ePublishing
ΕΘΝΙΚΟ ΚΕΝΤΡΟ ΤΕΚΜΗΡΙΩΣΗΣ
NATIONAL DOCUMENTATION CENTRE

Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση

Τόμ. 1 (2022)

7ο Πανελλήνιο Συνέδριο «Ένταξη και Χρήση των ΤΠΕ στην Εκπαιδευτική Διαδικασία»

Παράλληλες διεργασίες στην υπολογιστική σκέψη μαθητών

Δημήτριος Μπέκος, Αναστάσιος Μικρόπουλος, Αναστάσιος Λαδιάς

Βιβλιογραφική αναφορά:

Μπέκος Δ., Μικρόπουλος Α., & Λαδιάς Α. (2023). Παράλληλες διεργασίες στην υπολογιστική σκέψη μαθητών. *Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση*, *1*, 0615–0628. ανακτήθηκε από https://eproceedings.epublishing.ekt.gr/index.php/cetpe/article/view/5771

https://epublishing.ekt.gr | **e-Εκδότης:** EKT | **Πρόσβαση:** 22/02/2026 00:12:40

Παράλληλες διεργασίες στην υπολογιστική σκέψη μαθητών

Μπέκος Δημήτριος[1], Μικρόπουλος Αναστάσιος[2], Λαδιάς Αναστάσιος[3]
dim.bekos93@gmail.com, amikrop@uoi.gr, ladiastas@gmail.com
[1] Μεταπτυχιακός φοιτητής Παιδαγωγικού τμήματος Δημοτικής Εκπαίδευσης Πανεπιστημίου Ιωαννίνων
[2] Καθηγητής Παιδαγωγικού τμήματος Δημοτικής Εκπαίδευσης Πανεπιστημίου Ιωαννίνων
[3] Δρ. Πληροφορικής στη Δευτεροβάθμια Εκπαίδευση

Περίληψη

Ο παράλληλος προγραμματισμός έχει αρχίζει να απασχολεί την εκπαίδευση στην Πληροφορική κυρίως λόγω της αύξησης της πολυπλοκότητας των προβλημάτων και της τεχνολογικής ανάπτυξης. Η εργασία αφορά τον παράλληλο προγραμματισμό και έχει ως στόχο την αξιολόγηση της χρήσης του σε προγράμματα ανεπτυγμένα από μαθητές που βρίσκονται στις τελευταίες τάξεις της πρωτοβάθμιας εκπαίδευσης. Οι κώδικες που αξιολογούνται είναι γραμμένοι στο περιβάλλον οπτικού προγραμματισμού Scratch, το οποίο υποστηρίζει το προγραμματιστικό υπόδειγμα του παράλληλου προγραμματισμού. Η αξιολόγηση πραγματοποιήθηκε με την ένταξη κάθε κώδικα στα πέντε επίπεδα της ταξινομίας SOLO (Structure of Observed Learning Outcomes). Τα αποτελέσματα της έρευνας έδειξαν πως η πλειοψηφία των μαθητών χρησιμοποιεί τοπολογίες παράλληλου προγραμματισμού μεσαίας πολυπλοκότητας και αναπτύσσει προγραμματιστικούς κώδικες που εντάσσονται κυρίως στο τρίτο από τα πέντε επίπεδα της ταξινομίας SOLO. Ειδικότερα, κατά κύριο λόγο οι μαθητές προγραμματίζουν τμήματα κώδικα που τρέχουν παράλληλα, πυροδοτούνται από διαφορετικές αιτίες και λειτουργούν ανεξάρτητα καθώς δεν επικοινωνούν μεταξύ τους. Από τα αποτελέσματα αυτά αναδεικνύονται οι παράλληλες διεργασίες στην υπολογιστική σκέψη των νεαρών μαθητών, αλλά και η δυναμική για τη διδασκαλία του παράλληλου προγραμματισμού στις τελευταίες τάξεις του δημοτικού σχολείου.

Λέξεις κλειδιά: Παράλληλος Προγραμματισμός, Ταξινομία SOLO, Scratch

Εισαγωγή

Τα τελευταία χρόνια οι ηλεκτρονικοί υπολογιστές αυξάνουν συνεχώς τις δυνατότητές τους για παράλληλη επεξεργασία και υπάρχει ανάγκη για προγραμματιστές που θα σχεδιάζουν παράλληλο κώδικα. Ένα παράλληλο πρόγραμμα αποτελείται από δύο ή περισσότερα τμήματα κώδικα που εκτελούνται ταυτόχρονα, αντίθετα με το σειριακό προγραμματιστικό μοντέλο κατά το οποίο, σε οποιαδήποτε χρονική στιγμή εκτελείται μόνο μια εντολή (Δημακόπουλος, 2015). Η διδασκαλία του παράλληλου προγραμματισμού από μικρή ηλικία αποτελεί ένα πρώτο βήμα προς αυτή την κατεύθυνση για τους προγραμματιστές του μέλλοντος (Gregg, Tychonievich, Cohoon, & Hazelwood 2012). Η ένταξη της διδασκαλίας του παράλληλου προγραμματισμού στο γενικότερο κομμάτι της διδασκαλίας του προγραμματισμού κατά την πρωτοβάθμια εκπαίδευση, φέρνει στην επιφάνεια την ανάγκη για αξιολόγηση των δεξιοτήτων παράλληλης επεξεργασίας από μαθητές δημοτικού. Η παρούσα εργασία προτείνει την αξιολόγηση κώδικα μαθητών δημοτικού, με τη χρήση της ταξινομίας SOLO και του προγραμματιστικού περιβάλλοντος Scratch, στο πρότυπο που προτείνουν οι Λαδιάς και Καρβουνίδης (2021).

Παναγιωτακόπουλος, Χ., Καρατράντου, Α., & Αρμακόλας, Σ. (Επιμ.), *Πρακτικά Εργασιών 7ου Πανελλήνιου Συνεδρίου «Ένταξη και Χρήση των ΤΠΕ στην Εκπαιδευτική Διαδικασία»*, σ. 615-628, Τμήμα Επιστημών της Εκπαίδευσης και Κοινωνικής Εργασίας, Πανεπιστήμιο Πατρών, 16-18 Σεπτεμβρίου 2022. ISBN 978-618-83186-7-0.

