ladiastas@gmail.com](mailto:ladiastas@gmail.com), [papspyr@gmail.com](mailto:papspyr@gmail.com)

<sup>1</sup>Εκπαιδευτικός Πληροφορικής, <sup>2</sup>Σχολικός Σύμβουλος Πληροφορικής, <sup>3</sup>Καθηγητής-Σύμβουλος (ΣΕΠ)

## Περίληψη

Τα δεδομένα ως βασική μονάδα στον προγραμματισμό, είναι μία από τις αφηρημένες έννοιες που κατά κανόνα δύσκολα κατακτώνται από τον μαθητή. Η σπουδαιότητα του ρόλου τους καθιστά τη διδακτική παρέμβαση αναγκαία, καθώς ο προγραμματισμός υπολογιστών εξελίσσεται ταχύτατα. Στο παρόν άρθρο επιχειρήσαμε να εκτιμήσουμε τον βαθμό κατανόησης των δεδομένων από τους μαθητές της Α΄ Λυκείου και τους τρόπους διδασκαλίας του συνδυασμού τους για τη δημιουργία κώδικα. Στη βιβλιογραφική επισκόπηση, που ακολουθεί, αναδεικνύεται η ποικιλία των προβλημάτων που αντιμετωπίζουν οι μαθητές στη διδασκαλία του Προγραμματισμού και, κυρίως, στην κατανόηση και την αναπαράσταση των δεδομένων. Η προσέγγιση ακολουθεί τον άξονα της σύγχρονης διδακτικής με σκοπό την εμπέδωση της έννοιας και των χαρακτηριστικών των δεδομένων από τους μαθητές και την αποτελεσματική χρήση τους. Η σε βάθος κατανόηση της έννοιας των δεδομένων αποδεικνύεται ένα δύσκολο πρόβλημα, καθώς εμφανίζεται ανεξάρτητο από τον τύπο (κειμενικά ή οπτικά) των προγραμματιστικών περιβαλλόντων που χρησιμοποιούνται. Επομένως, προτείνεται ο εμπλουτισμός της διδασκαλίας του Προγραμματισμού με στόχο την εμβάθυνση στα δεδομένα και τη λειτουργία τους.

**Λέξεις κλειδιά:** Αναπαράσταση δεδομένων, οπτικός προγραμματισμός με πλακίδια, διδακτική προγραμματισμού

## Εισαγωγή

Το θέμα εμπίπτει στον κοινό τόπο των επιστημών της Διδακτικής και της Πληροφορικής, αφενός με την έννοια της διαχείρισης της πληροφορίας (δεδομένα/αναπαράσταση), αφετέρου με την έννοια του προγραμματισμού υπολογιστικών μηχανών. Από έρευνα της βιβλιογραφίας προκύπτει ότι οι μαθητές, κατά τη μετάβασή τους από το Γυμνάσιο στο Λύκειο, έχουν περιορισμένες δυνατότητες αναπαράστασης των δεδομένων. Επίσης, έρχονται αντιμέτωποι με ένα εύρος δυσκολιών στην κατανόηση και στην αναπαράσταση δεδομένων σε κειμενικά περιβάλλοντα προγραμματισμού και δεν έχουν εμπέδωση σε βάθος τη φύση των δεδομένων και, καθώς αυτά είναι σύνθετες μορφές, δεν μπορούν να τα αναπαραστήσουν σε ικανοποιητικό βαθμό. Η διαπίστωση της περιορισμένης χρήσης σύνθετων μορφών αναπαράστασης δεδομένων και ο μειωμένος βαθμός εμπέδωσής τους αναδεικνύει την αναγκαιότητα της αποτελεσματικότερης διδασκαλίας της αναπαράστασης δεδομένων, θέμα που αποτελεί το αντικείμενο της παρούσας εργασίας. Για τους παραπάνω λόγους θα ερευνηθούν τα δεδομένα και η αναπαράστασή τους σε περιβάλλοντα οπτικού προγραμματισμού.

Οι βασικοί λειτουργικοί ορισμοί που περιγράφουν τους όρους κλειδιά που χρησιμοποιούνται στην έρευνα και οριοθετούν το ερευνητικό μας αντικείμενο είναι: (α) Προγραμματισμός Ηλεκτρονικών Υπολογιστών και δημιουργία κώδικα είναι η διαδικασία

συγγραφής ενός προγράμματος, ώστε να εκτελείται μία συγκεκριμένη εργασία, (β) Δεδομένα είναι τα πάσης φύσεως στοιχεία που εντάσσονται σε ένα πρόγραμμα: αριθμητικά, αλφαριθμητικά, λογικά υπό τη μορφή σταθερών ή μεταβλητών, (γ) Η Διδακτική μελετά τις διαδικασίες μετάδοσης και οικοδόμησης της γνώσης και την καλλιέργεια δεξιοτήτων στο πλαίσιο ατομικών ή συλλογικών εκπαιδευτικών/μαθησιακών καταστάσεων, με απώτερο σκοπό τη βελτίωση των διαδικασιών αυτών.

Η παρούσα εργασία εντάσσεται στο πλαίσιο μεταπτυχιακής διπλωματικής εργασίας στα Πληροφοριακά και Επικοινωνιακά Συστήματα η οποία εκπονείται στο Ανοικτό Πανεπιστήμιο Κύπρου και αποτελεί μία βιβλιογραφική ανασκόπηση για τον ρόλο των δεδομένων στη διδακτική του Προγραμματισμού Η/Υ. Σκοπός της είναι, μέσα από τη βιβλιογραφική επισκόπηση, να εκτιμηθεί ο παράγοντας αναπαράστασης δεδομένων και ο βαθμός οργάνωσης του περιεχομένου σε περιβάλλον οπτικού προγραμματισμού κατά τη διδασκαλία της συγγραφής και αποσφαλμάτωσης κώδικα Η/Υ στην Α΄ τάξη Λυκείου. Η παρουσίαση μέσα από μία πιλοτική εφαρμογή διδακτικής παρέμβασης με εκπαιδευτικά σενάρια έχει ως στόχους (α) να καταγράψει τον βαθμό εμπέδωσης βασικών εννοιών, απαραίτητων για την παραγωγή κώδικα, από τους μαθητές της Α΄ Λυκείου και (β) να μελετήσει τους τρόπους, με τους οποίους οι μαθητές θα διδαχτούν πώς να δημιουργούν ένα οργανωμένο νοητικό σχήμα και πώς να συνδυάζουν δεδομένα και δομές δεδομένων για να δημιουργήσουν κώδικα.

## Μεθοδολογία Βιβλιογραφικής Ανασκόπησης

Η συγγραφή της εργασίας στηρίχθηκε θεωρητικά στη συστηματική βιβλιογραφική ανασκόπηση. Μελετήθηκαν άρθρα που επιλέχθηκαν με κριτήρια τη συνάφειά τους με τα ερευνητικά ερωτήματα και την πρόσφατη δημοσίευσή τους. Συγκεκριμένα, οι κύριες μελέτες που περιλαμβάνονται στη βιβλιογραφική ανασκόπηση έχουν διεξαχθεί μεταξύ των ετών 2008 και 2016.

