



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ

ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

**«Η Διδασκαλία του Αλγοριθμικού Προγραμματισμού
στη Δευτεροβάθμια Εκπαίδευση με Χρήση Ήχου»**



ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Σκληρού Θεοφανής του Στυλιανού

Τρίπολη, Ιούνιος 2021



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ

ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

**Η Διδασκαλία του Αλγοριθμικού Προγραμματισμού
στη Δευτεροβάθμια Εκπαίδευση με Χρήση Ήχου**

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Σκληρού Θεοφανής του Στυλιανού

Τριμελής Συμβουλευτική Επιτροπή : Τσελίκας Νικόλαος (Επιβλέπων)

Γεωργάκη Αναστασία

Πατρικάκης Χαράλαμπος

Υποστηρίχθηκε την

10^η Ιουνίου 2021

ενώπιον της **Επταμελούς Εξεταστικής Επιτροπής**

Η Επταμελής Εξεταστική Επιτροπή

Νικόλαος Τσελίκας
Αναπληρωτής Καθηγητής
Πανεπιστήμιο Πελοποννήσου

Digitally signed by
NIKOLAOS
TSELIKAS
Date: 2021.06.14
20:15:20 +03'00'

Αναστασία Γεωργάκη
Καθηγήτρια
Εθνικό και Καποδιστριακό
Πανεπιστήμιο Αθηνών

Χαράλαμπος Πατρικάκης
Καθηγητής
Πανεπιστήμιο Δυτικής Αττικής

Αρετή Ανδρεοπούλου
Επίκουρη Καθηγήτρια
Εθνικό και Καποδιστριακό
Πανεπιστήμιο Αθηνών

Digitally signed
by Charalampos
Patrikakis
Date: 2021.06.14
15:18:18 +03'00'

Ιωάννης Μοσχολιός
Αναπληρωτής Καθηγητής
Πανεπιστήμιο Πελοποννήσου

Χριστίνα Αναγνωστοπούλου
Αναπληρώτρια Καθηγήτρια
Εθνικό και Καποδιστριακό
Πανεπιστήμιο Αθηνών

Ιωάννης Ζάννος
Καθηγητής
Ιόνιο Πανεπιστήμιο

IOANNIS
MOSCHO
LIOS

Digitally signed
by IOANNIS
MOSCHOLIOS
Date: 2021.06.14
19:33:03 +03'00'

Christina Exakousti Anagnostopoulou

Christina Exakousti Anagnostopoulou
14.06.2021 14:11

.....

Θεοφανή Σ. Σκληρού

Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Πανεπιστήμιο Πελοποννήσου

Copyright © Θεοφανή Σ. Σκληρού, 2021.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας εξ' ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να θεωρηθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Πελοποννήσου.

Αφιερωμένη
στον γιο μου & τον σύζυγό μου

Πρόλογος - Ευχαριστίες

Η συγγραφή της παρούσας διδακτορικής διατριβής είναι το αποτέλεσμα εντατικής έρευνας και εξέτασης πηγών, μελέτης πολλών και σημαντικών συγγραμμάτων και προϊόν της τετραετούς τουλάχιστον ερευνητικής μου δραστηριότητας στο Τμήμα Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Πελοποννήσου. Πρώτο έναυσμα της διατριβής αποτέλεσε το διετές μεταπτυχιακό που εκπόνησα στο ίδιο τμήμα, θέλοντας αφενός να δώσω το παράδειγμα του «διδάσκω αεί διδασκόμενος» στους μαθητές μου και αφετέρου, μετά από είκοσι πέντε (25) και πλέον χρόνια διδασκαλίας του αλγοριθμικού προγραμματισμού και καθώς διαπιστώνω τη φθίνουσα πορεία κατανόησής του από το μαθητικό δυναμικό της δευτεροβάθμιας εκπαίδευσης, να βρω έναν τρόπο για την καλύτερη και ευκολότερη εκμάθησή του. Η εμπειρία στην εκπαίδευση, με διαφορετικές γενιές μαθητών και με τα διαφορετικά ερεθίσματα του καθενός, καθώς και το γεγονός ότι είμαι εν ενεργεία εκπαιδευτικός, λειτούργησαν ως βάση για την ολοκλήρωση της διατριβής μου.

Η γράφουσα έχει τη συναίσθηση ότι η παρούσα μελέτη ανταποκρίνεται στη σοβαρότητα και επικαιρότητα του θέματος. Είναι μια μελέτη ανοικτή σε κάθε είδους κριτική και διόρθωση. Δεν διεκδικεί την αυθεντία, αλλά φιλοδοξεί να αποτελέσει την αφορμή για την ενασχόληση εκπαιδευτικών με την αποτελεσματική διδακτική της Πληροφορικής.

Κύριος στόχος της διατριβής είναι η διδασκαλία του αλγοριθμικού προγραμματισμού στη Δευτεροβάθμια εκπαίδευση να γίνει κατανοητή στους μαθητές, με τη χρήση ηχητικών μηνυμάτων ως συμπληρωματικό εργαλείο στην εκμάθηση, προσομοιώνοντας αντίστοιχο ηχητικό μήνυμα με κάποια ενέργεια ή εντολή.

Αναγνωρίζοντας ότι η διατριβή μου θα ήταν αδύνατον να πραγματοποιηθεί χωρίς την καθοδήγηση και την αμέριστη συμπαράσταση των καθηγητών μου, θα ήθελα στο σημείο αυτό να ευχαριστήσω όλους όσους με βοήθησαν και με στήριξαν σ' αυτή μου την πορεία στο Πανεπιστήμιο Πελοποννήσου, στο ΕΚΠΑ και στο ΠΑΔΑ. Ξεκινώντας από τη συμβουλευτική μου επιτροπή, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, Αναπληρωτή Καθηγητή κύριο Νικόλαο Τσελίκα, ο οποίος με καθοδηγούσε και μου συμπαραστεκόταν σε κάθε βήμα του ερευνητικού μου έργου. Προσπάθησα να φανώ αντάξια της εμπιστοσύνης του και των προσδοκιών του. Με βαθύ επιστημονικό και τεχνικό υπόβαθρο αποτέλεσε σημαντικό αρωγό μου για την ολοκλήρωση της διατριβής. Συνεχίζοντας θέλω να ευχαριστήσω την Καθηγήτρια και Πρόεδρο του τμήματος Μουσικών Σπουδών του ΕΚΠΑ, κυρία Αναστασία

Γεωργάκη, η οποία αγκάλιασε την προσπάθειά μου με σεβασμό και εκτίμηση από την πρώτη στιγμή. Στάθηκε δίπλα μου άριστη συμβουλάτορας, καταπληκτική παιδαγωγός, με εξαιρετική διεπιστημονική κατάρτιση. Επίσης, τον Καθηγητή κύριο Χαράλαμπο Πατρικάκη, ο οποίος ήταν πάντοτε πρόθυμος και συνεπικουρούσε στο εγχείρημά μου. Επιπροσθέτως θα ήθελα να ευχαριστήσω την Επίκουρη Καθηγήτρια κυρία Αρετή Ανδρεοπούλου, η οποία πρόθυμα και ακούραστα με δίδαξε το προγραμματιστικό περιβάλλον MAX/MSP στο ΕΚΠΑ, ενώ με την άριστη μεταδοτικότητά της, με βοήθησε στην υλοποίηση των διδακτικών σεναρίων και μου έκανε την τιμή να συμπεριληφθεί στην επταμελή μου εξεταστική επιτροπή.

Θέλω να ευχαριστήσω επίσης τον Αναπληρωτή Καθηγητή κύριο Ιωάννη Μοσχολιό μέλος της εξεταστικής επιτροπής για την εκτίμηση και τη βοήθειά του, όσο ήμουν φοιτήτρια στο μεταπτυχιακό πρόγραμμα, καθώς και τον Καθηγητή κύριο Ιωάννη Ζάννο και την Αναπληρώτρια Καθηγήτρια κυρία Χριστίνα Αναγνωστοπούλου που μου έκαναν την τιμή να συμπεριληφθούν στην επταμελή μου εξεταστική επιτροπή.

Ευχαριστώ όλες τις μαθήτριες και όλους τους μαθητές Λυκείου που συμμετείχαν στα διδακτικά σενάρια και στις μελέτες που παρουσιάζονται στην παρούσα διατριβή.

Τέλος ευχαριστώ την οικογένειά μου, τον σύζυγό μου Άγη και τον γιο μου Γιώργο για την αγάπη που απλόχερα μου χάριζαν και μου χαρίζουν καθημερινά, καθώς και την πίστη και το θαυμασμό που μου έδειξαν καθ' όλη τη διάρκεια αυτών των συγκεκριμένων σπουδών μου (μεταπτυχιακών και διδακτορικών). Η πραγματοποίηση της διατριβής, αλλά και η γενικότερη έως τώρα επιστημονική και δημιουργική μου πορεία μου, οφείλεται κατά ένα πολύ σημαντικό μέρος σ' εκείνους.