Θεωρητικό πλαίσιο

Παράλληλος Προγραμματισμός

Τις τελευταίες δεκαετίες έχουν αναπτυχθεί πολλές γλώσσες προγραμματισμού και ένας τρόπος για την κατανόηση των διαφορών τους είναι ο διαχωρισμός των γλωσσών αυτών σε προγραμματιστικά υποδείγματα. Κατά τον Samuel (2017), οι βασικές κατηγορίες των προγραμματιστικών υποδειγμάτων είναι το προστακτικό, το συναρτησιακό, το αντικειμενοστραφές, το λογικό, το παράλληλο και το υπόδειγμα προγραμματισμού βασισμένο σε γεγονότα. Αυτά τα υποδείγματα φαίνεται να κυριαρχούν στην ανάπτυξη λογισμικού, ενώ πολλές γλώσσες προγραμματισμού συνδυάζουν στοιχεία από παραπάνω από ένα υποδείγματα. Ο Δημακόπουλος (2015) αναφέρει πως κρίσιμο ρόλο τα επόμενα χρόνια θα παίξει το παράλληλο υπόδειγμα προγραμματισμού.

Πολλά επιστημονικά και μηχανικά προβλήματα απαιτούν μεγάλες ποσότητες επιμέρους υπολογισμών, οι οποίοι πρέπει να πραγματοποιούνται σε φυσιολογικά χρονικά περιθώρια (Wilkinson & Allen, 2005). Προς την κατεύθυνση αυτή οι βελτιώσεις γίνονται κυρίως με την παραγωγή ταχυτέρων υλικών σε όλο και λιγότερο χώρο, κυρίως στον επεξεργαστή και την μνήμη μιας υπολογιστικής συσκευής (Δημακόπουλος, 2015). Όμως, οι βελτιώσεις αυτές έχουν φυσικά όρια που οδηγούν σε τεχνολογικά αδιέξοδα. Λύση στα παραπάνω όρια και αδιέξοδα προσπαθεί να δώσει ο παράλληλος προγραμματισμός, που επίσης άρχισε να χρησιμοποιείται λόγω της ανάγκης για αύξηση της υπολογιστικής ισχύος των υπολογιστικών συστημάτων (Wilkinson & Allen, 2005).

Ο Δημακόπουλος (2015) σημειώνει, πως στον παράλληλο προγραμματισμό υπάρχουν δύο ή περισσότεροι επεξεργαστές που εκτελούν δύο ή περισσότερες εντολές ταυτόχρονα, σε αντίθεση με το σειριακό μοντέλο που εκτελείται μία εντολή τη φορά. Αυτή η ταυτόχρονη εκτέλεση διαφορετικών τμημάτων κώδικα οδηγεί σε σημαντική μείωση του χρόνου και της υπολογιστικής ισχύος που απαιτούνται για την επίλυση ενός υπολογιστικού προβλήματος. Για την αποτελεσματική εκτέλεση ενός παράλληλου προγράμματος επισημαίνεται η ανάγκη για συντονισμό και συνεργασία μεταξύ των επεξεργαστών που το εκτελούν. Επιπρόσθετα, η διαμοίραση των εργασιών που εκτελεί κάθε επεξεργαστής εξαρτάται από το ίδιο το πρόβλημα, από το πλήθος των επεξεργαστών, από τον τρόπο επικοινωνίας τους και από τις χρονικές καθυστερήσεις που αυτή συνεπάγεται.

Στο εύρος του παράλληλου προγραμματισμού υπάρχει και μια άλλη έννοια, αυτή του ψευδοπαράλληλου προγραμματισμού (Bal, Steiner, & Tanenbaum., 1989). Ένας επεξεργαστής με μεγάλη ταχύτητα μπορεί να εκτελεί πολλαπλές διεργασίες ταυτόχρονα εναλλάσσοντας ταχύτατα την λειτουργία του από την εκτέλεση μιας διεργασίας στην άλλη και ξανά το ίδιο. Με αυτόν τον τρόπο δημιουργείται η ψευδαίσθηση ότι πολλές διεργασίες εκτελούνται πραγματικά ταυτόχρονα. Αυτή η εναλλαγή διεργασιών κατά την εκτέλεση από έναν επεξεργαστή ονομάζεται ψευδοπαραλληλισμός. Σε πολλές γλώσσες προγραμματισμού δεν γίνεται εμφανές στον προγραμματιστή αν προγραμματίζει παράλληλα ή ψευδοπαράλληλα. Όπως και στον παράλληλο προγραμματισμό, η χρήση του ψευδοπαράλληλου προγραμματισμού συνεισφέρει στην εξοικονόμηση χρόνου, όμως τα προβλήματα συγχρονισμού και επικοινωνίας παραμένουν.

Συχνά, οι τεχνικές επίλυσης προβλημάτων που εγείρονται στον χώρο της επιστήμης των υπολογιστών και οι προγραμματιστικές δεξιότητες του ατόμου, συνδέονται με την υπολογιστική του σκέψη (Aho, 2012). Για την καλλιέργεια της υπολογιστικής σκέψης και των αντίστοιχων δεξιοτήτων τους σε νεαρούς μαθητές, οι προγραμματιστικές γλώσσες που χρησιμοποιούνται κατά κόρον είναι αυτές του οπτικού προγραμματισμού (Sáez-López, Román-González & Vázquez-Cano, 2016). Η χρήση αυτού του τύπου γλωσσών

προγραμματισμού βελτιώνει διάφορα στοιχεία του εκπαιδευόμενου, όπως την εκμάθηση εννοιών προγραμματισμού, τη λογική και την υπολογιστική σκέψη. Μία από τις περισσότερο χρησιμοποιούμενες πλατφόρμες οπτικού προγραμματισμού είναι το Scratch και αυτό γιατί ξεπέρασε σε μεγάλο βαθμό τα προβλήματα των πρώιμων προγραμματιστικών γλωσσών που προορίζονταν για την εκπαίδευση νεαρών μαθητών (Resnick et al., 2009).

Βιβλιογραφική ανασκόπηση διδασκαλίας παράλληλου προγραμματισμού σε μαθητές

Στόχος της βιβλιογραφικής ανασκόπησης ήταν η διερεύνηση δεξιοτήτων παράλληλου προγραμματισμού σε μαθητές. Έγινε αναζήτηση στις ακαδημαϊκές βάσεις Scopus, ACM, ScienceDirect, IEEE και Google Scholar από το 2010 μέχρι σήμερα. Ο αλγόριθμος αναζήτησης ήταν (("parallel programming" OR "parallel computing" OR "concurrent computing" OR "concurrent programming" OR "parallel outcomes") AND (education OR teaching OR lesson) AND (primary OR elementary OR secondary OR children OR young OR beginners)). Η αναζήτηση επέστρεψε οκτώ άρθρα.