## Δυσκολίες στην Εκμάθηση του Προγραμματισμού

Η διερεύνηση των παραγόντων που καθιστούν την εκμάθηση του Προγραμματισμού Η/Υ ένα δύσκολο εγχείρημα, καταλαμβάνει αυτό το σημείο της έρευνας. Οι δυσκολίες που αντιμετωπίζει ένας μαθητής συναντώνται σε κάθε είδους προγραμματισμό, αφού θα πρέπει να μάθει να εκφράζει αυτό που επιθυμεί να κάνει η μηχανή με πολύ σαφή τρόπο και να αναπαριστά διαδικασίες και δεδομένα του φυσικού κόσμου μέσα στο πρόγραμμα. Οι δυσκολίες εντοπίζονται, κυρίως, στη διδακτική του Προγραμματισμού με γλώσσες γενικού σκοπού. Η διερεύνηση και η καταγραφή τους θα βοηθήσει να εξηγήσουμε, στη συνέχεια, ποια οφέλη προκύπτουν από τη διδασκαλία μέσω γραφικού και οπτικού προγραμματισμού.

Τα προβλήματα που δυσκολεύουν τόσο τους μαθητές όσο και τους εκπαιδευτικούς κατά την εκμάθηση και διδασκαλία του Προγραμματισμού, αντίστοιχα, απασχολεί τους ερευνητές από δεκαετίες. Ο du Boulay από το 1986 έκανε μία κατηγοριοποίηση των δυσκολιών των μαθητών σε πέντε ομάδες, που μερικώς επικαλύπτονται. Η πρώτη αφορά προβλήματα προσανατολισμού, με την έννοια ότι οι μαθητές δυσκολεύονται να κατανοήσουν τον σκοπό για τον οποίο μαθαίνουν Προγραμματισμό. Η δεύτερη έχει να κάνει με τη δυσκολία κατανόησης της «νοητής μηχανής». Εδώ υπεισέρχονται ζητήματα για το πώς η μηχανή και το πρόγραμμα αναπαριστούν και διαχειρίζονται τα δεδομένα. Για παράδειγμα, μία εντολή εκχώρησης  $a=b$  μέσα σε ένα πρόγραμμα, π.χ. Java, αντιγράφει την τιμή ενός αντικείμενου (ή μεταβλητής) σε ένα άλλο αντικείμενο. Μία κοινή παρανόηση που μπορεί να συμβεί στον μαθητή είναι να θεωρήσει ότι, ταυτόχρονα με το αντικείμενο που

μεταφέρεται/αντιγράφεται, μεταφέρονται/αντιγράφονται και οι ιδιότητές του. Σε αυτήν την ομάδα δυσκολίες συνδέονται με θέματα που δεν είναι άμεσα ορατά και κρύβονται μέσα στη διαδικασία εκτέλεσης του προγράμματος. Τέτοια παραδείγματα είναι αναφορές σε αντικείμενα και μεταβλητές, αυτόματες ενημερώσεις μετρητών μέσα σε βρόχους κ.ά. Η τρίτη σχετίζεται με τις δυσκολίες που έχουν οι μαθητές με τις τοπικές γλώσσες προγραμματισμού και το συντακτικό τους. Η τέταρτη αναφέρεται στη δυσκολία των μαθητών να κατανοήσουν προγραμματιστικές δομές, όπως η επανάληψη. Η περίπτωση λ.χ. που το πλήθος των επαναλήψεων ενός τμήματος του κώδικα ορίζεται από μία μεταβλητή, που ενδεχομένως να αλλάζει τιμή μέσα σε μία επανάληψη, είναι πιο δύσκολα κατανοήσιμη από τον μαθητή, συγκριτικά με την περίπτωση που η δομή επαναλαμβάνεται για έναν σταθερό αριθμό επαναλήψεων. Η πέμπτη αφορά τις καθαυτό δυνατότητες προγραμματισμού ως προς τη σχεδίαση, την υλοποίηση και τον έλεγχο ενός προγράμματος.

Ο *du Boulay* κατάταξε τους τύπους λαθών που κάνουν οι μαθητευόμενοι προγραμματιστές σε τρεις κατηγορίες: τη λανθασμένη εφαρμογή των αναλογιών, τις υπεργενικεύσεις και τις αλληλεπιδράσεις. Μία λανθασμένη κατανόηση και εφαρμογή μίας αναλογίας συμβαίνει όταν ο μαθητής την επεκτείνει περισσότερο από όσο πρέπει. Η έννοια λ.χ. της μεταβλητής μπορεί να διδαχτεί στον μαθητή ως ένα κουτί. Ο μαθητής μπορεί εσφαλμένα, να θεωρήσει ότι μέσα στο κουτί αυτό μπορούν να βρίσκονται ταυτόχρονα πολλές τιμές. Οι υπεργενικεύσεις, στις οποίες συχνά φτάνουν οι μαθητές, συνδέονται με λανθασμένες υποθέσεις τους σχετικά με τη λειτουργία των προγραμματιστικών δομών, με τον χειρισμό των δεδομένων από το πρόγραμμα ή με το ότι η αναπαράσταση των δεδομένων που υποθέτουν ότι έχουν κάνει είναι αυτή που στην πραγματικότητα επιτελείται. Τα λάθη που συνδέονται με την αλληλεπίδραση, μπορούν να αναφέρονται τόσο στην αλληλεπίδραση προγράμματος/μηχανής και χρήστη, όσο και στην αλληλεπίδραση των διαφόρων τμημάτων ενός προγράμματος μεταξύ τους. Στην πρώτη περίπτωση, υπεισέρχονται θέματα εισόδου και εξόδου δεδομένων από ένα πρόγραμμα, το πώς αυτά γίνονται αντικείμενα διαχείρισης και οργάνωσης από τον χρήστη. Στη δεύτερη περίπτωση, που αφορά την αλληλεπίδραση και τη σύνδεση διαφόρων μερών του προγράμματος (υπορουτίνες, συναρτήσεις ή διαδικασίες) τα λάθη αφορούν τη μεταξύ τους επικοινωνία, το πώς αυτή πραγματοποιείται με τη χρήση και τη μεταβίβαση μεταβλητών και αντικειμένων.

Σε σχέση με τις μεταβλητές, ο *du Boulay* ανέδειξε τις δυσκολίες που έχουν οι μαθητές να κατανοήσουν το γεγονός ότι όταν μία νέα τιμή καταχωρείται σε μία μεταβλητή η προηγούμενη διαγράφεται. Επίσης, ενδεχομένως να θεωρούν ότι μπορούν να ανακτήσουν τις τιμές που είχαν καταχωρηθεί στη μεταβλητή πριν από την τελευταία τιμή. Ακόμα, αναδεικνύει τις παρανοήσεις που μπορεί να έχουν οι μαθητές σχετικά με το αν μία μεταβλητή είναι τοπική σε μία διαδικασία ή καθολική (*global*), όπως και με την αρχικοποίηση των μεταβλητών.

Σε μεταγενέστερα χρόνια ποικίλες έρευνες έχουν ασχοληθεί με το να εξηγήσουν τους παράγοντες που καθιστούν την εκμάθηση του Προγραμματισμού ένα δύσκολο θέμα και με το να παρουσιάσουν τις διδακτικές μεθόδους που ακολουθούνται (Eagle & Barnes, 2009; Koulouri et al., 2014; Muratet et al., 2009). Αρκετές από τις μελέτες που έχουν γίνει επικεντρώνονται στα λάθη και τις παρανοήσεις που συνδέονται με τις δομές των γλωσσών προγραμματισμού, κυρίως γύρω από το συντακτικό της κάθε γλώσσας. Ο Kinnunen (2009) καταγράφει προβλήματα κατανόησης που σχετίζονται με το φύλο ή την ικανότητα απομνημόνευσης των μαθητών. Αντικείμενο προβληματισμού είναι και η επιλογή των κατάλληλων διδακτικών προσεγγίσεων με στόχο την αντιμετώπιση αυτών των δυσκολιών. Μερικές έρευνες (Barker et al., 2009; Coull & Duncan, 2011) δείχνουν ότι οι δυσκολίες συχνά

προκύπτουν λόγω ανεπαρκών στρατηγικών μάθησης και ελλιπούς αλληλεπίδρασης μεταξύ μαθητή και δασκάλου.