Θεοφανή Σ. Σκληρού

Περίληψη

Στη Δευτεροβάθμια Εκπαίδευση, ενώ επιτυγχάνουμε τους στόχους του προγράμματος σπουδών, θεωρείται δύσκολο να διδάξουμε τον προγραμματισμό (αρκετές μελέτες παρουσιάζονται σ' αυτήν την έρευνα). Ωστόσο, όταν η διδασκαλία υποστηρίζεται από κατάλληλες μεθοδολογίες, η μαθησιακή διαδικασία μπορεί να βελτιωθεί. Στο πλαίσιο αυτό, σε πρώτη φάση, στο Κεφάλαιο 1, που αποτελεί το εισαγωγικό κεφάλαιο, περιγράφεται ο ρόλος και ο σκοπός του αλγοριθμικού προγραμματισμού στη Δευτεροβάθμια Εκπαίδευση, καθώς και τα ερευνητικά ερωτήματα, ο σκοπός και η διάρθρωση της διατριβής. Στο Κεφάλαιο 2 γίνεται μία συνοπτική αναδρομή στη σημερινή κατάσταση των εκπαιδευτικών εργαλείων. Η παρούσα διατριβή ασχολείται με τις διαφορετικές προσεγγίσεις διδασκαλίας του προγραμματισμού στη Δευτεροβάθμια Εκπαίδευση και εξετάζει στα Κεφάλαια 3, 4 και 5 το πιθανό όφελος των ηχητικών σημάτων ως συμπληρωματικό εργαλείο διδασκαλίας. Βασίζεται σε μεθοδολογία συμμετοχικής παρατήρησης, πειραματισμού, ποσοτικής, ποιοτικής ανάλυσης και αξιολόγησης.

Επιπλέον, παρουσιάζονται η περιγραφή, η ανάλυση, η αξιολόγηση των προτεινόμενων διδακτικών σεναρίων και στο Κεφάλαιο 6 τα συμπεράσματα των μαθητών. Δημιουργήθηκαν ηχητικές ειδοποιήσεις με αντιστοίχιση διαφορετικών ερεθισμάτων ήχου σε συγκεκριμένες ενέργειες και λειτουργίες προγραμματισμού, όπως η επιτυχής ανάθεση του σωστού τύπου δεδομένων σε μια μεταβλητή. Τόσο η επιλογή των ηχητικών ερεθισμάτων, όσο και η χρήση των ηχητικών σημάτων στον προγραμματισμό, αξιολογήθηκαν μέσω υποκειμενικών μελετών.

Τα αποτελέσματα έδειξαν ότι οι συμμετέχοντες προτιμούν τους τεχνητούς ήχους σε σχέση με τους φυσικούς (προ-καταγεγραμμένους) για όλους τους τύπους προειδοποιήσεων. Διαπιστώθηκε, επίσης, ότι οι χρήστες προτιμούν τις προειδοποιήσεις ήχου που σχετίζονται με ενέργειες που πρόκειται να γίνουν, τα σφάλματα, την επιτυχή εκτέλεση κώδικα, έλεγχο συνθηκών και τις επαναληπτικές διαδικασίες, σηματοδοτώντας την εκτέλεση βήμα προς βήμα του κώδικα. Τα αποτελέσματα των δοκιμών έδειξαν ότι οι μαθητές κατανοούν καλύτερα τις έννοιες και τη χρήση εντολών κώδικα όταν χρησιμοποιούν ένα περιβάλλον προγραμματισμού εμπλουτισμένο με ήχο, από ένα συμβατικό περιβάλλον.

Η διατριβή ολοκληρώνεται με την παρουσίαση της πιθανής μελλοντικής εξέλιξης και αξιοποίησης του προτεινόμενου μοντέλου στα μελλοντικά προγραμματιστικά περιβάλλοντα διδασκαλίας στο Κεφάλαιο 7, καθώς και με την αξιολόγηση των σημαντικότερων αποτελεσμάτων της προσέγγισης. Αυτά τα αποτελέσματα είναι το κίνητρο για τη δημιουργία μιας βάσης δεδομένων ηχητικών και φωνητικών μηνυμάτων και ενός νέου ολοκληρωμένου εκπαιδευτικού εργαλείου που χρησιμοποιεί τον ήχο, που θα αποτελέσει ένα νέο βοήθημα για τους εκπαιδευτικούς ώστε να μπορέσουν να διδάξουν αποτελεσματικότερα τον αλγοριθμικό προγραμματισμό στους μαθητές τους, να ενισχύσουν-βελτιώσουν τη διάδραση στην τάξη και να χρησιμεύσει στην περαιτέρω έρευνα για τα οφέλη του ήχου και την ψυχοακουστική αντίληψη των εφήβων.

Λέξεις κλειδιά: Αλγοριθμικός προγραμματισμός, Προσομοίωση, Ακουστικές ειδοποιήσεις, Υπολογιστική Σκέψη, Ψυχοακουστική αντίληψη

Summary

It is considered hard to teach programming in secondary education while achieving the aims of curriculum (several studies are presented in this research). However, when teaching is supported by suitable methodologies, learning can be ameliorated. Under this premise, the first part, chapter 1, which is the introductory chapter, describes the role and purpose of algorithmic programming in Secondary Education, as well as the research questions, the purpose and the structure of the dissertation. Chapter 2 includes a concise retrospection of the current situation of teaching tools. This thesis discusses different teaching approaches to programming in secondary (high-school) education and examines the potential benefit of sound-alerts as a complementary teaching tool in Chapters 3, 4 and 5. It is based on a methodology of participatory observation, experimentation, quantitative, qualitative analysis and evaluation.

Furthermore a description, analysis, evaluation of proposed teaching scenarios and the students' conclusions in chapter 6, take place. Such alerts were created by pairing different sound stimuli to specific programming actions and operations, such as the successful assignment of the correct data type to a variable. Both the selection of the sound stimuli, as well as the potential impact of the use of sound alerts on programming where evaluated through subjective studies.

Results showed that participants preferred synthesized to natural (pre-recorded) stimuli for all types of alerts. It was also revealed that users prefer sound-alerts associated to pending actions, errors, successful code execution, conditional statements and looping over alerts highlighting the step-by-step execution of the code. The test results showed that students understand both the meaning and the use of code commands more clearly when they use a sound-enriched programming environment instead of a conventional one.

The thesis concludes with the presentation of the potential future evolution and exploitation of the proposed model in the Next Generation programming environments in Chapter 7, as well as with the evaluation of the more important results of the approach. These results are the motive for the creation of an audio and voice messages data base and a new comprehensive educational tool using sound, which will be a new teachers' aid to be able to teach algorithmic programming more effectively to their

students, enhance classroom interaction and serve to further research into the benefits of sound and psychoacoustic perception in adolescents.

Keywords: Algorithmic programming, Simulation, Sound alert, Computational thinking, Psychoacoustic perception

Λίστα δημοσιεύσεων – Συνέδρια

1. Θεοφανή Σκληρού, “Ο Προγραμματισμός στη Δευτεροβάθμια εκπαίδευση”, 4^ο Πανελλήνιο Συνέδριο Νέος Παιδαγωγός, σελ. 3523-3528, 1-2 Απριλίου 2017, Αθήνα.
2. T. S. Sklirou, "Programming in secondary education: Applications, new trends and challenges," *2017 IEEE Global Engineering Education Conference (IEEEEDUCON)*, Athens, Greece, 2017, pp. 580-585, doi: 10.1109/EDUCON.2017.7942904.
3. Theofani Sklirou, Areti Andreopoulou, & Anastasia Georgaki. (2018). Teaching Programming in Secondary Education through Sound. Presented at the 15th Sound and Music Computing Conference (SMC2018), Limassol, Cyprus: Zenodo. <http://doi.org/10.5281/zenodo.1422526>
4. Sklirou, T.S., Andreopoulou, A., Georgaki, A. and Tselikas, N.D. 2020. Introducing Secondary Education Students to Programming through Sound Alerts. *European Journal of Engineering and Technology Research*. 5, 12 (Dec. 2020), 130-139. DOI: <https://doi.org/10.24018/ejers.2020.5.12.2298>

Περιεχόμενα

<i>Πρόλογος - Ευχαριστίες</i>	7
<i>Περίληψη</i>	9
<i>Summary</i>	11
<i>Λίστα δημοσιεύσεων – Συνέδρια</i>	13
<i>Κατάλογος Σχημάτων</i>	17
<i>Κατάλογος Πινάκων</i>	19
1. Εισαγωγικό κεφάλαιο	20
1.1 Εισαγωγή.....	20
1.2 Σκοπός της διατριβής.....	24
1.3 Ερευνητικά ερωτήματα	25
1.4 Διάρθρωση της διατριβής.....	25
2. Περιβάλλοντα Προγραμματισμού στη Δευτεροβάθμια Εκπαίδευση	27
2.1 Οι απαρχές.....	27
2.2 Σύγχρονες διδακτικές προσεγγίσεις.....	29
2.3 Εκπαιδευτικά εργαλεία που χρησιμοποιούνται σήμερα.....	31
2.4 Δυσκολίες στην κατανόηση του αλγοριθμικού προγραμματισμού και της υπολογιστικής σκέψης	34
2.4.1 Χρήση μεταβλητών	38
2.4.2 Είσοδος δεδομένων	39
2.4.3 Δομές επιλογής.....	39
2.4.4 Δομές επανάληψης	40
2.4.5 Ταξινόμηση-αναζήτηση	40
2.4.6 Υποπρογράμματα	41
2.5 Περιβάλλοντα προγραμματισμού που χρησιμοποιούν ήχους.....	42
3. Πρόταση μεθοδολογίας διδασκαλίας του προγραμματισμού χρησιμοποιώντας ηχητικές ειδοποιήσεις	50
3.1 Χρήση λογισμικών στην παρούσα διατριβή	51
3.2 Επιλογή ήχων.....	51
3.3 Πρωτόκολλο πειραμάτων α΄ φάσης (καταγραφή της διαδικασίας επιλογής ήχων – φυσικός ή τεχνητός ήχος).....	52
3.4 Αποτελέσματα της προκαταρκτικής μελέτης	53
4. Πειράματα β΄ φάσης- Μελέτη αξιολόγησης	55
4.1 Πρωτόκολλο πειραμάτων β΄ φάσης	55
4.2 Σενάρια διδασκαλίας.....	56
4.2.1 Σενάριο 1: Εισαγωγή δεδομένων.....	56