Από τις μελέτες που αφορούν τη διδασκαλία παράλληλου προγραμματισμού σε μαθητές νεαρής ηλικίας, φαίνεται πως οι εκπαιδευόμενοι είναι δυνατό να χρησιμοποιήσουν στοιχεία παραλληλίας στην υπολογιστική τους σκέψη και να τα εμφανίσουν στους προγραμματιστικούς τους κώδικες. Κατά κύριο λόγο, αυτά εμφανίζονται με επιτυχία αφού οι μαθητές διδαχθούν έννοιες παράλληλου προγραμματισμού σε ηλικίες του δημοτικού (Gregg et al., 2012; Sáez-López et al., 2016; Fatourou et al., 2018; Weintrop et al., 2018) αλλά και ηλικίες γυμνασίου και λυκείου (Feldhausen et al., 2014; Pellas, 2014). Μάλιστα, οι μαθητές έχουν ιδέες που παραπέμπουν στην παράλληλη επεξεργασία και υπάρχει πιθανότητα να την προτιμούν σε σχέση με τη δομή επανάληψης (Tarkan et al., 2010). Επιπρόσθετα, οι μαθητές παράγουν παράλληλα τμήματα κώδικα χωρίς να το καταλαβαίνουν (Weintrop et al., 2018). Ακόμη και μαθητές αρχάριοι στον προγραμματισμό δείχνουν ικανοί να καταλάβουν τι είναι ο παράλληλος προγραμματισμός και να τον εφαρμόσουν σε επίπεδο αρχάριου (Feldhausen et al., 2014; Fatourou, 2018). Παρόλα αυτά, οι μελέτες δείχνουν πως οι ανήλικοι μαθητές δεν έχουν αντίληψη για το τι μπορεί να είναι ο παράλληλος προγραμματισμός εάν δεν έχουν διδαχθεί για αυτόν (Libert & Vanhoof, 2019). Επιπρόσθετα, οι περισσότερες έρευνες χρησιμοποιούν την πλατφόρμα οπτικού προγραμματισμού Scratch για την εκμάθηση στοιχείων παράλληλης επεξεργασίας (Feldhausen et al., 2014; Pellas, 2014; Sáez-López et al., 2016; Fatourou et al., 2018; Weintrop et al., 2018). Το Scratch προσφέρεται για αυτή την διαδικασία, καθώς υπάρχουν τοπολογίες παράλληλου προγραμματισμού σε αυτήν την πλατφόρμα (Λαδιάς & Καρβουνίδης, 2021). Συνοπτικά, οι παραπάνω μελέτες δείχνουν ότι υπάρχουν παράλληλες διεργασίες στην υπολογιστική σκέψη των μαθητών του δημοτικού, οι οποίες ενισχύονται με τη διδασκαλία των εννοιών και της εφαρμογής του παράλληλου προγραμματισμού.

Τα ελάχιστα, αλλά ενδιαφέροντα ευρήματα της ανασκόπησης, αναδεικνύουν την αξία διερεύνησης εμφάνισης παράλληλου προγραμματισμού σε κώδικες μαθητών που βρίσκονται στα πρώτα βήματα διδασκαλίας προγραμματισμού.

Μεθοδολογία

Σκοπός

Σκοπός της παρούσας μελέτης είναι να διερευνηθεί κατά πόσο υπάρχουν παράλληλες διεργασίες στην υπολογιστική σκέψη μαθητών δέκα έως δεκατριών ετών. Αυτό επιτυγχάνεται

ελέγχοντας με μετρήσιμα κριτήρια τον τρόπο με τον οποίο χρησιμοποιούν τον παράλληλο προγραμματισμό σε προγράμματά τους, τα οποία ανέπτυξαν για να ελέγξουν ρομποτικές κατασκευές.

Δείγμα

Στη μελέτη χρησιμοποιήθηκαν 25 προγράμματα για τον έλεγχο αυτοματισμών και των προσομοιώσεών τους από το αποθετήριο έργων της ανοιχτής Κατηγορίας για το Δημοτικό, του Πανελλήνιου Διαγωνισμού Εκπαιδευτικής Ρομποτικής του WRO - Hellas του 2018. Οι μαθητές που ανέπτυξαν τους κώδικες αυτούς είναι ηλικίας δέκα έως δεκατριών ετών ενώ κάθε κώδικας γράφτηκε από ομάδες τριών έως έξι ατόμων στο περιβάλλον οπτικού προγραμματισμού Scratch. Το γνωστικό επίπεδο των μαθητών και η επίδοσή τους στα αντικείμενα που διδάσκονται είναι άγνωστο. Επίσης, για την αναπαράσταση των προγραμμάτων χρησιμοποιήθηκαν, τα αντίστοιχα σε αυτά, κωδικΟράματα. Τέλος, τα προγράμματα που επιλέχθηκαν για αξιολόγηση είναι αυτά που εκτελούνται με επιτυχία σε συνδυασμό με ένα καλά γραμμένο κωδικΌραμα ώστε να υπάρχει η δυνατότητα για ανάλυση του κώδικα των μαθητών.

Το κωδικΌραμα είναι ένας οπτικοποιημένος δισδιάστατος πίνακας που απεικονίζει το σύνολο ενός κώδικα το οποίο έχει γραφτεί στην πλατφόρμα οπτικού προγραμματισμού Scratch (Λαδιάς & Λαδιάς, 2016). Στην οριζόντια γραμμή αυτού του δισδιάστατου πίνακα βρίσκονται όλα τα αντικείμενα που υπάρχουν στον κώδικα, ενώ στην κατακόρυφη βρίσκονται οι καταστάσεις που μπορεί να υπάρξουν τα αντικείμενα ενώ το πρόγραμμα εκτελείται. Έτσι δημιουργείται ο πίνακας με διαστάσεις τον αριθμό των αντικειμένων επί τον αριθμό των καταστάσεων. Κάθε κελί περιέχει το τμήμα κώδικα που εκτελεί ένα αντικείμενο στην αντίστοιχη κατάσταση, ενώ πολλά από τα κελιά που δημιουργούνται υπάρχει πιθανότητα να παραμένουν κενά. Στα κωδικΟράματα επίσης, υπάρχουν διασυνδέσεις που απεικονίζονται με βέλη και δείχνουν την κίνηση της πληροφορίας κατά την επικοινωνία των διαφόρων τμημάτων κώδικα.