Πολλές δυσκολίες προκύπτουν από την ίδια τη φύση του αντικειμένου (Coull & Duncan, 2011), εφόσον η προγραμματιστική διαδικασία απαιτεί (α) την κατανόηση των εννοιών της επιστήμης της Πληροφορικής και (β) την ταξινόμηση αυτών των εννοιών και όλων των δεδομένων που απαιτούνται για να λυθεί ένα πρόβλημα. Οι μαθητές θα πρέπει να αποκτήσουν τη δυνατότητα να σκέφτονται με έναν πιο αφηρημένο τρόπο για τη λύση ενός προβλήματος, την ίδια στιγμή που θα πρέπει να είναι πολύ ακριβείς σε σχέση με τις προγραμματιστικές δομές, το συντακτικό του εκάστοτε προγραμματιστικού εργαλείου και τον τρόπο αναπαράστασης των δεδομένων. Ακόμα, σημαντικό εμπόδιο στην εκμάθηση του Προγραμματισμού είναι η έλλειψη ενδιαφέροντος από μέρους του εκπαιδευόμενου.

Οι κύριες δυσκολίες κατά την εκμάθηση αλγοριθμικού/διαδικαστικού Προγραμματισμού σύμφωνα με τους Pillay & Jugoo (2006) εντοπίζονται μεταξύ άλλων: (α) στην κατανόηση της αρχικοποίησης των μεταβλητών, καθώς και στην κατανόηση και υλοποίηση προγραμματιστικών δομών επιλογής και επανάληψης, (β) στην κατανόηση των διαφορών μεταξύ παρόμοιων συντακτικά εκφράσεων, όπως για παράδειγμα η διαφορά μεταξύ του αλφαριθμητικού "200" από το αριθμητικό 200 και (γ) στην κατανόηση και τον τρόπο χρήσης των δεικτών (pointers) και στον δυναμικό καταμερισμό μνήμης.

Πέρα από τις δυσκολίες που αντιμετωπίζουν οι μαθητές στην εκμάθηση του Προγραμματισμού, υπάρχουν και δυσκολίες που αφορούν τους διδάσκοντες και τον τρόπο με τον οποίο αυτοί μεταφέρουν τη γνώση. Ο Kinnunen (2009) αναφέρεται σε τέτοιες δυσκολίες που σχετίζονται με τις μεθόδους, τις διδακτικές πρακτικές, το πρόγραμμα σπουδών ή τη διδακτέα ύλη. Στο ίδιο πνεύμα μία μελέτη του Dale (2006) κατέγραψε τα προβλήματα και τα θέματα που τους δυσκολεύουν περισσότερο και τα κατηγοριοποίησε: (α) σε ζητήματα επίλυσης και σχεδιασμού προβλημάτων, (β) σε γενικά θέματα προγραμματισμού, όπως χρήση παραμέτρων, πινάκων, δεικτών, βρόχων και δομών επιλογής, (γ) σε αντικειμενοστρεφείς δομές και (δ) στην ωριμότητα των μαθητών.

Σύμφωνα με τους Γρηγοριάδου κ.ά. (2009), οι αιτίες στις οποίες οφείλονται οι δυσκολίες των μαθητών συνοψίζονται στις παρακάτω: (α) στην προϋπάρχουσα γνώση των μαθητών που έρχεται σε σύγκρουση με τις έννοιες που διδάσκονται στο μάθημα του Προγραμματισμού, (β) στην εσφαλμένη αντίληψη των μαθητών για τις δυνατότητες του Η/Υ και τα λειτουργικά χαρακτηριστικά του, (γ) στη σημασιολογία και το συντακτικό μίας γλώσσας προγραμματισμού που είναι πιο αυστηρά από τη φυσική γλώσσα με την οποία εκφράζονται οι μαθητές στην καθημερινότητά τους, (δ) σε προσεγγίσεις που ακολουθούνται στη διδακτική του Προγραμματισμού και δίνουν περισσότερο βάρος στο συντακτικό και τη σημασιολογία μίας γλώσσας και λιγότερο σε βασικές προγραμματιστικές έννοιες, (ε) στα προγραμματιστικά περιβάλλοντα που έχουν στη διάθεσή τους οι μαθητές και τους υποχρεώνουν να «μεταφράζουν»/κωδικοποιούν τη λύση ενός προβλήματος σε χαμηλού επιπέδου προγραμματιστικές δομές, υιοθετώντας λεξιλόγιο και συντακτικό που δεν τους είναι οικείο.

Από τα παραπάνω γίνεται εμφανές ότι οι μαθητές αδυνατούν να καταλάβουν ουσιώδεις έννοιες και τα χαρακτηριστικά τους. Αυτό κατ' επέκταση δημιουργεί πρόσθετο εμπόδιο στον τρόπο αναπαράστασης των δεδομένων. Κρίνεται, επομένως, απαραίτητο να επαναξιολογήσουμε και να επανασχεδιάσουμε τη διδακτική του Προγραμματισμού στο Λύκειο.

## Εκπαιδευτικά Περιβάλλοντα Προγραμματισμού

Στον πίνακα 1 γίνεται μία συγκριτική καταγραφή από τα πιο αντιπροσωπευτικά εκπαιδευτικά περιβάλλοντα, που χρησιμοποιούνται στη διδασκαλία του Προγραμματισμού σε μαθητές, σχετικά με την αναπαράσταση δεδομένων, που αυτά υποστηρίζουν. Τα περισσότερα είναι γραφικά ή περιλαμβάνουν συνδυασμό γραφικού μέρους με εντολές κειμένου.

**Πίνακας 1. Συγκριτική καταγραφή, σχετικά με την αναπαράσταση δεδομένων, εκπαιδευτικών προγραμματιστικών περιβαλλόντων. Όπου V=ΝΑΙ, Χ=ΟΧΙ, Λ=Λίστες, Ο=Ουρές, Π=Πίνακες, Σ=Στοιβές, Μ=Maps, PQ=Priority Queues, \*: 12 χρωματικές + 26 αλφαβητικές.**

	Microworlds Pro	Easy Logo	Turtle Art	Star Logo TNT	Scratch-2	BYOB	App Inventor	Alice	Kodu	Game Maker
Σταθερές	V	V	V	V	V	V	V	V	V	V
Μεταβλητές	V	X	2	V	V	V	V	V	12+26*	V
Δομές Δεδομένων Διαδικασίες χωρίς παραμέτρους	Λ	X	∩	Λ	Λ	Λ	Λ	Λ Π	X	Λ Ο Π Σ Μ Ρ Q
Διαδικασίες με παραμέτρους	V	V	V	V	V	V	V	V	V	X
Περιβάλλον 2D ή 3D	2	2	2	3	2	2	2	3	3	2
Δυνατότητα εισαγωγής πολυμεσικών στοιχείων στο υπόβαθρο Πλήθος αντικειμένων που μπορούν να χρησιμοποιηθούν	V	∩	X	V	V	V	V	V	X	V
Δυνατότητα προγραμματισμού οδηγούμενου από τα γεγονότα (event-driven programming)	N	1	1	N	N	N	∩	N	N	N
Δυνατότητα προγραμματισμού οδηγούμενου από τα γεγονότα (event-driven programming)	V	X	∩	∩	V	V	∩	∩	V	∩

## Οπτικός Προγραμματισμός

Η εκμάθηση του Προγραμματισμού και η ανάπτυξη προγραμμάτων σε ένα περιβάλλον οπτικού/γραφικού προγραμματισμού διαφέρουν από τις αντίστοιχες σε ένα περιβάλλον προγραμματισμού κειμένου. Οι διαφορές είναι συντακτικές και εννοιολογικές. Ο οπτικός προγραμματισμός χρησιμοποιεί οπτικό συντακτικό, δηλαδή εικόνες ή άλλες

αναπαραστάσεις, όπως κουμπιά ή γραφικά. Το οπτικό συντακτικό συχνά περιλαμβάνει οπτικά στοιχεία, όπως χρώμα, βάθος και χωροταξικό σημείο για να εννοιοποιήσει ένα χαρακτηριστικό (Ahmad, 1999).