4.2.2 Σενάριο 2: Αριθμητική πράξη	60
4.2.3 Σενάριο 3: Σχεσιακή πράξη (πράξη σύγκρισης- εκτέλεση λογικής πράξης)	61
4.2.4 Σενάριο 4: Δομή επιλογής (έλεγχος συνθηκών)	63
4.2.5 Σενάριο 5: δομή επανάληψης - "όσο επανάλαβε (whiledo)"	63
4.2.6 Σενάριο 6: δομή επανάληψης - "Αρχή επανάληψης μέχρις ότου (repeat...until)"	64
4.3 Αποτελέσματα του δεύτερου πειράματος	65
(Στατιστικά –Βαθμός επιρροής ήχου)	65
5. Τρίτο πείραμα - Μελέτη αξιολόγησης	69
5.1 Πρωτόκολλο τρίτου πειράματος	69
5.2 Δοκιμή	69
5.3 Αποτελέσματα δοκιμών.....	69
6. Αξιολόγηση και συμπεράσματα μαθητών	72
7. Συμπεράσματα & νέα πεδία διερεύνησης.....	75
7.1 Συμπεράσματα έρευνας.....	75
7.2 Προοπτικές – Μελλοντικά πλάνα.....	77
7.2.1 Εμφωλευμένες επαναληπτικές δομές	77
7.2.2 Δομές δεδομένων (Πίνακες, Δυναμικές δομές).....	78
7.2.3 Αναζήτηση.....	78
7.2.4 Ταξινόμηση	78
7.2.5 Κλήση υποπρογραμμάτων.....	78
7.2.6 Κατηγορίες λαθών	79
7.2.7 Διαγράμματα ροής.....	79
7.2.8 Σενάρια διδασκαλίας με μουσικά δεδομένα.....	79
7.2.9 Ολοκληρωμένο προγραμματιστικό περιβάλλον με ήχο	79
7.2.10 Τελικό συμπέρασμα	80
8. Πίνακας ορολογίας.....	82
9. Συντμήσεις – Αρκτικόλεξα – Ακρωνόμια	84
10. Παραρτήματα	85
10.1 Ερωτηματολόγιο - Επιλογή ήχου	85
10.2 Ερωτηματολόγιο – Κατανόηση εντολών	86
10.3 Ερωτηματολόγιο – Κατανόηση δομών επανάληψης.....	87
10.4 Ερωτηματολόγιο – Αξιολόγηση της διαδικασίας των πειραμάτων της διατριβής από τους μαθητές.....	88
10.5 Ερωτηματολόγιο – Συμμετοχή στη δημιουργία ηχητικών ειδοποιήσεων.....	89
10.6 Ερωτηματολόγιο – Διαδικασίες & ηχητικές ειδοποιήσεις	90
11. Βιβλιογραφία	91

Περιεχόμενα

11.1 Ελληνόγλωσση Βιβλιογραφία.....	91
11.2 Ξενόγλωσση Βιβλιογραφία	93
11.3 Πηγές από το Διαδίκτυο.....	97

Κατάλογος Σχημάτων

Σχήμα 1. Το περιβάλλον προγραμματισμού της Γλωσσομάθειας (ψευδοκώδικας).....	32
Σχήμα 2. Το περιβάλλον προγραμματισμού Max / MSP.....	46
Σχήμα 3. Επισκόπηση αποτελεσμάτων των προκαταρκτικών δοκιμών.....	53
Σχήμα 4. Περιβάλλον Max / MSP - Εκτέλεση προγράμματος με ήχο (2ο πείραμα).....	54
Σχήμα 5α. Καταχώριση δεδομένων	56
Σχήμα 5β. Κώδικας Max/MSP για τον ήχο αναμονής.....	57
Σχήμα 5γ. Κώδικας Max/MSP για τον ήχο επιτυχίας.....	58
Σχήμα 5δ. Κώδικας Max/MSP για τον ήχο αποτυχίας.....	58
Σχήμα 5ε. Κώδικας Max/MSP ενεργοποίησης εκάστοτε ηχητικής ειδοποίησης.....	59
Σχήμα 6α. Αριθμητική πράξη.....	59
Σχήμα 6β. Κώδικας Max/MSP για ήχο που αντιστοιχεί με αριθμητική πράξη.....	60
Σχήμα 7α. Πράξη σύγκρισης.....	61
Σχήμα 7β. Κώδικας Max/MSP για ήχο που αντιστοιχεί με λογική πράξη.....	61
Σχήμα 8. Δομή επιλογής (έλεγχος συνθηκών).....	62
Σχήμα 9. Δομή επανάληψης - "Όσο επανάλαβε (while ...do)".....	63
Σχήμα 10. Δομή επανάληψης - "Αρχή επανάληψης μέχρις ότου (repeat...until)".....	64
Σχήμα 11. Συνολική αξιολόγηση της χρήσης των ηχητικών προειδοποιήσεων σχετικά με την κατανόηση του κώδικα, κατά μέσον όρο στα τρία πρώτα σενάρια διδασκαλίας.....	66
Σχήμα 12. Συνολική αξιολόγηση της χρήσης των ηχητικών προειδοποιήσεων σχετικά με την κατανόηση του κώδικα, κατά μέσον όρο στα τρία τελευταία σενάρια διδασκαλίας.....	67
Σχήμα 13. Αξιολόγηση των μαθητών συμμετεχόντων και μη, στα διδακτικά σενάρια της έρευνας (Ποσοστά βαθμολογιών)	70

Σχήμα 14. Αξιολόγηση και συμπεράσματα των συμμετεχόντων μαθητών στα διδακτικά σενάρια της έρευνας.....	71
Σχήμα 15. Συμμετοχή των μαθητών /τριών στη δημιουργία ηχητικών ειδοποιήσεων.....	72
Σχήμα 16. Αξιολόγηση των ηχητικών ειδοποιήσεων ανά διαδικασία.....	73

Κατάλογος Πινάκων

Πίνακας 1. Πίνακας εκπαιδευτικών περιβαλλόντων προγραμματισμού που χρησιμοποιούνται σήμερα.....	30
Πίνακας 2. Πίνακας εκπαιδευτικών περιβαλλόντων προγραμματισμού στα οποία χρησιμοποιείται Ήχος	41
Πίνακας 3. Στατιστικά στοιχεία των τριών πρώτων σεναρίων-εντολές κατανόησης με ήχους (αναλυτικά ανά διαδικασία).....	66
Πίνακας 4. Στατιστικά στοιχεία των τριών τελευταίων σεναρίων - εντολές κατανόησης με ήχους (αναλυτικά ανά διαδικασία).....	67
Πίνακας 5. Στατιστικά στοιχεία βαθμολογίας.....	69
Πίνακας 6. Αξιολόγηση της διαδικασίας των πειραμάτων της διατριβής από τους μαθητές.....	71
Πίνακας 7. Δημιουργία ηχητικών ειδοποιήσεων από τους ίδιους/ίδιες τους /τις μαθητές/μαθήτριες.....	72
Πίνακας 8. Διαδικασίες και ηχητικές ειδοποιήσεις.....	73

1. Εισαγωγικό κεφάλαιο

Στο εισαγωγικό κεφάλαιο, γίνεται παρουσίαση της έννοιας του προγραμματισμού και αναπτύσσεται η σημασία της διδασκαλίας του, ο ρόλος του και οι στόχοι που πρέπει να επιτευχθούν στη Δευτεροβάθμια Εκπαίδευση, ο σκοπός της διατριβής και τα ερευνητικά ερωτήματα που τέθηκαν. Τέλος, παρουσιάζεται συνοπτικά και η δομή των υπολοίπων κεφαλαίων, η διάρθρωση της διατριβής.

1.1. Εισαγωγή

Η εποχή μας χαρακτηρίζεται από τη ραγδαία εξέλιξη των επιστημών, της τεχνολογίας, καθώς και από τον ερευνητικό ανταγωνισμό. Η επιστημονική και η τεχνολογική πρόοδος επιδρούν καταλυτικά στις εργασιακές σχέσεις, στην οικονομική και κοινωνική ζωή, στην εκπαίδευση και στον πολιτισμό. Τα τελευταία είκοσι χρόνια οι Τεχνολογίες της Πληροφορίας και της Επικοινωνίας, λόγω του χαρακτήρα τους, έχουν αναπτυχθεί εντυπωσιακά, διαπερνούν σταδιακά το σύνολο της κοινωνίας και η παρουσία τους καθίσταται καταλυτική σε όλες τις ανθρώπινες δραστηριότητες (κοινωνικές, οικονομικές και πολιτιστικές). Η εισαγωγή στη εκμάθηση των νέων Τεχνολογιών της Πληροφορίας και της Επικοινωνίας είναι πλέον αναγκαία.

Παράλληλα, αναδεικνύεται μια γενιά παιδιών που είναι εξοικειωμένα με τα νέα τεχνολογικά εργαλεία και επιδεικνύουν εξαιρετική άνεση και σιγουριά, όταν χειρίζονται υπολογιστές και λογισμικά. Η σημασία του τεχνολογικού αλφαριθμητισμού των μαθητών της Δευτεροβάθμιας Εκπαίδευσης έχει διεθνώς αναγνωριστεί σε συνολικό επίπεδο, δηλαδή στο πρόγραμμα σπουδών, τη σχολική κοινότητα το κοινωνικό περιβάλλον. Στο σύγχρονο σχολείο οι ΤΠΕ δεν περιορίζονται απλά στον πληροφορικό αλφαριθμητισμό, αλλά παρέχουν δυναμικά εργαλεία και εφαρμογές υποστήριξης, ενίσχυσης και εμπλουτισμού της διδασκαλίας και της μάθησης, ώστε οι μαθητές να μετεξελιχθούν στο Πανεπιστήμιο και σε ερευνητικούς τομείς.