Ανάλυση δεδομένων

Η κατηγοριοποίηση των δεδομένων γίνεται μέσω της αξιολόγησης του κώδικα με κριτήριο την παραλληλία. Ειδικότερα, επιτυγχάνεται με την ένταξη κάθε ΚωδικΟράματος σε ένα από τα πέντε επίπεδα της ταξινομίας SOLO με κριτήριο τις τοπολογίες παράλληλου προγραμματισμού που περιέχει, όπως περιεγράφηκαν από τους Λαδιά και Καρβουνίδη το 2021. Η ταξινομία SOLO αποτελείται από πέντε επίπεδα πολυπλοκότητας και προτάθηκε από τους Biggs & Collins (1982) καθώς σημειώνουν πως η γνώση μπορεί να αξιολογηθεί με βάση τη δομή του παρατηρούμενου μαθησιακού αποτελέσματος.

Στο πλαίσιο της παρούσας μελέτης η κατηγοριοποίηση του κώδικα στις πέντε κατηγορίες της ταξινομίας SOLO έχει ως εξής:

- Στο προδομικό επίπεδο εντάσσονται τα προγράμματα που τρέχουν τον κώδικα τους σειριακά, χωρίς να εκτελείται μια διεργασία ταυτόχρονα με κάποια άλλη και τα προγράμματα με κώδικες που δεν δίνουν σημασία στην μη-αιτιοκρατική συμπεριφορά των παράλληλων σεναρίων.
- Στο μονοδομικό επίπεδο εντάσσονται οι κώδικες που τρέχουν παράλληλα αλλά πυροδοτούνται από την ίδια αιτία και αρχίζουν την εκτέλεση τους ταυτόχρονα.
- Οι κώδικες που ενεργοποιούνται από διαφορετική αιτία και δεν ξεκινούν ταυτόχρονα αλλά είναι ανεξάρτητοι, εντάσσονται στο πολυδομικό επίπεδο.

- Στο σχεσιακό επίπεδο περιέχονται οι κώδικες που τρέχουν παράλληλα και έχουν την ανάγκη για μεταξύ τους επικοινωνία, καθώς εξαρτάται ο ένας από τον άλλον.
- Τέλος, στο επίπεδο εκτεταμένης γενίκευσης εντάσσονται οι κώδικες που ο προγραμματιστής έχει επιλύσει προβλήματα μη-αιτιοκρατικής συμπεριφοράς των παράλληλων τμημάτων, όπως η πρόσβαση σε κρίσιμες περιοχές κώδικα και η διαχείριση κοινόχρηστων πόρων.

Για τη διαδικασία εξαγωγής συμπερασμάτων μέσω των αποτελεσματων αντλούνται επιπλέον δεδομένα από τα κωδικΟράματα. Αρχικά, υπολογίζεται για κάθε κωδικΌραμα το μέγεθος του προγράμματος, δηλαδή το άθροισμα των κελιών τα οποία δεν είναι κενά. Στη συνέχεια, σε κάθε κατηγορία της ταξινομίας SOLO υπολογίζεται ο μέσος όρος του μεγέθους αυτού. Επίσης, σε κάθε κωδικΌραμα υπολογίζεται το πλήθος των καταστάσεων του και εν συνεχεία, στο κάθε επίπεδο προσεγγίζεται ο μέσος όρος των καταστάσεων αυτών. Επιπλέον, για κάθε κωδικΌραμα εντοπίζεται το πλήθος διαδικασιών και τελικά ο αντίστοιχος μέσος όρος για κάθε επίπεδο της ταξινομίας. Τέλος, υπολογίζεται το πλήθος των κελιών με περισσότερα του ενός παράλληλα σενάρια του ίδιου αντικειμένου στην ίδια κατάσταση και ο αντίστοιχος μέσος όρος κάθε επιπέδου.

Αποτελέσματα και συζήτηση

Μετά την αξιολόγηση των ΚωδικΟραμάτων με κριτήριο την παραλληλία και την ένταξη τους στις πέντε κατηγορίες της ταξινομίας SOLO, προέκυψαν οι συχνότητες που παρουσιάζει ο Πίνακας 1.

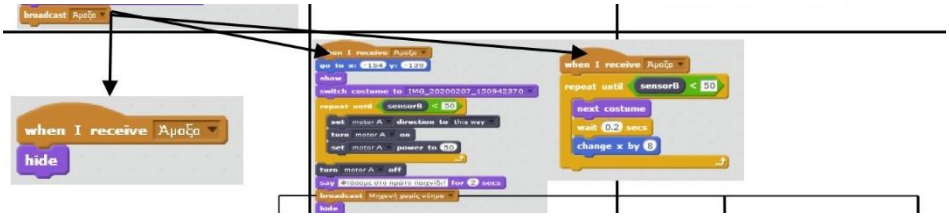
Πίνακας 1. Συχνότητα και ποσοστό ΚωδικΟραμάτων για κάθε κατηγορία της SOLO

Επίπεδο	Συχνότητα	Ποσοστό %
Προδομικό	1	4
Μονοδομικό	6	24
Πολυδομικό	14	56
Σχεσιακό	4	16
Εκτεταμένης γενίκευσης	0	0
Σύνολο	**25**	**100.0**

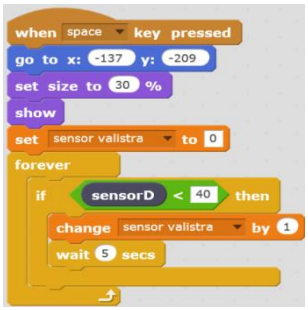
Από τον πίνακα 1 προκύπτει ότι το ποσοστό των ΚωδικΟραμάτων που ανήκουν στο προδομικό επίπεδο της ταξινομίας SOLO είναι μόλις 4%. Ένα τόσο χαμηλό ποσοστό σε αυτή την κατηγορία, δείχνει πως σχεδόν όλοι οι μαθητές δεν χρησιμοποιούν αποκλειστικά σειριακούς κώδικες και πως δεν έχουν περιορισμένη αντίληψη για την διαχείριση του κώδικά τους ως προς την παραλληλία.