Όλα τα παραπάνω δομικά συστατικά είναι μορφές αναπαράστασης δεδομένων, των οποίων ο μαθητής κατακτά τη σημασία και τον τρόπο οργάνωσης μέσα στο οπτικό περιβάλλον. Ένα τέτοιο περιβάλλον χρησιμοποιεί οπτικές αναπαραστάσεις αντικειμένων, μερικές φορές σε συνδυασμό με κείμενο, για να αποδώσει κάτι που διαφορετικά θα έπρεπε να γραφεί σε κείμενο προγράμματος. Επίσης, ένα οπτικό περιβάλλον περιλαμβάνει γραφικά εργαλεία και γραφικές τεχνικές για να αποδώσει τη δομή ενός προγράμματος και τις δομές δεδομένων καθώς και για να αλληλεπιδρά με τον προγραμματιστή. Ο προγραμματιστής δεν χρειάζεται να απομνημονεύει και να θυμάται κάθε φορά τις εντολές που θα χρησιμοποιήσει.

### **Προγραμματιστικά Περιβάλλοντα με Πλακίδια**

Τα προγραμματιστικά περιβάλλοντα με πλακίδια, ως εργαλεία εκμάθησης οπτικού Προγραμματισμού, έχουν κερδίσει μεγάλο έδαφος τα τελευταία χρόνια. Πρόκειται για περιβάλλοντα γραφικού προγραμματισμού, στα οποία οι εντολές έχουν τη μορφή πλακιδίων (τουβλάκια ή blocks), με τα οποία οι μαθητές μπορούν να φτιάξουν μία αλληλουχία/πρόγραμμα, ενώνοντας το ένα πλακίδιο με το άλλο, όπως τα κομμάτια ενός πάζλ. Τα πλακίδια, κυρίως μέσω του σχήματός τους, παρέχουν περιορισμούς στον τρόπο σύνδεσης, προκειμένου να εξασφαλιστεί ότι οι μαθητές θα τα ενώσουν με σωστό τρόπο, ώστε να αποφευχθούν συντακτικά λάθη. Με αυτόν τον τρόπο η συνδεσμολογία και τα εργαλεία που προσφέρονται στον μαθητή του δίνουν μία νύξη για το πού και πώς θα πρέπει να τοποθετεί τις εντολές. Η διαδικασία προγραμματισμού σε αυτά τα περιβάλλοντα έχει τη μορφή συρσίματος των πλακιδίων σε μία περιοχή σύνθεσης και συναρμολόγησής τους, ώστε να δημιουργηθούν διαδικασίες και προγράμματα. Αν δύο πλακίδια δεν μπορούν να συνδεθούν, επειδή τα σχήματά τους δεν ταιριάζουν, τότε πρόκειται για συντακτικό λάθος, που το περιβάλλον δεν (πρέπει να) επιτρέπει. Έτσι, οι μαθητές μαθαίνουν να δομούν ένα πρόγραμμα τοποθετώντας μία-μία τις εντολές και αποφεύγοντας τα συντακτικά λάθη.

Το σχήμα των πλακιδίων και ο τρόπος που συνδυάζονται δεν είναι το μόνο χαρακτηριστικό που υπαγορεύει στους μαθητές πώς θα τα οικοδομήσουν. Το χρώμα κάθε ομάδας πλακιδίων και ειδικά εμφωλευμένα πλακίδια είναι πρόσθετοι τρόποι αναγνώρισής τους. Αυτό το χαρακτηριστικό των προγραμματιστικών περιβαλλόντων με πλακίδια τα καθιστά κατάλληλα για τη διδασκαλία Προγραμματισμού σε αρχάριους μαθητές (Resnick et al., 2009). Από την άλλη πλευρά, μερικοί ερευνητές είναι της άποψης ότι αυτή η προσέγγιση δεν λύνει το πρόβλημα με το συντακτικό των γλωσσών προγραμματισμού, απλώς το μεταθέτει (Parsons & Haden, 2007; Powers et al., 2007). Ωστόσο, τα προγραμματιστικά περιβάλλοντα με πλακίδια γίνονται πιο εύκολα αντιληπτά από τους μαθητές, οι οποίοι τα χαρακτηρίζουν πιο ενδιαφέροντα, πιο φιλικά και ελκυστικά (Armoni et al., 2015; Malan & Leitner, 2007; Maloney et al., 2008; Weintrop & Wilensky, 2015a; Weintrop & Wilensky, 2015b; Wilson & Moffat, 2010).

### **Ο ρόλος των Δεδομένων στον Προγραμματισμό**

Ο ακριβής ορισμός των δεδομένων εξαρτάται σε μεγάλο βαθμό από τη γλώσσα προγραμματισμού που θα χρησιμοποιηθεί. Ανεξάρτητα από αυτήν όμως, υπάρχουν νοητές ομαδοποιήσεις των δεδομένων αυτών. Τόσο η κατανόηση των τύπων και των δομών δεδομένων, όσο και η αναπαράστασή τους, είναι ζωτικής σημασίας για την ορθή εκμάθηση

του Προγραμματισμού. Είναι έννοιες με τις οποίες ο μαθητής έρχεται ή θα έπρεπε να έρχεται σε επαφή από τα πρώτα κιόλας μαθήματα Προγραμματισμού.

Μία πρώτη κατηγοριοποίηση των δεδομένων είναι οι σταθερές και οι μεταβλητές. Οι σταθερές είναι συγκεκριμένες τιμές, όχι απαραίτητα αριθμητικές, που ορίζονται μέσα σε ένα πρόγραμμα. Οι μεταβλητές είναι θέσεις στη μνήμη του υπολογιστή, στις οποίες καταχωρούνται τιμές, οι οποίες είναι δυνατόν να αλλάζουν κατά τη διάρκεια εκτέλεσης του προγράμματος. Σημειώνουμε ότι στην Python οι μεταβλητές αντιμετωπίζονται με διαφορετικό τρόπο. Ωστόσο, ενδέχεται να συναντήσουμε σε προγράμματα μεταβλητές των οποίων οι τιμές δεν αλλάζουν, που ουσιαστικά χρησιμοποιούνται ως σταθερές.