Η Πληροφορική στο Λύκειο δεν έχει σκοπό την επαγγελματική κατάρτιση των μαθητών στα επαγγέλματα της Πληροφορικής, αλλά τη συνέχιση και εμπάθυνση των γνώσεων που έχουν αποκτηθεί στις προηγούμενες βαθμίδες της εκπαίδευσης, καθώς και την προσαρμογή τους στις νέες εξελίξεις των ΤΠΕ.

Όπως επισημαίνουν οι Saeli, Perrenet, Jochems και Zwaneved (2011), καθώς και οι Selby και Woollard (2013), ο προγραμματισμός είναι η διαδικασία της γραφής, της

δοκιμής, του εντοπισμού σφαλμάτων / της αντιμετώπισης προβλημάτων και της διατήρησης του πηγαίου κώδικα των προγραμμάτων σε υπολογιστή. Η επιστήμη των ηλεκτρονικών υπολογιστών αποκτά όλο και μεγαλύτερη σημασία για τη δημιουργία νέων γνώσεων και πρέπει να διδάσκεται ως ξεχωριστό θέμα ή περιεχόμενο, ιδίως στα σχολεία δευτεροβάθμιας εκπαίδευσης (Fluck, Webb, Cox, Angeli, Malyn-Smith, Voogt, και Zagami, 2016).

Η διδασκαλία του προγραμματισμού είναι σχεδόν τόσο παλιά όσο η ίδια η τεχνολογία της πληροφορίας (Szlavi, P. & Zsako, L., 2003). Στα πρώτα βήματα (πριν από το 1950), σήμαινε την κατανόηση του τρόπου λειτουργίας των υπολογιστών και του πώς θα μπορούσαν να χρησιμοποιηθούν. Αμέσως στην επόμενη εποχή, ο προγραμματισμός ήταν μια «καλλιτεχνική» δραστηριότητα, οι προγραμματιστές έπρεπε να μάθουν διαφορετικά τεχνάσματα και επιπλέον έπρεπε να βρουν νέους αλγόριθμους για κάθε πρόβλημα. Η γνώση του προγραμματισμού διαδόθηκε από στόμα σε στόμα ακόμη και στη δεκαετία του 1950. Χρησιμοποίησαν γλώσσες προγραμματισμού που βασιζόνταν στο μηχάνημα, στο υλικό, επομένως αναγκάστηκαν να αφιερώσουν περισσότερο χρόνο για να επεξεργαστούν τις λεπτομέρειες παρά να επικεντρωθούν στην ουσία του προβλήματος. Εν τω μεταξύ, έπρεπε να συνειδητοποιήσουν ότι οι υπολογιστές ήταν αργοί και δεν είχαν αρκετή χωρητικότητα μνήμης.

Η διδασκαλία του προγραμματισμού έγινε πολύ σημαντική, μόλις εμφανίστηκαν οι γλώσσες προγραμματισμού υψηλού επιπέδου. Ωστόσο, λόγω της επιρροής της προηγούμενης περιόδου, βασίστηκε περισσότερο στη διδασκαλία μιας συγκεκριμένης γλώσσας προγραμματισμού. Οι ασκήσεις δίνονταν για να κατανοήσουν τις οδηγίες της γλώσσας προγραμματισμού και να εισαγάγουν τον τρόπο εφαρμογής της. Στην πραγματικότητα, η διδασκαλία του προγραμματισμού όπως την εννοούμε σήμερα, χρονολογείται από τη δεκαετία του 1970.

Στην Ελλάδα, λόγω της ανάπτυξης των Τ.Π.Ε., τα μαθήματα προγραμματισμού προστέθηκαν στο πρόγραμμα σπουδών της Δευτεροβάθμιας Εκπαίδευσης, όπως αναφέρεται στο ΦΕΚ 327/3-4-1998 τεύχος Β' και στη σελίδα του Π.Ι, αρχικά ως μαθήματα επιλογής στις τάξεις Α και Β του Λυκείου και κατεύθυνσης πανελλαδικά εξεταζόμενα στην τάξη Γ (Pi-schools.gr, 2021). Την τρέχουσα περίοδο αποτελούν υποχρεωτικά μαθήματα γενικής παιδείας στις Α και Β τάξεις, καθώς και μάθημα προσανατολισμού με 3πλάσιο αριθμό διδακτικών ωρών στην τάξη Γ, όπως αναφέρεται

στη σελίδα του Ι.Ε.Π.(Iep.edu.gr, 2021). Στο Λύκειο δεν έχουν επέλθει αλλαγές στο γνωστικό αντικείμενο του προγράμματος σπουδών της Πληροφορικής μετά την καθιέρωσή του το 1998, παρά μόνο στον αριθμό των ωρών διδασκαλίας.

Το σημαντικότερο έργο του εκπαιδευτικού είναι η διδασκαλία, αφού μέσα από αυτήν οι μαθητές κατακτούν τη γνώση. Πρέπει, λοιπόν, ο εκπαιδευτικός να σχεδιάσει εκείνη τη στρατηγική που θα ακολουθήσει καθώς και τις διδακτικές ενέργειες εκ των προτέρων, με βάση τις οποίες θα διεξαχθεί η διδασκαλία. Ο σχεδιασμός της διδασκαλίας γίνεται, αφού μελετηθεί το γνωστικό μέρος. Ο εκπαιδευτικός μπορεί ελεύθερα να επιλέξει εκείνη την πορεία διδασκαλίας, χρησιμοποιώντας κατάλληλες μεθόδους, έτσι ώστε η μάθηση να πραγματοποιείται κατά τον καλύτερο δυνατό τρόπο. Βέβαια με τον όρο μάθηση, δεν εννοούμε μόνο την απόκτηση γνώσεων και δεξιοτήτων, αλλά και τη βαθύτερη κατανόησή τους, καθώς και την ικανότητα να τις ανακαλεί, να τις αναπαράγει και να τις εφαρμόζει. Η επιλογή και ο συνδυασμός τρόπων και μεθόδων εξαρτάται από τη διδακτέα ύλη, τους σκοπούς – στόχους που πρέπει να επιτευχθούν, τα εποπτικά μέσα που είναι διαθέσιμα, καθώς και τις γνώσεις που έχουν αποκτήσει κατά το παρελθόν οι μαθητές.

Στη συνέχεια παρουσιάζεται η πορεία διδασκαλίας της θεματικής ενότητας «Ανάπτυξη εφαρμογών σε προγραμματιστικό περιβάλλον» της Γ΄ τάξεως Ενιαίου Λυκείου Τεχνολογικής Κατεύθυνσης, ή με την αλλαγή του προγράμματος σπουδών, του μαθήματος «Πληροφορική» της Γ΄ τάξεως Γενικού Λυκείου, του Προσανατολισμού Οικονομίας & Πληροφορικής.

Ο γενικός σκοπός του μαθήματος της Πληροφορικής τώρα στο Λύκειο, ιδιαίτερα στη Β και στη Γ τάξη, όπως αναφέρεται και στο πρόγραμμα σπουδών (Edu.klimaka.gr, 2021), είναι η ανάπτυξη δεξιοτήτων και ικανοτήτων σχετικών με την αλγοριθμική και την ορθολογική χρήση τους στην καθημερινή ζωή. Πολλές βασικές έννοιες της αλγοριθμικής και του προγραμματισμού συνιστούν αναπόσπαστο τμήμα των γενικών γνώσεων και δεξιοτήτων που πρέπει να αποκτήσει ο μαθητής. Πιο συγκεκριμένα σκοπός είναι οι μαθητές να καλλιεργήσουν αναλυτική σκέψη και συνθετική ικανότητα, να αναπτύξουν τη φαντασία, τον σχεδιασμό, τη δημιουργικότητα, να καλλιεργήσουν και να επικεντρωθούν στη σαφήνεια της έκφρασης και της διατύπωσης, να αναπτύξουν την ικανότητα της μεθοδικότητας και να γίνουν ικανοί να υλοποιούν πρωτίστως τις λύσεις απλών προβλημάτων, με τη χρήση βασικών προγραμματιστικών γνώσεων.

Έμφαση ιδιαίτερη δίνεται σε εκείνες τις ενότητες του μαθήματος που πραγματεύονται θέματα ανάλυσης προβλημάτων και σχεδίασης αλγορίθμου. Οι μαθητές θα πρέπει να κατανοούν πλήρως τα προβλήματα που τους ανατίθενται προς επίλυση, να έχουν τη δυνατότητα να διακρίνουν τα μέρη ενός προβλήματος, να είναι ικανοί να προσδιορίζουν και να αναφέρουν με σαφήνεια τα δεδομένα και τα ζητούμενα ενός προβλήματος (προσδιορισμός απαιτήσεων), να είναι σε θέση να διακρίνουν τη σημασία των λέξεων, σε μια φυσική γλώσσα και σε μια γλώσσα προγραμματισμού, κατανοήσουν ότι ο αλγόριθμος αποτελεί τον κρίκο ενοποίησης για όλες τις δραστηριότητες της επιστήμης των υπολογιστών, αντιληφθούν ότι ο αλγόριθμος είναι ανεξάρτητος της γλώσσας προγραμματισμού, στην οποία εκφράζεται και του υπολογιστικού συστήματος που τον εκτελεί, είναι ικανοί να εφαρμόζουν τους κανόνες για να σχεδιάσουν έναν αλγόριθμο και να μπορούν να έχουν άποψη για την ορθότητα του αλγορίθμου.

Η γενική μεθοδολογία του μαθήματος θα πρέπει να προωθεί, να ενισχύει και να ενθαρρύνει τη δημιουργική δράση του μαθητή μέσω της ενεργοποίησής του, τη συνεργατική μάθηση σε ομαδικό περιβάλλον, την ανάπτυξη δεξιοτήτων μεθοδολογικού χαρακτήρα, την ανάπτυξη κριτικής σκέψης μέσω της συζήτησης και του προβληματισμού και την αποτύπωση και κατάθεση ελεύθερης σκέψης και έκφρασης.