Στο μονοδομικό επίπεδο της ταξινομίας προκύπτει ένα ποσοστό 24%. Η ερμηνεία αυτού του ποσοστού είναι πως σε ένα στα τέσσερα ΚωδικΟράματα, παρατηρείται μια περιορισμένη χρήση τοπολογιών παράλληλου προγραμματισμού χωρίς μεταξύ τους συνδέσεις, καθώς λειτουργούν ανεξάρτητα και ξεκινούν ταυτόχρονα. Ένα χαρακτηριστικό παράδειγμα αποτελεί ο κώδικας του σχήματος 1, ο οποίος εντάσσεται στο μονοδομικό επίπεδο. Στο σχήμα αυτό αποτυπώνεται ένα στιγμιότυπο κώδικα, κατά το οποίο η ίδια αιτία πυροδοτεί την ταυτόχρονη εκκίνηση παράλληλων σεναρίων. Επιπρόσθετα, σε αυτό το επίπεδο οι μαθητές χρησιμοποιούν παράλληλα σενάρια, που βρίσκονται σε μια διαρκή επανάληψη έως ότου

ικανοποιηθεί κάποια συνθήκη. Στο σχήμα 2 αποτυπώνεται το στιγμιότυπο ενός ΚωδικΟράματος που δείχνει μια τέτοια ατέρμονη επανάληψη. Η επανάληψη αυτή βρίσκεται σε κατάσταση εκτέλεσης, διαρκεί για πάντα και ελέγχει αν ισχύει κάποια συνθήκη για να εκτελέσει ένα τμήμα κώδικα. Το Scratch δίνει την δυνατότητα να χρησιμοποιηθεί η εντολή "Όταν", ώστε το σενάριο να βρίσκεται σε κατάσταση επαγρύπνησης, αλλά στις περιπτώσεις αυτές οι μαθητές δεν το χρησιμοποιούν.



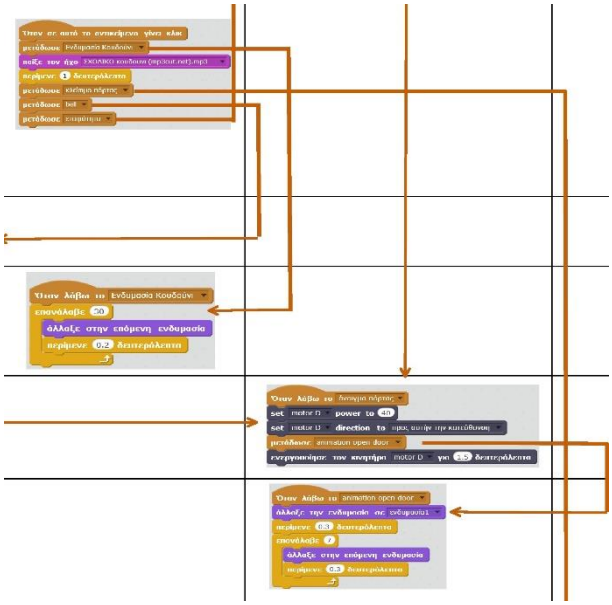
Σχήμα 1. Στιγμιότυπο κώδικα μονοδομικού επιπέδου με την πυροδότηση τριών σεναρίων από την ίδια αιτία



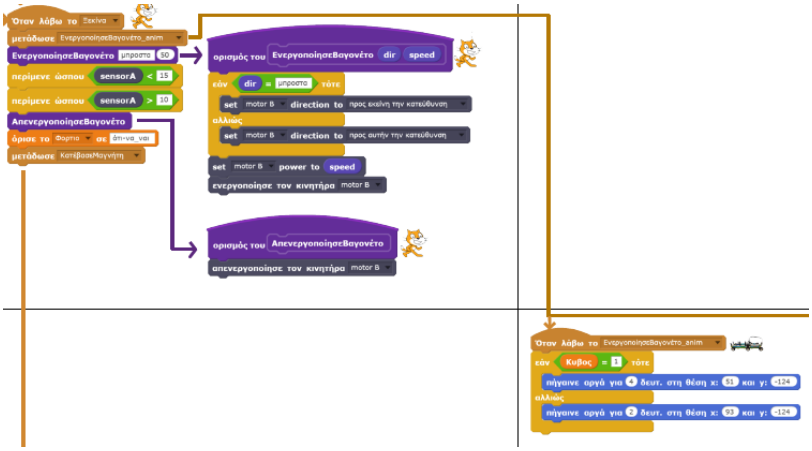
Σχήμα 2. Στιγμιότυπο κώδικα μονοδομικού επιπέδου με ατέρμονη επανάληψη

Στο πολυδομικό επίπεδο βρίσκεται η πλειοψηφία των κωδικΟραμάτων με ένα ποσοστό 56%. Αυτό το ποσοστό δείχνει, πως στην υπολογιστική σκέψη των περισσοτέρων μαθητών υπάρχουν οι διεργασίες για τη χρήση ποικίλων παράλληλων σεναρίων με διάφορους τρόπους, αλλά χωρίς να υπάρχουν σημαντικές συνδέσεις και επικοινωνία μεταξύ τους. Σε αυτή την κατηγόρια το βασικότερο χαρακτηριστικό είναι πως υπάρχουν παράλληλα σενάρια που δεν ξεκινούν ταυτόχρονα, καθώς δεν πυροδοτούνται από την ίδια αιτία. Τέτοιο παράδειγμα αποτελεί το στιγμιότυπο του σχήματος 3. Ανάλογο παράδειγμά αποτυπώνεται στο σχήμα 4, που το πατρικό σενάριο συνεχίζει να εκτελείται παράλληλα με το σενάριο απόγονο, κάτι το οποίο συμβαίνει συχνά στους κώδικες που αξιολογήθηκαν. Επίσης, από το ποσοστό αυτού του επίπεδου φαίνεται πως οι μαθητές στην υπολογιστική τους σκέψη χρησιμοποιούν το «όταν» και όχι το «αν» σε μια παντοτινή επανάληψη, κάτι που πιθανότατα προκύπτει από την καθημερινότητά τους και τον τρόπο σκέψης τους σε αυτή. Χαρακτηριστικό παράδειγμα με την χρήση του «όταν» αποτελεί το στιγμιότυπο που αποτυπώνεται στο σχήμα 5. Όμως, σε πολλές περιπτώσεις που υπήρχε η ανάγκη συγχρονισμού των σεναρίων, αυτή επιτυγχανόταν με ανορθόδοξο τρόπο, δηλαδή με την εντολή «Περίμενε x δευτερόλεπτα», όπου x ένας πραγματικός αριθμός, ώστε να ολοκληρωθεί κάτι πριν συνεχίσει ο κώδικας. Ο συγχρονισμός δεν πραγματοποιούταν με τη χρήση μεταβλητών ή τη χρήση της εντολής «Μετάδωσε και περίμενε», δηλαδή τεχνικές παράλληλου προγραμματισμού που θα

κατηγοριοποίησαν το κωδικΌραμα σε κάποιο παραπάνω επίπεδο της SOLO. Στο σχήμα 6 αποτυπώνεται μια τέτοια περίπτωση που ο κώδικας χρησιμοποιεί στο πατρικό σενάριο την εντολή «περίμενε 0.5 δευτερόλεπτα» μετά την πυροδότηση ενός άλλου σεναρίου ώστε να μην τρέξει παράλληλα με αυτό και να το περιμένει να τελειώσει.