Ο τύπος των δεδομένων εξαρτάται από το περιβάλλον προγραμματισμού. Οι πιο συνηθισμένοι τύποι που αφορούν σταθερές και μεταβλητές είναι: (α) ακέραιος τύπος: για την αναπαράσταση ακεραίων αριθμών, (β) πραγματικός τύπος: για την αναπαράσταση πραγματικών αριθμών, (γ) λογικός τύπος: για την αναπαράσταση λογικών δεδομένων και (δ) αλφαριθμητικός τύπος: για την αναπαράσταση αλφαριθμητικών δεδομένων. Ανάλογα με το αν οι μεταβλητές έχουν μία απλή τιμή ή συνδυασμό τιμών, διακρίνονται σε απλές μεταβλητές και σε δομές δεδομένων, δηλαδή αποθηκευμένα δεδομένα, οργανωμένα έτσι ώστε να μπορούν να υποστούν συγκεκριμένες μόνο επεξεργασίες.

### Αναπαράσταση Δεδομένων σε Κειμενικό Περιβάλλον

Σε αυτό το σημείο κρίνεται σκόπιμο να παρουσιαστούν αποτελέσματα ερευνών σχετικών με την αναπαράσταση δεδομένων σε κειμενικό περιβάλλον ως ένας δείκτης του τρόπου κατανόησης των δεδομένων και της ικανότητας αναπαράστασής τους από την πλευρά των μαθητών, προκειμένου να παρουσιαστεί μία πληρέστερη εικόνα των ζητούμενων της παρούσας έρευνας.

Ξεκινώντας από την απλή εντολή εκχώρησης τιμής σε μία μεταβλητή, οι έννοιες που πρέπει να αφομοιώσουν οι μαθητές διαφέρουν από τις προηγούμενες γνώσεις τους, όπως π.χ. η αντίληψη που έχουν από τα Μαθηματικά. Σε μελέτη των Δουκάκη και Γρηγοριάδου (2010), αναδεικνύεται αυτό το σημείο, μέσα από την ανάλυση της εννοιολογικής αλλαγής της μεταβλητής στους μαθητές. Ένα από τα σημεία που επισημαίνονται είναι η εννοιολογική διαφορά αναφορικά με την απόδοση τιμής στον προγραμματισμό με τη μαθηματική ισότητα, για την οποία σημειώνεται ότι ισχύει η αντιμεταθετική ιδιότητα που δεν ισχύει για τον Προγραμματισμό. Επίσης, γίνεται αναφορά σε εφαρμογή αριθμητικών τελεστών σε αλφαριθμητικές μεταβλητές και στη σύγχυση που πιθανόν να δημιουργηθεί στους μαθητές, αν θεωρήσουν ότι όταν αποδώσουν μίαν ακέραια τιμή σε πραγματική μεταβλητή, αυτή θα αποθηκευτεί ως ακέραια.

Δύσκολη υπόθεση είναι για τους μαθητές η κατανόηση της χρήσης λογικών τελεστών και η χρήση λογικών μεταβλητών. Σε έρευνα που διεξήχθη σε μαθητές της Γ' Λυκείου στην Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον, έγινε καταγραφή των δυσκολιών των μαθητών σχετικά με τις λογικές διαδικασίες, τους τελεστές και τις μεταβλητές (Αδαμόπουλος, 2010). Στη συγκεκριμένη έρευνα διαπιστώθηκε ότι ιδιαίτερη σημασία χρειάζεται να δοθεί στο πώς οι μαθητές αντιλαμβάνονται και χρησιμοποιούν τον έλεγχο των τιμών μεταβλητών με τη χρήση των λογικών τελεστών (ΚΑΙ, Ή και ΟΧΙ). Για παράδειγμα: αναφέρεται η χρήση του τελεστή ΟΧΙ στην έκφραση «ΟΧΙ  $A > 10$  στη θέση της  $A \leq 10$ , την ΟΧΙ ( $B \geq 1$  ΚΑΙ  $B \leq 20$ ) στη θέση της  $B < 1$  Ή  $B > 20$ , την ΟΧΙ ( $\Phi = "A"$  Ή  $\Phi = "K"$ ) στη θέση της  $\Phi \neq "A"$  ΚΑΙ  $\Phi \neq "K"$ , και την ΟΧΙ ( $Y = 3$  Ή  $Y = 5$  Ή  $Y = 10$ ) στη θέση της  $Y \neq 3$  ΚΑΙ  $Y \neq 5$  ΚΑΙ  $Y \neq 10$ ».

Σημαντικές είναι οι δυσκολίες που αντιμετωπίζουν οι μαθητές με τις σύνθετες δομές δεδομένων. Σε έρευνα που έγινε σε μαθητές της Γ' Λυκείου στα πλαίσια του μαθήματος

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον (Βραχνός & Τζιμογιάννης, 2010), μελετήθηκε ο τρόπος που αντιλαμβάνονται την έννοια του πίνακα και οι δυσκολίες που αντιμετωπίζουν. Τα αποτελέσματα δείχνουν ότι οι περισσότεροι μαθητές δεν είναι σε θέση να επισημάνουν την αναγκαιότητα της χρήσης των πινάκων σε αλγορίθμους που επεξεργάζονται ομοειδή δεδομένα. Αποτιμήθηκε ότι οι μαθητές ήταν δύσκολο να διακρίνουν τη διαφορά μεταξύ του δείκτη ενός στοιχείου του πίνακα και της αντίστοιχης τιμής. Επιπλέον, η πλειοψηφία των μαθητών κατανοούν την έκφραση «A[i]» ως μία οντότητα, ενώ στην πραγματικότητα είναι δύο. Μία άλλη παρανόηση που κάνουν οι μαθητές είναι ότι μία μεταβλητή συνοδεύεται από όλο το ιστορικό των τιμών που της έχουν εκχωρηθεί (du Boulay, 1986).

Τα λάθη που κάνουν οι μαθητεύομενοι προγραμματιστές σε σχέση με την αναπαράσταση δεδομένων σε κειμενικά περιβάλλοντα, όπως η Java, αυξάνουν σε συχνότητα, όταν προσπαθούν να χρησιμοποιήσουν αφηρημένους τύπους δεδομένων, για τους οποίους δεν έχουν προσλαμβάνουσες από τον πραγματικό κόσμο και την καθημερινότητά τους (Chan Mow, 2012). Επίσης, τους δυσκολεύουν οι δείκτες (pointers) και οι αναφορές.

Οι δομές δεδομένων και η αναπαράστασή τους είναι ένα αντικείμενο απαιτητικό όσο και βασικό που δυσκολεύει τους μαθητές σε όλες τις βαθμίδες εκπαίδευσης. Διαχρονικά συναντούν δυσκολία στην κατανόησή τους λόγω της φύσης τους. Πρόκειται για αφηρημένες έννοιες και η προσέγγισή τους απαιτεί υψηλά επίπεδα αναλυτικής σκέψης και ικανότητας επίλυσης προβλημάτων (Garcia & Al-Safadi, 2014). Αυτός είναι και ο λόγος που δημιουργούν την ανάγκη για στοχευμένες διδακτικές παρεμβάσεις.

Μία δυσκολία που αντιμετωπίζει ο μαθητής από την αρχή της φοίτησής του στη Δευτεροβάθμια Εκπαίδευση είναι η κατανόηση της έννοιας «μεταβλητή» ασχέτως του προγραμματιστικού περιβάλλοντος που χρησιμοποιείται (Δεϊμέζης & Γυφτοδήμος, 2016). Τέλος, η Περδικούρη (2016) επισημαίνει ότι οι μαθητές συναντούν δυσκολία στην ονομασία παραμέτρων και στο πέρασμα παραμέτρων/ορισμάτων, όταν γίνεται κλήση μίας συνάρτησης μέσα σε ένα πρόγραμμα.