Στο εργαστηριακό μέρος του μαθήματος, χρησιμοποιώντας υπολογιστικά εργαλεία και τεχνικές, οι μαθητές εμπλέκονται σε ποικίλες δραστηριότητες οι οποίες διευκολύνουν την ανάπτυξη της δημιουργικής ικανότητας, υποστηρίζουν τον συμμετοχικό και συνεργατικό χαρακτήρα της μάθησης, καλλιεργούν την ανάπτυξη των δεξιοτήτων μοντελοποίησης και επίλυσης προβλημάτων και ευνοούν την αναλυτική και συνθετική σκέψη.

Έμφαση πρέπει να δίνεται στην ανάλυση και τον σχεδιασμό της λύσης των προβλημάτων και όχι σε αυτήν την ίδια τη λύση. Η βάση των εργασιών και των δραστηριοτήτων πρέπει να είναι η ανάλυση και ο σχεδιασμός. Οι δραστηριότητες για το σπίτι που ανατίθενται στους μαθητές θα πρέπει να επεξηγούνται στην τάξη. Οι μαθητές θα μπορούν να διατυπώνουν τις απορίες τους τόσο σε σχέση με τη διατύπωση της δραστηριότητας, όσο και σε σχέση με το περιεχόμενο και τα ζητούμενα. Ο διδάσκων θα πρέπει να παρέχει τις απαραίτητες διευκρινίσεις. Οι απαντήσεις των

μαθητών στις δραστηριότητες που τους ανατίθενται για το σπίτι, αποτελούν θέμα συζήτησης σε επίπεδο τάξης μετά την υλοποίησή τους. Θα πρέπει να αφιερώνεται χρόνος από τη διδακτική ώρα, έτσι ώστε να παρουσιάζονται οι περισσότερες δυνατές απαντήσεις των μαθητών, να σχολιάζονται και να κρίνονται από τους συμμαθητές τους και τον διδάσκοντα.

Η πρακτική αυτή έχει σαν αποτέλεσμα να ενισχύει την ανάπτυξη της κριτικής ικανότητας και της επιχειρηματολογίας των μαθητών, να καλλιεργεί την ιδέα του γόνιμου διαλόγου, να τροφοδοτεί νέες σκέψεις και να μπορεί να περάσει στον μαθητή την αίσθηση της επιβράβευσης της προσπάθειας του και να τον παρακινεί προς την κατεύθυνση της μεγιστοποίησης της.

Ο διδάσκων θα πρέπει να παρακινήσει τους μαθητές να ασκούν καλοπροαίρετη κριτική, να εμφυσήσει στους μαθητές το μήνυμα της συνεργατικότητας και να καλλιεργήσει το πνεύμα της ομαδικότητας που είναι πρωταρχικής σημασίας παιδαγωγικός στόχος.

Στην αίθουσα θα γίνεται, επίσης, η εισαγωγή και ανάπτυξη των διάφορων εννοιών παράλληλα με την πρακτική άσκηση στο εργαστήριο. Έτσι, η θεωρία και η πράξη θα βαδίζουν παράλληλα.

Τόσο όμως στο θεωρητικό όσο και στο πρακτικό μέρος, όπως θα παρουσιαστούν στη συνέχεια, προκύπτουν δυσκολίες στην κατανόηση εννοιών και ενεργειών ακόμη και με τα υπάρχοντα εργαλεία και μεθόδους, παρά την εξέλιξή τους.

1.2 Σκοπός της διατριβής

Σκοπός της παρούσας διατριβής είναι, συνεκτιμώντας και συνδυάζοντας τις προηγούμενες έρευνες και διδακτικές προσεγγίσεις, να αξιολογήσουμε κατά πόσο ο/η διδασκόμενος/η μαθητής/μαθήτρια Δευτεροβάθμιας Εκπαίδευσης θα είναι σε θέση να κατανοήσει έννοιες του προγραμματισμού, όπως μεταβλητές, τελεστές, δεσμευμένες λέξεις, εντολές, αλγοριθμικές δομές, συντακτικά ή λογικά λάθη, των οποίων η ερμηνεία αποδίδεται με χρήση ήχου, από ηλεκτρονική βιβλιοθήκη που θα χρησιμοποιηθεί γι' αυτό τον σκοπό, ώστε να ανταποκριθεί στις νέες μαθησιακές και κοινωνικές ανάγκες, σε ένα σύγχρονο σχολείο που στοχεύει στην ανάπτυξη ικανοτήτων και τεχνικών μάθησης.

Αναλυτικότερα, θα παρουσιαστούν οι μέχρι τώρα διδακτικές προσεγγίσεις, οι γλώσσες προγραμματισμού, τα εκπαιδευτικά εργαλεία, οι δυσκολίες στην κατανόηση της υπολογιστικής σκέψης και της εκμάθησης του προγραμματισμού, τα προγραμματιστικά περιβάλλοντα που χρησιμοποιούν ήχους, οι μελέτες για τα οφέλη των ηχητικών ειδοποιήσεων, οι εφαρμογές και τα διδακτικά σενάρια που χρησιμοποιούν ήχο σε τάξεις μαθητών/τριών Λυκείου.

Δηλαδή, η διατριβή εξετάζει τη χρήση των ηχητικών προειδοποιήσεων ως συμπληρωματικών εργαλείων για τα μαθήματα προγραμματισμού στη Δευτεροβάθμια Εκπαίδευση και αναλύει τον πιθανό αντίκτυπό τους στην ανάπτυξη των ικανοτήτων επίλυσης προβλημάτων από τους/τις μαθητές/τριες. Οι ηχητικές προειδοποιήσεις, διεγείροντας την ψυχοακουστική αντίληψη των μαθητών να αποτελέσουν το εφόδιο στον εκπαιδευτικό να υπερκεράσει τις δυσκολίες που μέχρι τώρα αντιμετώπιζε στο να διδάξει τις έννοιες του προγραμματισμού, αλλά και οι μαθητές διαδραστικά να κατανοήσουν έννοιες που μέχρι τώρα δυσκολεύονταν να αντιληφθούν.

1.3 Ερευνητικά ερωτήματα

Στην παρούσα διατριβή διερευνήθηκαν τα ακόλουθα ερωτήματα:

- α.** Ποιες οι γλώσσες προγραμματισμού που διδάσκονται και ποιες οι αδυναμίες τους;
- β.** Ποια η πρωτοτυπία, η συμβολή και τα παιδαγωγικά οφέλη της νέας προσέγγισης;
- γ.** Οι μαθητές που μαθαίνουν τον δομημένο προγραμματισμό με ενσωματωμένο ήχο επιτυγχάνουν καλύτερη μάθηση/κατανόηση από εκείνους που μαθαίνουν με τη συμβατική διδασκαλία χωρίς ήχο;

Πρόκειται για μια διεπιστημονική μελέτη, όπου εξετάζεται από την πλευρά της παιδαγωγικής, της διδακτικής, της πληροφορικής και της ψυχολογίας, η καλύτερη κατανόηση του αλγοριθμικού προγραμματισμού με ηχητικές ειδοποιήσεις.

1.4 Διάρθρωση της διατριβής

Στο σημείο αυτό, παρατίθεται η διάρθρωση των ενοτήτων που ακολουθούν. Συγκεκριμένα, η διατριβή είναι οργανωμένη ως εξής:

Το παρόν κεφάλαιο αποτελεί το εισαγωγικό κεφάλαιο, όπου παρουσιάζονται ο ρόλος του αλγοριθμικού προγραμματισμού στη Δευτεροβάθμια Εκπαίδευση καθώς και πού αποσκοπεί, τα ερευνητικά ερωτήματα που έχουν τεθεί στην παρούσα διατριβή, όπως η πρωτοτυπία της διατριβής, τα παιδαγωγικά οφέλη της νέας προσέγγισης, οι αντιλήψεις των μαθητών πριν και μετά τη νέα διδακτική προσέγγιση. Επίσης, περιγράφεται ο σκοπός της διατριβής και, φυσικά, η διάρθρωσή της.

Το Κεφάλαιο 2 παρουσιάζει μια επισκόπηση των εκπαιδευτικών εργαλείων που χρησιμοποιούνται στη Δευτεροβάθμια Εκπαίδευση, δηλαδή τι ακολουθείται ως τώρα, πώς γίνεται η διδασκαλία του προγραμματισμού στην εκπαίδευση, ποιο το επικρατέστερο προγραμματιστικό περιβάλλον που ακολουθείται έως σήμερα στο Λύκειο, τις διδακτικές προσεγγίσεις που ακολουθούνται, τη διδακτική προσέγγιση που ακολουθεί η πρόταση της διατριβής, τις δυσκολίες στην εκμάθηση του προγραμματισμού, την ανάγκη για κάτι διαφορετικό στη διδασκαλία, ώστε να προσπεραστούν οι δυσκολίες που προβληματίζουν μαθητές και καθηγητές, την έννοια της ακουστικής αντίληψης και των ακουστικών δεξιοτήτων καθώς και τα προγραμματιστικά περιβάλλοντα που χρησιμοποιούν ήχο, κάποια από τα οποία θα αποτελέσουν έναυσμα, αλλά και εργαλεία, για τις μελέτες που θα παρουσιαστούν.

Το Κεφάλαιο 3 εισάγει με τη συνεργασία των μαθητών/μαθητριών μας, μια νέα εκπαιδευτική προσέγγιση, μια μεθοδολογία βασισμένη σε ηχητικές προειδοποιήσεις, ενώ παράλληλα αναλύει και παρουσιάζει τη μεθοδολογία για την επιλογή των χρησιμοποιούμενων ηχητικών ερεθισμάτων.