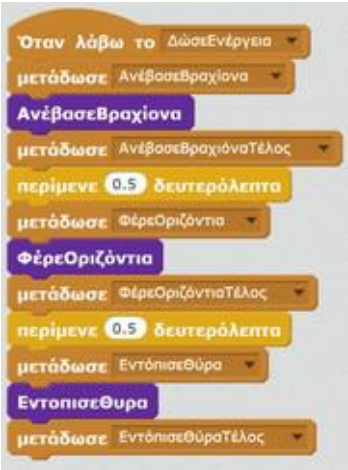


Σχήμα 3. Στιγμιότυπο κώδικα πολυδομικού επιπέδου κατά το οποίο πυροδοτούνται δύο σενάρια από διαφορετικές αιτίες



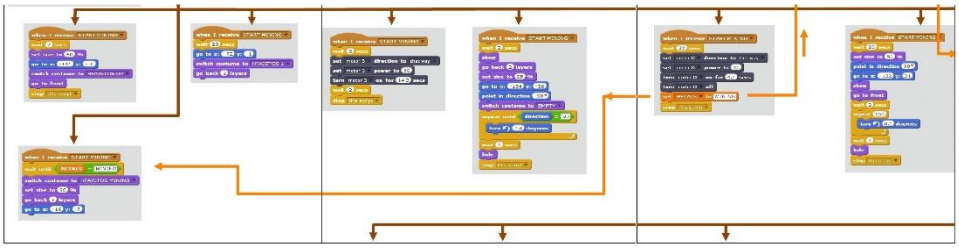
Σχήμα 4. Στιγμιότυπο κώδικα κατά το οποίο το πατρικό σενάριο συνεχίζει να εκτελείται παράλληλα με το σενάριο απόγονο

Σχήμα 5. Στιγμιότυπο κώδικα που γίνεται χρήση της εντολής "Όταν" και όχι της "Αν"



Σχήμα 6. Στιγμιότυπο ΚωδικΟράματος που γίνεται χρήση της εντολής "Περίμενε x δευτερόλεπτα"

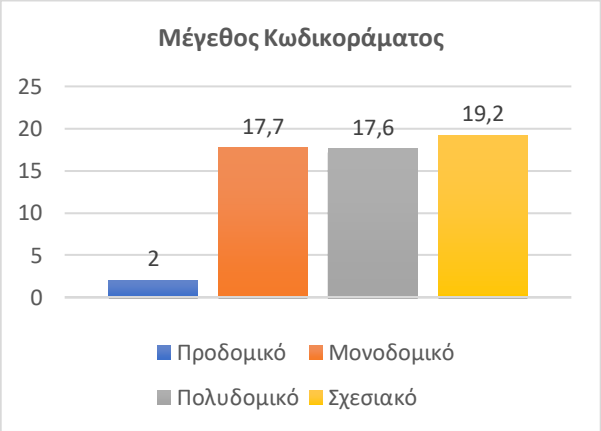
Το ποσοστό κωδικΟραμάτων που ανήκουν στο σχεσιακό επίπεδο της SOLO είναι 16%. Αυτό υποδηλώνει πως κάποιοι μαθητές, παρά τη νεαρή τους ηλικία, εκτιμούν την αναγκαιότητα επικοινωνίας των διαφόρων σεναρίων και την επιτυγχάνουν είτε σύγχρονα είτε ασύγχρονα. Στο σχήμα 7 αποτυπώνεται ένα τμήμα κώδικα, το οποίο εντάσσεται στο σχεσιακό επίπεδο. Στο συγκεκριμένο στιγμιότυπο επιτυγχάνεται ασύγχρονη επικοινωνία, καθώς ένα τμήμα κώδικα περιμένει να αλλάξει τιμή η μεταβλητή «METALS». Αυτή η αλλαγή μεταβλητής πραγματοποιείται σε κάποιο άλλο σενάριο και ακολουθεί η ενημέρωση στο σενάριο που αναμένει. Στο σχήμα 8 αποτυπώνεται το στιγμιότυπο ενός ΚωδικΟράματος, στο οποίο επιτυγχάνεται μια μορφή σύγχρονης επικοινωνίας. Αυτό πραγματοποιείται με τη χρήση της εντολής «Μετάδωσε και Περίμενε», με αποτέλεσμα το πατρικό σενάριο να αναμένει το σενάριο απόγονο να ολοκληρωθεί, ώστε να συνεχίσει τη λειτουργία του. Στους κώδικες αυτούς γίνεται αντιληπτή η σημασία των διαφόρων τμημάτων σε σχέση με το σύνολο του ΚωδικΟράματος.



Σχήμα 7. Στιγμιότυπο ΚωδικΟράματος που επιτυγχάνεται ασύγχρονη επικοινωνία με χρήση μεταβλητής

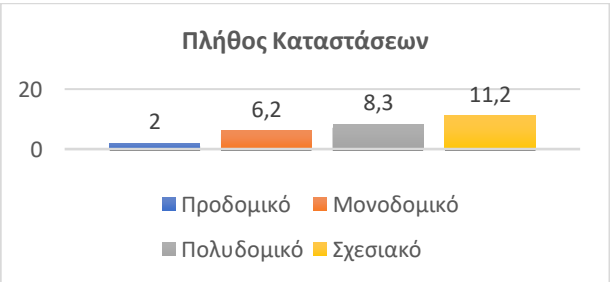
Σχήμα 8. Στιγμιότυπο ΚωδικΟράματος που επιτυγχάνεται σύγχρονη επικοινωνία με χρήση της εντολής "Μετάδωσε και Περίμενε"

Τα αποτελέσματα του σχήματος 9 αναδεικνύουν τον μέσο όρο του μεγέθους (πλήθους κελιών) των κωδικΟραμάτων για κάθε επίπεδο της ταξινομίας SOLO. Μεταξύ του προδομικού επιπέδου και των άλλων επιπέδων της SOLO υπάρχει μια μεγάλη διαφορά. Όμως μεταξύ μονοδομικού, πολυδομικού και σχεσιακού επιπέδου δεν προκύπτει αξιοσημείωτη διαφορά. Στο προδομικό επίπεδο εντάσσεται μόλις ένα κωδικΌραμα, το οποίο έχει έναν πολύ μικρό αριθμό μη κενών κελιών, καθώς όλος ο κώδικας βρίσκεται συμπυκνωμένος σε μόλις δύο κελιά. Ένα τόσο μικρός πλήθος κελιών με σειριακό κώδικα δεν επιτρέπει την παράλληλη λειτουργία ποικίλων σεναρίων και επικρατεί ο δομημένος προγραμματισμός.