Γενικό συμπέρασμα είναι πως οι μαθητές έρχονται αντιμέτωποι με ένα εύρος δυσκολιών στην κατανόηση και στην αναπαράσταση δεδομένων σε κειμενικά περιβάλλοντα. Ο βαθμός κατανόησης των δεδομένων και της αναπαράστασής τους σε κειμενικά περιβάλλοντα εμφανίζεται στις έρευνες ελλιπής, αν όχι μηδενικός. Θα ερευνηθούν στη συνέχεια τα ίδια ζητούμενα στη χρήση περιβαλλόντων οπτικού προγραμματισμού.

## **Αναπαράσταση Δεδομένων σε Οπτικό Περιβάλλον**

Από τη βιβλιογραφική επισκόπηση μελετών σχετικών με τη διδακτική του Προγραμματισμού και την αναπαράσταση δεδομένων παρατηρήθηκε ότι τα τελευταία χρόνια, τόσο στην Ελλάδα όσο και στο εξωτερικό, οι περισσότερες εστιάζουν στα οπτικά περιβάλλοντα, κυρίως Scratch και λιγότερο App Inventor, Kodu Game Lab και Alice, ιδίως όταν πρόκειται για διδασκαλία μαθητών.

Εργασίες που διερεύνησαν τον βαθμό κατανόησης και αναπαράστασης δεδομένων σε γραφικά περιβάλλοντα πραγματοποιήθηκαν σε μαθητές ηλικιακού εύρους που καλύπτει και τις τρεις βαθμίδες εκπαίδευσης, Δημοτικό (Dongsoo et al., 2010; Wilson & Moffat, 2010; Σεραλίδου, 2013), Γυμνάσιο και την αντίστοιχη βαθμίδα στο εξωτερικό (Grover & Pea, 2013; Weintrop & Wilensky, 2015a; Weintrop & Wilensky, 2015b; Φουντουλάκη, 2011), Λύκειο (Armoni et al., 2015; Σαρηπαλίδης, 2012; Φουντουλάκη, 2011; Χασανίδης & Μπράττισης, 2010) και Πανεπιστήμιο (Νικολός, 2010; Φεσάκης & Σεραφείμ, 2009).

Σε έρευνα που πραγματοποιήθηκε από το Ινστιτούτο Τεχνολογίας (MIT) και αφορούσε τις προγραμματιστικές εμπειρίες νέων 8-18 ετών, σε μία περίοδο 18 μηνών, αναδείχθηκε η



δυσκολία των μαθητών στη χρήση συγκεκριμένων δομών δεδομένων (Maloney et al., 2008). Η χρήση εξεζητημένων δομών δεδομένων, όπως είναι οι μεταβλητές, οι τυχαίοι αριθμοί και η λογική Boole, τις οποίες δύσκολα ανακαλύπτει από μόνος του ο μαθητής, αν δεν τις διδαχθεί, δεν ήταν συχνή, αλλά αυξήθηκε με την πάροδο του χρόνου. Παρατηρήθηκε ότι τέτοιου είδους διεργασίες και δομές δεν είναι εύκολο να αφομοιωθούν και να χρησιμοποιηθούν από τους μαθητούμενους χωρίς τις υποδείξεις και την καθοδήγηση ενός διδάσκοντα.

Οι Φεσάκης και Σεραφείμ (2009) ερευνήσαν τη διδασκαλία του Scratch στους φοιτητές του Τμήματος των Επιστημών της Προσχολικής Αγωγής και του Εκπαιδευτικού Σχεδιασμού (ΤΕΠΑΕΣ) του Πανεπιστημίου Αιγαίου, προκειμένου να εξοικειωθούν με τη χρήση του και να παραγάγουν διαδραστικό ψηφιακό εκπαιδευτικό υλικό. Τα προβλήματα που αντιμετώπισαν οι φοιτητές αφορούσαν τη διάδραση, τη διαχείριση του ήχου, τον συγχρονισμό και την κίνηση. Με μικρότερη συχνότητα παρουσιάστηκαν προβλήματα σχετικά με την εκσφαλμάτωση, τις σκηνές, με διαδικαστικά, τις μεταβλητές και το σύστημα αντικειμένων.

Έρευνα του Νικολού (2010) με χρήση του Scratch σε τριτοετείς φοιτητές του Τμήματος Επιστημών της Εκπαίδευσης και της Αγωγής στην Προσχολική Ηλικία του Πανεπιστημίου Πατρών, έδειξε ότι, αναφορικά με την αναπαράσταση δεδομένων, οι εκπαιδευόμενοι συνάντησαν δυσκολίες στις μεταβλητές, ενώ χρησιμοποίησαν με μεγαλύτερη ευκολία τον μηχανισμό μηνυμάτων.

Σε πτυχιακή εργασία, που εκπονήθηκε στο Τμήμα Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς, σχετική με τη διδασκαλία οπτικού Προγραμματισμού με το περιβάλλον Scratch σε μαθητές Γυμνασίου και Λυκείου (Φουντουλάκη, 2011), σημειώθηκε ότι οι μαθητές εξοικειώθηκαν εύκολα με το προγραμματιστικό περιβάλλον και έμαθαν να χρησιμοποιούν τις μεταβλητές και τις βασικές προγραμματιστικές δομές.

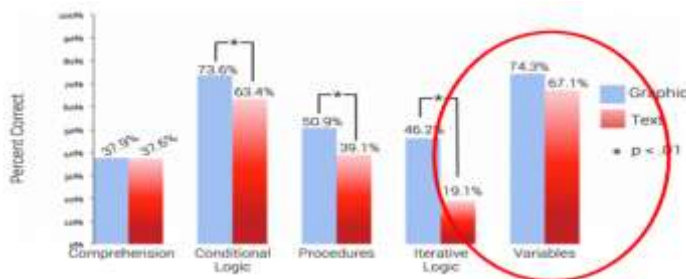
Συμπεράσματα για την αναπαράσταση δεδομένων μπορούμε να αντλήσουμε από την εργασία του Σαρημπαλίδη (2012) για τη μελέτη της διδασκαλίας Προγραμματισμού σε μαθητές της Α΄ Λυκείου με Scratch. Οι μαθητές χωρίστηκαν σε δύο περίπου ισοπληθείς ομάδες, μία πειραματική (με πολύ μικρή εμπλοκή του καθηγητή) και μία ομάδα ελέγχου (με εισήγηση από τον καθηγητή). Η έρευνα σκόπευε να διερευνήσει τον βαθμό εκμάθησης του Προγραμματισμού και να εντοπίσει δυσκολίες των μαθητών. Τα αποτελέσματα έδειξαν ότι οι μαθητές χρησιμοποίησαν σε μεγάλο βαθμό έννοιες πολυνηματικού προγραμματισμού και συγχρονισμού και σε ικανοποιητικό βαθμό έννοιες των δομών επιλογής και επανάληψης. Μεταβλητές χρησιμοποιήθηκαν αρκετά μόνο από την πειραματική ομάδα. Όπως και σε άλλες έρευνες, η χρήση τυχαίων αριθμών, λογικών δομών και διαχείρισης γεγονότων χρησιμοποιήθηκαν πολύ λίγο. Από τα δομικά στοιχεία του Scratch χρησιμοποιήθηκαν σε ικανοποιητικό βαθμό οι σκηνές, τα αντικείμενα και τα σενάρια, αλλά όχι οι μεταβλητές.