Το Κεφάλαιο 4 εξετάζει και παρουσιάζει τη μελέτη της ψυχοακουστικής αξιολόγησης έξι (6) σεναρίων διδασκαλίας, τα οποία χρησιμοποιούν ηχητικά σήματα ειδοποίησης ως συμπληρωματικό εργαλείο για τη διδασκαλία του προγραμματισμού σε ηλεκτρονικούς υπολογιστές.

Το Κεφάλαιο 5 αξιολογεί την αποτελεσματικότητα των ηχητικών προειδοποιήσεων, ενώ στο Κεφάλαιο 6 παρουσιάζονται η αξιολόγηση και τα συμπεράσματα των μαθητών/τριών, που συμμετείχαν στα πειράματα.

Τέλος, τα συμπεράσματα της έρευνας, οι προοπτικές, τα μελλοντικά πλάνα, τα νέα πεδία διερεύνησης και οι εργασίες που μελετώνται να πραγματοποιηθούν, ως συνέχεια της παρούσας εργασίας, συνοψίζονται στο Κεφάλαιο 7.

2. Περιβάλλοντα Προγραμματισμού στη Δευτεροβάθμια Εκπαίδευση

Οι σημερινοί μαθητές έχουν αλλάξει ριζικά και δεν είναι πλέον εκείνοι που το εκπαιδευτικό μας σύστημα σχεδιάστηκε για να διδάξει, διότι την εποχή μας την χαρακτηρίζει κάτι μοναδικό, η γρήγορη διάδοση της ψηφιακής τεχνολογίας (Prensky, 2001). Έχουν περάσει όλη τους τη ζωή περιτριγυρισμένοι από υπολογιστές, βιντεοπαιχνίδια, συσκευές αναπαραγωγής ψηφιακής μουσικής, βιντεοκάμερες, κινητά τηλέφωνα και όλα τα άλλα παιχνίδια και εργαλεία της ψηφιακής εποχής. Ο σημερινός μέσος απόφοιτος έχει περάσει λιγότερες από 5.000 ώρες της ζωής του διαβάζοντας, αλλά πάνω από 10.000 ώρες παίζοντας βιντεοπαιχνίδια και παιχνίδια υπολογιστών και επικοινωνώντας μέσω email, διαδικτύου και κινητών τηλεφώνων, τα οποία αποτελούν αναπόσπαστο κομμάτι της ζωής του. Γι' αυτό και οι σημερινοί μαθητές σκέφτονται και επεξεργάζονται τις πληροφορίες διαφορετικά από τους προκατόχους τους. Αυτή τη διαφορετικότητα οι περισσότεροι εκπαιδευτικοί δεν τη συνειδητοποιούν.

Η δεξιότητα συγγραφής κώδικα προγραμμάτων σε υπολογιστή αποτελεί σημαντικό στοιχείο γραμματισμού στη σημερινή κοινωνία. Η κωδικοποίηση είναι μέρος της λογικής σκέψης, αλλά η εκμάθηση του προγραμματισμού είναι δύσκολη για τους μαθητές της δευτεροβάθμιας εκπαίδευσης, επειδή υπάρχουν δυσκολίες στη συγγραφή, τον εντοπισμό και τον σχεδιασμό του προγράμματος, όπως θα παρουσιάσουμε παρακάτω.

Σ' αυτήν την ενότητα θα παρουσιαστεί τι ακολουθείται ως τώρα και πώς γίνεται η διδασκαλία του προγραμματισμού στην εκπαίδευση, οι διδακτικές προσεγγίσεις, τα εκπαιδευτικά εργαλεία που χρησιμοποιούνται, ποιες οι δυσκολίες κατανόησης των αλγοριθμικών εννοιών, των δομών και της υπολογιστικής σκέψης, καθώς και έρευνες για την ψυχοακουστική αντίληψη των εφήβων.

2.1 Οι απαρχές

Κατ' αρχάς η μεγάλη τομή στη μεθοδολογία του προγραμματισμού έγινε από το βιβλίο *Structures in Programming* (Dahl, Dijkstra, Hoare, 1972). Το πολύτιμο και ξεχωριστό ήταν ότι χρησιμοποίησε μερικούς, αλλά καλά καθορισμένους αλγόριθμους και δομές δεδομένων, διαίρεσε τα προβλήματα σε υποπροβλήματα σύμφωνα με την αρχή

«Divide et impera» (διαίρει και βασίλευε) και, δημιούργησε αφηρημένους αλγόριθμους και τύπους δεδομένων, ανεξαρτήτως της γλώσσας προγραμματισμού για την επίλυση των υπολογιστικών προβλημάτων.

Στο τέλος της δεκαετίας του '70 προσεγγίστηκε και η μεθοδολογία της διδασκαλίας του προγραμματισμού και από μία διαφορετική άποψη, εκείνη με βάση τον τρόπο σκέψης των παιδιών. Αυτή η μέθοδος αναπτύχθηκε κυρίως από τον Papert (1980), ο οποίος εξηγεί τη σημασία της διδασκαλίας των παιδιών στα βασικά της πληροφορικής και πώς μπορεί να τα προετοιμάσει να πετύχουν στον συνεχώς εξελισσόμενο τεχνολογικό κόσμο. Υποστηρίζει ότι οι υπολογιστές έχουν αλλάξει εντελώς τον τρόπο διδασκαλίας στα παιδιά και κάνει λόγο για την αξία της διδασκαλίας με υπολογιστές. Θεωρεί ότι τα παιδιά είναι πολύ ικανά να κυριαρχήσουν στους υπολογιστές και ότι η διδασκαλία υπολογιστικών διαδικασιών όπως η εντοπισμός - διόρθωση λαθών στην τάξη μπορεί να αλλάξει τον τρόπο που μαθαίνουμε τα πάντα. Δείχνει επίσης ότι τα σχολεία που έχουν εξοικείωση με την τεχνολογία μπορούν πραγματικά να βελτιώσουν την κοινωνικοποίηση και την αλληλεπίδραση μεταξύ των μαθητών και μεταξύ των μαθητών και των εκπαιδευτικών. Η τεχνολογία αλλάζει κάθε μέρα, αλλά οι βασικοί τρόποι με τους οποίους οι υπολογιστές μπορούν να μας βοηθήσουν να μάθουμε παραμένουν ίδιοι.

Οι πρώτες προσπάθειες για την παρουσίαση του προγραμματισμού με πιο ενεργό και αποτελεσματικό τρόπο άρχισαν στις αρχές της δεκαετίας του '70 (Noss 1984, Lockwood και Mooney 2018).

Μεταξύ των πιο δημοφιλών ήταν η **Logo** και τα παράγωγά της (Noss 1984). Ορισμένες από αυτές τις προτάσεις προώθησαν τη χρήση οπτικών ή εικονικών γλωσσών προγραμματισμού και την προσομοίωση της δυναμικής επαλήθευσης της εκτέλεσης του προγράμματος (Beanz 2013). Χρησιμοποιεί τεχνικές, όπου οι εντολές του χρήστη εκτελούνται με την κλήση ειδικών λειτουργιών, που λέγονται συναρτήσεις (functions).

Χρησιμοποιείται σήμερα στις πρώτες τάξεις της δευτεροβάθμιας εκπαίδευσης, συγκεκριμένα στην Ελλάδα χρησιμοποιείται στα βιβλία της Γ' Γυμνασίου ως εξεταστέα γλώσσα προγραμματισμού, δίνοντας στους μαθητές τη δυνατότητα επικοινωνίας με τον υπολογιστή και του προγραμματισμού του, πολύ εύκολα και γρήγορα, ακόμη και στην πρώτη επαφή τους με τη γλώσσα. Οι μαθητές λύνουν προβλήματα κατασκευάζοντας μικρά προγράμματα. Η γλώσσα αυτή χρησιμοποιήθηκε

(και χρησιμοποιείται ακόμα) ως ένα τυπικό παράδειγμα ανακαλυπτικού περιβάλλοντος μάθησης.

Η **Pascal** είναι μια «φιλική προς τον χρήστη» γλώσσα προγραμματισμού, για πρώτη επαφή με τον αλγοριθμικό προγραμματισμό (Αλεξανδράκης 2001). Είναι σχετικά εύκολο να μάθεις στη συγκεκριμένη γλώσσα τις θεμελιώδεις έννοιες του δομημένου προγραμματισμού και τις βασικές εντολές. Η Pascal θεωρείται μία από τις πιο κατάλληλες γλώσσες προγραμματισμού για τη διδασκαλία τεχνικών δομημένου προγραμματισμού (Wirth 1973, Αλεξανδράκης 2001). Στη Δευτεροβάθμια Εκπαίδευση στην Ελλάδα διδάχθηκε παράλληλα με το περιβάλλον της Γλωσσομάθειας, που θα παρουσιαστεί στη συνέχεια, από το 1998 ως και το 2001 βάσει του προγράμματος σπουδών και στη συνέχεια προαιρετικά.

Η **C** είναι μια ισχυρή γλώσσα προγραμματισμού με πολλές δυνατότητες αλλά από την άλλη έχει μια δύσκολη σύνταξη. Η **C** παρέχει έναν περιορισμένο αριθμό λέξεων-κλειδιών και έναν μεγάλο αριθμό αριθμητικών και λογικών τελεστών. Μερικές φορές ονομάζεται "assembly υψηλού επιπέδου" ή "φορητή assembly". Έτσι, η **C** θεωρείται μια δύσκολη γλώσσα προγραμματισμού, τόσο για τη διδασκαλία όσο και για τη μάθηση, εξαιτίας της μινιμαλιστικής της φύσης (Kernighan & Ritchie 1988). Χρησιμοποιείται εναλλακτικά αντί της Python στα Επαγγελματικά Λύκεια.