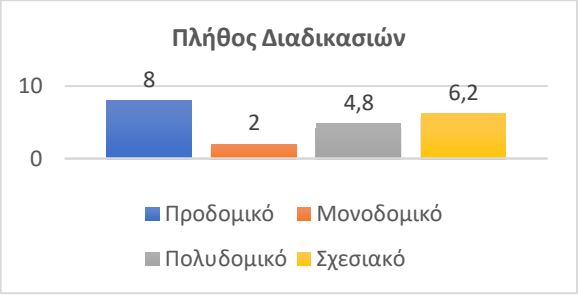
Σχήμα 9. Μέσος όρος για το μέγεθος των κωδικΟραμάτων κάθε επιπέδου της ταξινομίας SOLO

Από το σχήμα 10 προκύπτει, πως σε όσο μεγαλύτερη κατηγορία της ταξινομίας SOLO ανήκει ένα κωδικΌραμα, τόσο μεγαλύτερο πλήθος καταστάσεων έχει. Αυτό το αποτέλεσμα αναδεικνύει την πιθανότητα της συνύπαρξης πολλών καταστάσεων με τοπολογίες παράλληλου προγραμματισμού που αντιστοιχούν σε υψηλά επίπεδα της ταξινομίας. Φαίνεται, πως στους κώδικες των μαθητών με μεγάλο πλήθος καταστάσεων υπάρχει και μεγαλύτερη ανάγκη για παραλληλία μεταξύ των σεναρίων. Αυτή η ανάγκη γίνεται αντιληπτή από τους μαθητές και πραγματοποιείται χρήση ανάλογων τοπολογιών παράλληλου προγραμματισμού.



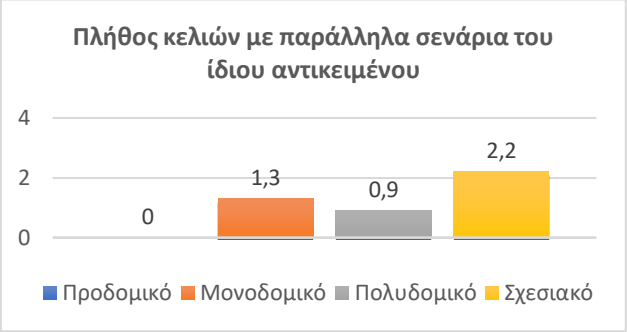
Σχήμα 10. Μέσος όρος για το πλήθος των καταστάσεων των κωδικΟραμάτων κάθε επιπέδου της ταξινομίας SOLO

Το σχήμα 11 δείχνει τον μέσο όρο του πλήθους των διαδικασιών των κωδικΟραμάτων για κάθε επίπεδο της ταξινομίας SOLO. Τα αποτελέσματα αναδεικνύουν πως στο προδομικό επίπεδο γίνεται χρήση διαδικασιών σε μεγάλο βαθμό σε σχέση με τα άλλα επίπεδα της ταξινομίας. Αυτό το γεγονός φανερώνει πως εκεί που δεν γίνεται χρήση τοπολογιών παράλληλου προγραμματισμού, επικρατούν στοιχεία συναρτησιακού προγραμματισμού, όπως είναι η ύπαρξη και η χρήση πολλαπλών διαδικασιών. Στις υπόλοιπες κατηγορίες (μονοδομικό, πολυδομικό, σχεσιακό) της ταξινομίας φαίνεται να υπάρχει αύξηση του πλήθους των διαδικασιών, όσο αυξάνεται το επίπεδο ένταξης του κώδικα. Κάτι τέτοιο ίσως συμβαίνει, καθώς στα μεγαλύτερα επίπεδα η πολυπλοκότητα αυξάνεται και η χρήση διαδικασιών εξυπηρετεί μια τέτοια ανάγκη.



Σχήμα 11. Μέσος όρος για το πλήθος των διαδικασιών των κωδικΟραμάτων κάθε επιπέδου της ταξινομίας SOLO

Τέλος, το σχήμα 12 αφορά τον μέσο όρο του πλήθους των κελιών με παράλληλα σενάρια που αφορούν το ίδιο αντικείμενο, για κάθε επίπεδο της ταξινομίας SOLO. Από το αποτέλεσμα αυτό δεν προκύπτει κάποια παρατήρηση καθώς οι διαφορές μεταξύ των επιπέδων είναι πολύ μικρές.



Σχήμα 12. Μέσος όρος για το πλήθος κελιών με παράλληλα σενάρια του ίδιου αντικειμένου των κωδικΟραμάτων κάθε επιπέδου της ταξινομίας SOLO

Συμπεράσματα

Το βασικό συμπέρασμα που εξάγεται από την ερμηνεία των αποτελεσμάτων είναι πως οι μαθητές μπορούν ακόμη και από νεαρή ηλικία να σκεφτούν παράλληλα και να