Η μελέτη της Σεραλίδου (2013) σε μαθητές της Ε΄ και της ΣΤ΄ τάξης Δημοτικού σχολείου στη Νίκαια σε διδασκαλία του Προγραμματισμού με το γραφικό προγραμματιστικό περιβάλλον Kodu Game Lab, έδειξε ότι οι μαθητές χαρακτήρισαν το περιβάλλον Kodu πρωτότυπο στην εμφάνιση και εύκολο στην κατανόηση, τη χρήση και τις διαδικασίες επιλογής, τοποθέτησης και προγραμματισμού των αντικειμένων. Ένα σημαντικό στοιχείο της έρευνας είναι ότι οι μαθητές της ΣΤ΄ τάξης που είχαν διδαχθεί κατά το προηγούμενο σχολικό έτος Προγραμματισμό (με Scratch) έδειξαν ότι προτιμούν περισσότερο το Kodu και ότι το βρήκαν ευκολότερο και πιο ενδιαφέρον.

Σημαντική είναι η εργασία των Armoní, Meerbaum-Salant και Ben-Ari (2015), σε 120 μαθητές, 15-16 ετών, από τέσσερα σχολεία στο Ισραήλ. Οι μαθητές ήταν χωρισμένοι σε πέντε τμήματα, από τα οποία τα δύο δεν είχαν διδαχθεί στο παρελθόν Προγραμματισμό (ομάδα

ελέγχου), ενώ στα άλλα τρία κάποιοι μαθητές είχαν διδαχθεί κατά το προηγούμενο έτος Scratch (πειραματική ομάδα). Η μελέτη περιλάμβανε μία πεντάμηνη διδασκαλία Προγραμματισμού στο γραφικό περιβάλλον Scratch, ακολουθούμενη από αξιολόγηση. Τα αντικείμενα που διδάχτηκαν οι μαθητές ήταν μεταβλητές, δομή επιλογής, δομές επανάληψης με σταθερό αριθμό επαναλήψεων και επαναλήψεις με λογικό έλεγχο. Τα αποτελέσματα της μελέτης έδειξαν ότι, ενώ για τα διδακτικά αντικείμενα των μεταβλητών και της δομής επιλογής δεν υπήρχε στατιστικά σημαντική διαφορά ανάμεσα στις δύο ομάδες, για τις δομές επανάληψης (σταθερών και μεταβλητών επαναλήψεων) υπήρχε σημαντική διαφορά σε όλα τα γνωστικά επίπεδα, με την πειραματική ομάδα να πετυχαίνει καλύτερες επιδόσεις. Από την ποιοτική ανάλυση προέκυψε ότι οι μαθητές είχαν περιορισμούς στην κατανόηση των εννοιών, όπως μεταβλητές και έλεγχοι, και τις κατανοούσαν κυρίως για το παράδειγμα που έκαναν, αλλά δεν μπορούσαν να τις γενικεύσουν και σε άλλα παραδείγματα.

Μία συγκριτική μελέτη προγραμματιστικών περιβαλλόντων πραγματοποίησαν οι Weintrop και Wilensky (2015) σε οπτικό περιβάλλον με πλακίδια (κυρίως Scratch και Alice), σε κειμενικό περιβάλλον και σε ένα υβριδικό περιβάλλον, που συνδυάζει τα δύο, σε μαθητές γυμνασίου σε πόλη των ΗΠΑ. Αναφορικά με τις μεταβλητές, η μελέτη έδειξε ότι δεν υπήρχε στατιστικά σημαντική διαφορά στις επιδόσεις των μαθητών, ενώ για άλλα προγραμματιστικά αντικείμενα όπως δομές επιλογής, διαδικασίες και επαναληπτικές δομές η διαφορά υπέρ του γραφικού περιβάλλοντος ήταν σημαντική.



**Σχήμα 1. Σύγκριση εκμάθησης Προγραμματισμού σε γραφικό περιβάλλον και σε περιβάλλον εντολών κειμένου (Weintrop & Wilensky, 2015)**

Από την ανασκόπηση των ερευνών σε οπτικά και κειμενικά περιβάλλοντα προγραμματισμού διαπιστώνουμε ότι δεν καταγράφονται σημαντικές διαφορές στα ερευνητικά ζητούμενα. Ωστόσο, εντοπίζονται κάποιες διαφορές σε σχέση με άλλους παράγοντες, όπως είναι η στάση των μαθητών απέναντι στα διαφορετικά περιβάλλοντα και οι επιδόσεις τους. Τα οπτικά περιβάλλοντα αντιμετωπίζονται από τους μαθητές, ειδικά μικρότερων ηλικιών, με πολύ θετικό τρόπο. Τα θεωρούν φιλικά και ενδιαφέροντα, γεγονός που διευκολύνει και προωθεί τη διδασκαλία του Προγραμματισμού.

## Συμπεράσματα

Στη βιβλιογραφική ανασκόπηση που παρουσιάστηκε διερευνήθηκαν ο βαθμός κατανόησης των δεδομένων και της ορθής αναπαράστασής τους σε διάφορα προγραμματιστικά περιβάλλοντα. Από τις έρευνες που μελετήθηκαν προκύπτει το γενικό συμπέρασμα ότι ο βαθμός ανταπόκρισης των μαθητών είναι ελλιπής, ανεξαρτήτως του προγραμματιστικού

περιβάλλοντος, κειμενικού ή οπτικού, που χρησιμοποιήσαν. Επομένως, οι δυσκολίες που συναντούν οι μαθητές ανάγονται όχι στον τύπο του προγραμματιστικού περιβάλλοντος όσο στη διδακτική προσέγγιση που ακολουθείται. Για την αποτελεσματικότερη αναπαράσταση των δεδομένων προτείνεται η ενίσχυση της διδακτικής τους στη Δευτεροβάθμια Εκπαίδευση με στόχο την ανάπτυξη της υπολογιστικής σκέψης των μαθητών.