2.2 Σύγχρονες διδακτικές προσεγγίσεις

Έκτοτε έχουν αναπτυχθεί αρκετές μέθοδοι για τη διδασκαλία του προγραμματισμού. Μερικές από αυτές έχουν ήδη ξεπεραστεί, οι περισσότερες από αυτές, ωστόσο, εξακολουθούν να χρησιμοποιούνται σε ένα συγκεκριμένο επίπεδο εκπαίδευσης. Αν και μερικές από αυτές σχετίζονται με τη μεθοδολογία σχεδιασμού προγραμμάτων, η πλειονότητα των μεθόδων είναι ανεξάρτητες από εκείνη.

Ας σημειωθεί ότι, η πλειονότητα των εκπαιδευτικών, προτιμά να χρησιμοποιεί διάφορες μεθόδους ταυτόχρονα, αντί να ακολουθήσει μια μόνο μέθοδο. Η παραδοσιακή διδακτική προσέγγιση, λοιπόν, ήταν η εκμάθηση συγκεκριμένων γλωσσών προγραμματισμού, που αποτελεί ένα συμπεριφοριστικό μοντέλο κλειστό και δασκαλοκεντρικό. Η αποτελεσματικότητα εξαρτάται από την ποιότητα και ποσότητα των γνώσεων που διαθέτει ο εκπαιδευτικός πάνω στο αντικείμενό του, καθώς και από τη μεταδοτικότητα του.

Σύμφωνα με τους (Jerinic et al. 2014), υπάρχουν τόσοι πολλοί τρόποι μάθησης όσοι και οι μαθητές. Στο πλαίσιο αυτών των προσπαθειών αναδύεται η ιδέα της ανακαλυπτικής προσέγγισης του προγραμματισμού (programming by discovery). Σύμφωνα με αυτό το μοντέλο, ο μαθητής ανακαλύπτει τα σφάλματα και τις παρανοήσεις και αυτοαναβαθμίζει τον ρόλο του. Ένα άλλο μοντέλο είναι το εποικοδομητικό, όπου η γνώση δε μεταδίδεται, αλλά οικοδομείται. Κέντρο βάρους αυτού του μοντέλου είναι οι ενέργειες και οι ανάγκες του κάθε μαθητή.

Στις νέες προσεγγίσεις διδασκαλίας, επιδιώκεται η αφομοίωση στοιχείων και αρχών της διερευνητικής και συνεργατικής μάθησης. Ο στόχος του νέου μοντέλου είναι να βελτιώσει τη μαθησιακή διαδικασία και να ξεπεράσει τα εμπόδια που προέκυψαν κατά την εφαρμογή των προηγούμενων διδακτικών προσεγγίσεων. Η διαδικασία αποτελείται από πέντε (5) διαφορετικά, αλλά επαναλαμβανόμενα στάδια, την Ερώτηση, την Ενεργή έρευνα, τη Δημιουργία, τη Συζήτηση και την Αξιολόγηση.

Η έρευνα των (Hsu & Hwang 2017) βασίστηκε σε αυτήν τη μεθοδολογία. Βασιζόμαστε ομοίως σε αυτήν τη διδακτική προσέγγιση των Hsu & Hwang και στο παραπάνω συνδυαστικό νέο μοντέλο. Στη μελέτη μας, υιοθετήθηκε η ενότητα μαθημάτων «βασικές έννοιες του δομημένου προγραμματισμού» (δομή ακολουθίας, δομή επιλογής, δομή επανάληψης) στο Λύκειο, για τη διεξαγωγή των πειραμάτων με τη χρησιμοποίηση ήχου.

Υιοθετούνται οι βασικές ιδέες του Piaget, σύμφωνα με τις οποίες η μάθηση δεν μεταδίδεται, αλλά χτίζεται μέσω της ενεργούς συνεργασίας μεταξύ του δασκάλου και των μαθητών. Βασίζεται επίσης στη θεωρία ανακάλυψης του J. Bruner, σύμφωνα με την οποία οι μαθητές ανακαλύπτουν τη γνώση (αναζήτηση) μέσω διεργασιών ανακάλυψης, όπως η εκτέλεση ενός έτοιμου προγράμματος με διαφορετικά δεδομένα εισόδου μέσω δοκιμών και πειραματισμών.

2.3 Εκπαιδευτικά εργαλεία που χρησιμοποιούνται σήμερα

Σήμερα στην εκπαίδευση χρησιμοποιούνται περιβάλλοντα προγραμματισμού τόσο στο εξωτερικό όσο και στην Ελλάδα όπως παρουσιάζονται στον Πίνακα 1 και περιγράφονται παρακάτω.

Πίνακας 1 Πίνακας εκπαιδευτικών περιβαλλόντων προγραμματισμού που χρησιμοποιούνται σήμερα

Εκπαιδευτικό Περιβάλλον	Βαθμίδα Εκπαίδευσης	Χώρα
LEGO Mindstorms NXT	Δημοτικό/Γυμνάσιο	Εξωτερικό/Ελλάδα
Python	Τριτοβάθμια/ Επαγγελματικό Λύκειο	Εξωτερικό/Ελλάδα
Arduino	Γυμνάσιο/Επαγγελματικό Λύκειο	Εξωτερικό/Ελλάδα
LittleBits	Δημοτικό/Γυμνάσιο/ Επαγγελματικό Λύκειο	Εξωτερικό/Ελλάδα
Alice	Λύκειο	Εξωτερικό/Ελλάδα
Γλωσσομάθεια	Λύκειο	Ελλάδα

Η εκπαιδευτική πλατφόρμα **LEGO Mindstorms NXT** της ομώνυμης εταιρείας Lego δίνει τη δυνατότητα σε μια ομάδα 2-6 μαθητών (8 ετών και άνω) να προσομοιώσουν σχεδόν όλους τους σύγχρονους αυτοματισμούς και να προσεγγίσουν τα συστήματα αυτόματου ελέγχου μέσω ενός δημιουργικού και ευχάριστου μαθησιακού περιβάλλοντος. Η χρήση του περιλαμβάνει τόσο προγραμματισμό όσο και απλές εφαρμογές μηχανικού χαρακτήρα (Ringwood, Monaghan & Maloco 2005, Χρονάκη & Κούριας). Χρησιμοποιείται στο Δημοτικό και στη Δευτεροβάθμια εκπαίδευση στο Γυμνάσιο.

Η **Python** είναι μια ευρέως χρησιμοποιούμενη γλώσσα σεναρίων (scripting) υψηλού επιπέδου, γενικής χρήσης, ερμηνευτικής και δυναμικής (Αγγελιδάκης 2015). Η φιλοσοφία σχεδίασής της τονίζει την αναγνωσιμότητα του κώδικα και η σύνταξή της επιτρέπει στους προγραμματιστές να εκφράζουν έννοιες σε λιγότερες γραμμές κώδικα από ό, τι είναι δυνατόν σε γλώσσες όπως C ή Java. Η γλώσσα παρέχει δομές που

επιτρέπουν τη σύνταξη σαφών προγραμμάτων, σε μικρές και μεγάλες κλίμακες. Χρησιμοποιείται, βάσει του προγράμματος σπουδών, στα Επαγγελματικά Λύκεια.

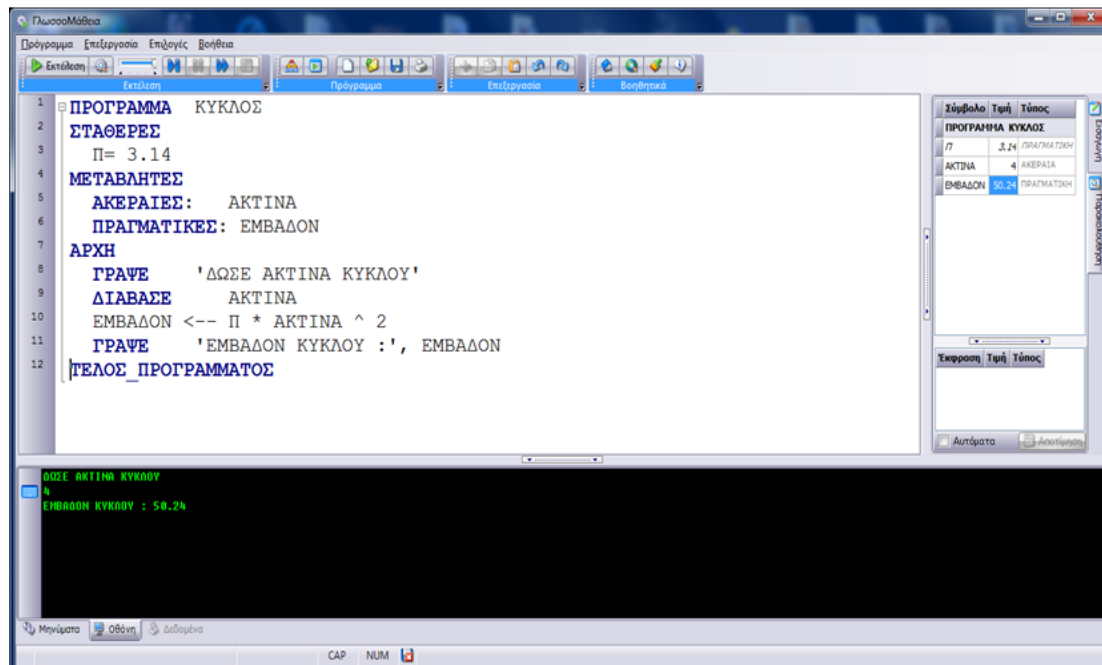
Το **Arduino** είναι μια μικρή συσκευή (μικροεπεξεργαστής) συνδεδεμένη στον υπολογιστή (είτε ασύρματα είτε μέσω USB). Οι καλλιτέχνες, οι σχεδιαστές, οι αρχιτέκτονες, οι μαθητές, οι σπουδαστές και οι φοιτητές μαθαίνουν να προγραμματίζουν και να χτίζουν ολόκληρα έργα για αυτοματοποίηση και ρομποτικές εφαρμογές (Παπάζογλου & Λιώνης 2014). Έτσι στα παιδιά, κάνοντας χρήση leds (λαμπτήρες), γίνεται μια εισαγωγή στην εκμάθηση του προγραμματισμού. Η πλατφόρμα Arduino βασίζεται σε τεχνολογίες ανοιχτού κώδικα και το περιβάλλον ανάπτυξης λογισμικού στην καλωδίωση. Χρησιμοποιείται γλώσσα προγραμματισμού ανοιχτού κώδικα τύπου C, η οποία όμως θεωρείται ευκολότερη στην κατανόηση του προγραμματισμού (Πουλάκης 2015). Οι ερευνητικές μελέτες που πραγματοποιήθηκαν τα τελευταία 10 χρόνια εξετάζουν τη θετική επίδραση των ρομπότ, στην αντίληψη και την εκπαίδευση των παιδιών (Toh, Causo, Tzuo, Chen, & Yeo 2016). Η πλατφόρμα Arduino χρησιμοποιείται στα Γυμνάσια και στα επαγγελματικά Λύκεια, όσον αφορά στη Δευτεροβάθμια Εκπαίδευση.