χρησιμοποιήσουν με επιτυχία στον κώδικα τους τον παράλληλο προγραμματισμό. Το αποτέλεσμα αυτό συμφωνεί με τον Kahn (1996), ο οποίος τονίζει ότι ο παράλληλος προγραμματισμός είναι πιο κοντά στο φυσικό τρόπο σκέψης των νεαρών μαθητών, καθώς στον πραγματικό κόσμο πολλαπλές οντότητες δρουν ταυτόχρονα και επικοινωνούν μεταξύ τους. Τα αποτελέσματα της μελέτης δείχνουν, πως οι μαθητές δέκα έως δεκατριών ετών χρησιμοποιούν τοπολογίες παράλληλου προγραμματισμού όταν τους δίνεται αυτή η δυνατότητα, όπως δηλαδή συμβαίνει στο περιβάλλον οπτικού προγραμματισμού Scratch. Η βιβλιογραφική ανασκόπηση συμφωνεί με αυτό, καθώς οι μαθητές ηλικιών δημοτικού σε ποικίλες μελέτες αναδείχθηκε πως μετά από τη διαδικασία διδασκαλίας άρχισαν να σκέφτονται παράλληλα και να χρησιμοποιούν παράλληλο κώδικα με επιτυχία (Gregg et al., 2012; Sáez-López et al., 2016; Fatourou et al., 2018; Weintrop et al., 2018). Σύμφωνα με τους Weintrop et al. (2018), οι νεαροί μαθητές πολλές φορές παράγουν παράλληλα τμήματα κώδικα χωρίς να το καταλαβαίνουν. Αυτό επιβεβαιώνεται και από την παρούσα έρευνα, καθώς το επίπεδο διδασκαλίας πάνω στον παράλληλο προγραμματισμό που προηγήθηκε της ανάπτυξης κώδικα από τους μαθητές, είναι μικρό έως ανύπαρκτο. Επίσης, γίνεται αντιληπτό ότι αρκετές φορές προτιμούν να χρησιμοποιούν το «Όταν» ή στοιχεία που παραπέμπουν σε αυτό και όχι το «Για πάντα… Αν»". Κάτι τέτοιο έρχεται σε συμφωνία με τους Tarkan et al, οι οποίοι το 2010 συμπέραναν, πως οι μαθητές προτιμούν τις παράλληλες διεργασίες σε σχέση με τη δομή επανάληψης. Πιο συγκεκριμένα, προτιμούν πολλαπλές οντότητες να λειτουργούν παράλληλα εκτελώντας διεργασίες, παρά μία οντότητα να εκτελεί όλες τις διεργασίες μέσω της δομής επανάληψης. Τέλος, όσον αφορά την πλατφόρμα οπτικού προγραμματισμού Scratch, με βάση τη μεγάλη χρήση παράλληλου προγραμματισμού από τους μαθητές, που παρατηρήθηκε στην παρούσα εργασία, αναδεικνύεται, πως προσφέρεται για την διδασκαλία παράλληλου προγραμματισμού σε αρχάριους. Το συμπερασμα αυτό έρχεται σε συμφωνία με τους Λαδιά και Καρβουνίδη, που το 2021 τονίζουν αυτή τη χρησιμότητα του Scratch.

Συνοψίζοντας, οι εκπαιδευόμενοι ηλικίας δέκα έως δεκατριών ετών έχουν τις κατάλληλες διεργασίες στην υπολογιστική τους σκέψη, ώστε ο παράλληλος προγραμματισμός να γίνει μέρος του αναλυτικού προγράμματος στις τελευταίες τάξεις του δημοτικού.

Βιβλιογραφικές Αναφορές

Aho, A. V. (2012). Computation and Computational Thinking. *The Computer Journal, 55*(7), 832–835.

Bal, H. E., Steiner, J. G., & Tanenbaum, A. S. (1989) Programming Languages for Distributed Computing Systems. *ACM Computing Surveys, 21*(3), 261-322.

Biggs, J. B., & Collis, K. F. (1982). *Evaluating the quality of learning. The SOLO taxonomy*. NY: Academic Press.

Fatourou, E., Zygouris, N. C., Loukopoulos, T., & Stamoulis, G. I. (2018). Teaching concurrent programming concepts using scratch in primary school: Methodology and evaluation. *International Journal of Engineering Pedagogy, 8*(4), 543–552.

Feldhausen, R., Bell, S., Andresen, D. (2014). Minimum Time, Maximum Effect: Introducing Parallel Computing in CS0 and STEM Outreach Activities Using Scratch. *Conference: Proceedings of the Extreme Science and Engineering Discovery Environment* (pp. 1-7). Atlanta, GA.

Gregg, C., Tychonievich, L., Cohoon, J., & Hazelwood, K. (2012). EcoSim: A Language and Experience Teaching Parallel Programming in Elementary School. *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education (SIGCSE '12)* (pp. 51-56). NC: ACM.

Kahn, K. (1996). Drawing on napkins, video-game animation, and other ways to program computers. *Communications of the ACM, 39*(8), 49-59.

Libert, C., Vanhoof, W. (2019). Analysis of students' preconceptions of concurrency. *EASEAI 2019: Proceedings of the 1st ACM SIGSOFT International Workshop on Education through Advanced Software Engineering and Artificial Intelligence* (pp. 9-12). Tallin: ACM.

Pellas, N. (2014). The development of a virtual learning platform for teaching concurrent programming languages in Secondary education: The use of Open Sim and Scratch4OS. *Journal of E-Learning and Knowledge Society 10*(1).

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.

Sáez-López, J.-M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two-year case study using "Scratch" in five schools. *Computers & Education, 97*, 129–141.

Samuel, M. (2017). An insight to Programming Paradigms and Their Programming Languages. *Journal of Applied Technology and Innovation, 1*(1), 37-57.

Tarkan, S., Sazawal, V., Druin, A., Golub, E., Bonsignore, E. M., Walsh, G., & Atrash, Z. (2010). Toque: Designing a Cooking-Based Programming Language for And with Children. *Proceedings of the 28th International Conference on Human Factors in Computing Systems - CHI '10* (pp. 2417–2426). Atlanta, USA: ACM.

Weintrop, D., Hansen, A. K., Harlow, D. B., & Franklin, D. (2018). Starting from Scratch: Outcomes of Early Computer Science Learning Experiences and Implications for What Comes Next. *Proceedings of the 2018 ACM Conference on International Computing Education Research - ICER '18* (pp 142-150). Espoo, Finland: ACM.

Wilkinson, B. & Allen, M (2005). *Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers*. 2nd Edition, Pearson Prentice Hall.

Δημακόπουλος Β., (2015). *Παράλληλα Συστήματα και Προγραμματισμός*. [ηλεκτρ. βιβλ.] Αθήνα: Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. Ανακτήθηκε Ιανουάριο 15,2021, από : http://hdl.handle.net/11419/3209

Λαδιάς, Δ., Καρβουνίδης, Θ., Λαδιάς, Α. (2020). Αξιοποίηση τεχνικών παράλληλου προγραμματισμού στο περιβάλλον Scratch. *12th CIE2020 - Η Πληροφορική στην Εκπαίδευση* (σ. 2-13). Πειραιάς.

Λαδιάς, Δ. & Καρβουνίδης, Θ. (2021). Τοπολογίες παράλληλου προγραμματισμού στο Scratch και κατηγοριοποίησή τους με χρήση της ταξινομίας SOLO. *13th CIE2020 - Η Πληροφορική στην Εκπαίδευση* (σ. 484-499). Κέρκυρα.

Λαδιάς, Α. & Λαδιάς Δ. (2016). Αναπαράσταση αλγορίθμων με τη βοήθεια του ΚωδικΟράματος σε περιβάλλοντα οπτικού προγραμματισμού. *Θέματα Επιστημών και Τεχνολογίας στην Εκπαίδευση, 9*(2), 103-117.