## Αναφορές

- Ahmad, R. (1999). Visual Languages: A New Way of Programming. *Malaysian Journal of Computer Science*, 12, 1, 76-81.
- Armoni, M., Meerbaum-Salant, O. & Ben-Ari, M. (2015). From Scratch to “Real” Programming. *Transactions on Computing Education*, 14(4), 25.
- Barker, L. J., McDowell, C. & Kalahar, K. (2009). Exploring factors that influence computer science introductory course students to persist in the major. *SIGCSE, Bull.* 41(1), 153-157.
- Chan Mow, I.T. (2012). Analyses of Student Programming Errors In Java Programming Courses. *Journal of Emerging Trends in Computing and Information Sciences*, 3(5), 739-749.
- Coull, N.J. & Duncan, I.M.M. (2011). Emergent Requirements for Supporting Introductory Programming. *ITALICS*, 10(1), 78-85.
- Dale, N.B. (2006). Most difficult topics in CS1: results of an online survey of educators. *SIGCSE, Bull.* 38, 2, 49-53.
- Dongsoo, N., Yungsik, K. & Taewook, L. (2010). The Effects of Scaffolding - Based Courseware for the Scratch Programming Learning on Student Problem Solving Skill. *Proceedings of the 18th International Conference on Computers in Education*, 723-727. Putrajaya, Malaysia.
- du Boulay, B. (1986). Some Difficulties of Learning to Program. *Journal of Educational Computing Research*, 2(1), 57-73.
- Eagle, M. & Barnes, T. (2009). Experimental evaluation of an educational game for improved learning in introductory computing. *SIGCSE, Bull.* 41, 1, 321-325.
- Garcia, R.A. & Al-Safadi, L.A. (2014). Intervention Strategies for the Improvement of Students’ Academic Performance in Data Structure Course. *International Journal of Information and Education Technology*, 4(5), 383.
- Grover, S. & Pea, R. (2013). Using a Discourse - Intensive Pedagogy and Android’s App Inventor for Introducing Computational Concepts to Middle School Students. *Proceedings of the 44th ACM Technical Symposium on Computer Science Education*, 723-728. Denver, Colorado, USA.
- Kinnunen, P. (2009). *Challenges of Teaching and Studying Programming at a University of Technology - Viewpoints of Students, Teachers and the University*. Doctoral Dissertation, Helsinki University of Technology.
- Koulouri, T., Lauria, S. & Macredie, R.D. (2014). Teaching Introductory Programming: A Quantitative Evaluation of Different Approaches. *ACM Transactions on Computing Education*, 14(4), 26.
- Malan, D. & Leitner, H. (2007). Scratch for Budding Computer Scientists. *Proceedings of the 38th SIGCSE technical symposium on Computer science education*, 223-227. Covington, Kentucky, USA.
- Maloney, J., Peppler, K., Kafai, Y., Resnick, M. & Rusk, N. (2008). Programming by Choice: Urban Youth Learning Programming with Scratch. *Proceedings of the 39th SIGCSE technical symposium on Computer science education*, 367-371. Portland, OR, USA.
- Muratet, M., Torguet, P., Jessel, J.P. & Viallet, F. (2009). Towards a serious game to help students learn computer programming. *International Journal of Computer Games Technology*, 1-12.
- Parsons, D. & Haden, P. (2007). Programming osmosis: Knowledge transfer from imperative to visual programming environments. *In Conference of the National Advisory Committee on Computing Qualifications*.
- Pillay, N. & Jugoo V.R. (2006). An Analysis of the Errors Made by Novice Programmers in a First Course in Procedural Programming in Java. *Full research paper in the Proceedings of SACLA*, Van Belle J. & Brown I. (eds.), 11-20.
- Powers, K., Ecott, S. & Hirshfield, L.M. (2007). Through the looking glass: teaching CS0 with Alice. *ACM SIGCSE, Bull.* 39, 1, 213-217.

- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B. & Kafai, Y. (2009). Scratch: Programming for All. *Communications of the ACM*, 52(11), 60-67.
- Weintrop, D. & Wilensky, U. (2015a). Using Commutative Assessments to Compare Conceptual Understanding in Blocks-based and Text-based Programs. *Proceedings of the 11th annual International Conference on International Computing Education Research*, 101-110. Omaha, Nebraska, USA.
- Weintrop, D. & Wilensky, U. (2015b). To Block or not to Block, That is the Question: Students' Perceptions of Blocks-based Programming. *Proceedings of the 14th International Conference on Interaction Design and Children*, 199-208. New York, NY, USA.
- Wilson, A. & Moffat, D.C. (2010). Evaluating Scratch to introduce younger schoolchildren to programming. Lawrance, J. & Bellamy, R. (eds.). *Proceedings of the 22nd Annual Workshop of the Psychology of Programming Interest Group*, 64-74. Madrid, Spain.
- Αδαμόπουλος, Ν. (2010). Χρησιμοποιώντας τελεστές, εκφράσεις και μεταβλητές λογικού τύπου κατά την ανάπτυξη της αλγοριθμικής σκέψης των μαθητών: δυσκολίες, παρανοήσεις, προτάσεις. *5ο Πανελλήνιο Συνέδριο «Διδακτική της Πληροφορικής»*. Αθήνα.
- Βραχνός, Ε. & Τζιμογιάννης, Α. (2010). Μελέτη των αναπαραστάσεων μαθητών της Γ' Λυκείου για την έννοια του πίνακα χρησιμοποιώντας την ταξινόμια SOLO. *5ο Πανελλήνιο Συνέδριο «Διδακτική της Πληροφορικής»*. Αθήνα.
- Γρηγοριάδου, Μ., Γόγουλου, Α., Γουλή, Ε., Γλέζου, Κ., Μπουμπούκα, Μ., Παπανικολάου, Κ., Τσαγκάνου, Γ., Κανίδης, Ε., Δουκάκης, Δ., Φράγκου, Σ. & Βεργίνης, Η. (2009). *Διδακτικές Προσεγγίσεις και Εργαλεία για τη διδασκαλία της Πληροφορικής*. Αθήνα: Εκδόσεις Νέων Τεχνολογιών.
- Δεϊμέζης, Μ. & Γυφτοδήμος, Γ. (2016). Προς μια ολοκληρωμένη διδασκαλία των εννοιών της μεταβλητής στα διάφορα γνωστικά πεδία, μέσω προγραμματισμού. *8th Conference on Informatics in Education «Η Πληροφορική στην Εκπαίδευση»*. Πειραιάς.
- Δουκάκης, Δ. & Γρηγοριάδου, Μ. (2010). Συνθετικά μοντέλα στην κατανόηση της μεταβλητής και της εντολής απόδοσης τιμής στον Προγραμματισμό. *5ο Πανελλήνιο Συνέδριο «Διδακτική της Πληροφορικής»*. Αθήνα.
- Νικολός, Δ. (2010). Ταυτόχρονα περιβάλλοντα προγραμματισμού: Διδακτικές Προσεγγίσεις. Διπλωματική εργασία στο τμήμα Επιστημών της Εκπαίδευσης και της Αγωγής στην Προσχολική Ηλικία του Πανεπιστημίου Πατρών.
- Περδικούρη, Κ. (2016). Ένα διδακτικό σενάριο για τον ορισμό και χρήση συναρτήσεων σε Python, ολοποιώντας μια αριθμομηχανή. *8th Conference on Informatics in Education «Η Πληροφορική στην Εκπαίδευση»*. Πειραιάς.
- Σαρημαπαλιδης, Ι. (2012). Μάθηση Προγραμματισμού Η/Υ από μαθητές Α' Λυκείου με το Scratch. *6ο Πανελλήνιο Συνέδριο «Διδακτική της Πληροφορικής»*. Φλώρινα.
- Σεραλίδου, Ε. (2013). Οι ΤΠΕ στην Α/θμια Εκπαίδευση - Διδασκαλία προγραμματισμού με το εκπαιδευτικό περιβάλλον Kodu Game Lab - Πρακτική εφαρμογή στην τάξη. *5th Conference on Informatics in Education «Η Πληροφορική στην Εκπαίδευση»*. Πειραιάς.
- Φεσάκης, Γ. & Σεραφείμ, Κ. (2009). Μάθηση προγραμματισμού Η/Υ από εκκολλημένους εκπαιδευτικούς με το Scratch. *1ο Εκπαιδευτικό Συνέδριο «Ένταξη και χρήση των ΤΠΕ στην Εκπαιδευτική Διαδικασία»*. Βόλος.
- Φουντουλάκη, Μ. (2011). *Η συμβολή του Scratch στη διδασκαλία του προγραμματισμού στη Β'/θμια εκπαίδευση*. Διπλωματική εργασία στο τμήμα Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιά.
- Χασανίδης, Δ. & Μπράτισης, Θ. (2010). Μαθήματα αλγοριθμικής σκέψης στη Γ' Λυκείου, με χρήση του Scratch: Μια πρόταση για τη διδασκαλία της δομής επιλογής. Γρηγοριάδου, Μ. (επιμ.). *Πρακτικά του 5ου Συνεδρίου Διδακτικής της Πληροφορικής*, 25-30. Αθήνα.