Ανάλογη πλατφόρμα είναι και η πλατφόρμα **LittleBits**. Αποτελείται από ηλεκτρονικά δομικά στοιχεία, εύκολα στη χρήση, που εξουσιοδοτούν τους μαθητές να εφεύρουν οτιδήποτε, χωρίς καλώδια, χωρίς να απαιτείται προγραμματισμός και χρησιμοποιείται από την ηλικία των 8 ετών και άνω.

Το **Alice** αποτελεί ένα από τα πιο καινοτόμα προγράμματα για την εκμάθηση και την εισαγωγή της Πληροφορικής στη Δευτεροβάθμια Εκπαίδευση. Παράλληλα αποτελεί κι ένα από τα πιο μακρόχρονα ερευνητικά project το οποίο έχει τις ρίζες του στις αρχές της δεκαετίας του 1990 (Alice, 2021). Στα πρώτα του βήματα χρησιμοποιήθηκε ως πρόγραμμα εικονικής πραγματικότητας και σταδιακά στη συνέχεια ως παιδαγωγικό εργαλείο. Στόχος ήταν η προσέλκυση του ενδιαφέροντος περισσότερων μαθητών για σπουδές στον κλάδο. Εισήγαγε βασικές έννοιες του αντικειμενοστραφούς προγραμματισμού με έναν προσιτό και διασκεδαστικό τρόπο σε εφήβους. Θεωρήθηκε καταλληλότερο για χρήση στις τελευταίες τάξεις του Λυκείου έχοντας βασικό πλεονέκτημα την ευκολότερη και πιο ομαλή μετάβαση στις πραγματικές γλώσσες προγραμματισμού όπως είναι η Java.

Η «**Γλωσσομάθεια**» είναι ένα περιβάλλον προγραμματισμού (με χρήση ψευδοκώδικα), (Σχήμα 1α) βασισμένο στην Pascal, το οποίο χρησιμοποιείται κυρίως

στη Δευτεροβάθμια Εκπαίδευση στην Ελλάδα (Spinnet, Algorithmos, Ebooks.edu.gr, 2020), ιδιαίτερα στη Γ Λυκείου, για την πρακτική εφαρμογή της ύλης του μαθήματος, μέχρι πρότινος ΑΕΠΠ, τώρα Πληροφορική, το οποίο είναι πανελλαδικά εξεταζόμενο μάθημα για την πρόσβαση στην τριτοβάθμια εκπαίδευση. Πρόκειται για ένα εκπαιδευτικό πακέτο που αναπτύχθηκε με επίκεντρο τα μαθήματα εργαστηριακής υποστήριξης που σχετίζονται με την καλλιέργεια αλγοριθμικών και αναλυτικών σκέψεων και την ανάπτυξη μεθοδολογικών δεξιοτήτων για τους μαθητές. Κατά τη σύνταξη και εκτέλεση του προγράμματος, η οθόνη είναι γεμάτη πληροφορίες, όμως αρκετά μηνύματα που εμφανίζονται σε αυτήν δεν τα αντιλαμβάνεται ένα μεγάλο ποσοστό μαθητών (Spinnet, 2020), αν δεν επιμείνει ο διδάσκων στην επεξήγησή τους, όπως για παράδειγμα αναμονή για καταχώρηση δεδομένων, δεδομένα λογικού τύπου, επιτυχής καταχώριση κ.λπ. Να σημειωθεί ότι στο περιβάλλον της Γλωσσομάθειας υπάρχει δυνατότητα παρουσίασης διαγραμμάτων ροής, δηλαδή μια αυτοματοποιημένη μετάβαση από τον ψευδοκώδικα σε διαγραμματική αναπαράσταση, αλλά καμία ειδοποίηση ήχου.



```
1  ΠΡΟΓΡΑΜΜΑ ΚΥΚΛΟΣ
2  ΣΤΑΘΕΡΕΣ
3  Π = 3.14
4  ΜΕΤΑΒΛΗΤΕΣ
5  ΑΚΕΡΑΙΕΣ: ΑΚΤΙΝΑ
6  ΠΡΑΓΜΑΤΙΚΕΣ: ΕΜΒΑΔΟΝ
7  ΑΡΧΗ
8  ΓΡΑΨΕ 'ΔΩΣΕ ΑΚΤΙΝΑ ΚΥΚΛΟΥ'
9  ΔΙΑΒΑΣΕ ΑΚΤΙΝΑ
10 ΕΜΒΑΔΟΝ <-- Π * ΑΚΤΙΝΑ ^ 2
11 ΓΡΑΨΕ 'ΕΜΒΑΔΟΝ ΚΥΚΛΟΥ :', ΕΜΒΑΔΟΝ
12 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Σύμβολο	Τιμή	Τύπος
ΠΡΟΓΡΑΜΜΑ ΚΥΚΛΟΣ		
7	3.14	ΠΡΑΓΜΑΤΙΚΗ
ΑΚΤΙΝΑ	4	ΑΚΕΡΑΙΑ
ΕΜΒΑΔΟΝ	50.24	ΠΡΑΓΜΑΤΙΚΗ

```
ΔΩΣΕ ΑΚΤΙΝΑ ΚΥΚΛΟΥ
4
ΕΜΒΑΔΟΝ ΚΥΚΛΟΥ : 50.24
```

Σχήμα 1. Το περιβάλλον προγραμματισμού της Γλωσσομάθειας (ψευδοκώδικας)

Οι γλώσσες προγραμματισμού Pascal και Python, αλλά και η Γλωσσομάθεια, που χρησιμοποιούνται για την εκμάθηση του αλγοριθμικού προγραμματισμού στη Δευτεροβάθμια Εκπαίδευση, παρουσιάζουν όλες τις γενικότερες δυσκολίες για την

κατανόησή τους από τους μαθητές της αντίστοιχης βαθμίδας, που παρουσιάζονται στην επόμενη ενότητα.

2.4 Δυσκολίες στην κατανόηση του αλγοριθμικού προγραμματισμού και της υπολογιστικής σκέψης

Η εισαγωγή του προγραμματισμού στη Δευτεροβάθμια Εκπαίδευση είχε άμεση συνέπεια τη γένεση νέων προβλημάτων που θα έπρεπε να επιλυθούν, όπως τα προβλήματα επικοινωνίας που αντιμετώπιζαν οι μαθητές με το μηχάνημα μέσω της κωδικοποίησης, η οποία απαιτεί έναν ακριβή τρόπο σκέψης, που γίνεται με συγκεκριμένη σύνταξη (Αβούρης 2000, Γρηγοριάδου et al. 2009, Robins et al. 2003, Τζιμογιάννης et al. 2019).

Η διδασκαλία της Πληροφορικής αποδείχτηκε περίπλοκη στη διδακτική πράξη. Πολλοί μαθητές και σπουδαστές αντιμετωπίζουν προβλήματα κατανόησης εννοιών, παρανόησης των ήδη χρησιμοποιούμενων τεχνικών, δυσκολίες στο να επιλέξουν την κατάλληλη μέθοδο όταν καλούνται να επιλύσουν ένα πρόβλημα, το οποίο σχετίζεται με τον προγραμματισμό. Η διαπίστωση αυτή κατέστησε αναγκαία τη συστηματική διερεύνηση των θεμάτων που σχετίζονται με τη διδασκαλία/εκμάθηση της Πληροφορικής, έτσι ώστε τελικά η διδασκαλία της να καταστεί πιο αποτελεσματική, δηλαδή οι μαθητές ή οι σπουδαστές να μαθαίνουν και να κατανοούν «καλύτερα». Πιο συγκεκριμένα, οι μαθητές, λόγω των παραπάνω προβλημάτων, απογοητεύονται και αποφεύγουν τα «δύσκολα» σημεία ή ακόμη και τον προγραμματισμό γενικότερα, ενώ οι εκπαιδευτικοί δυσκολεύονται στη διδασκαλία μερικών εννοιών.

Πολλές φορές οι μαθητές ερμηνεύουν τα προγράμματα και γενικότερα την επικοινωνία τους με ένα πληροφορικό σύστημα, όχι με βάση τους συντακτικούς και σημασιολογικούς κανόνες (syntax, semantics) της χρησιμοποιούμενης γλώσσας (προγραμματισμού, γλώσσας εντολών κ.λπ.), αλλά αποδίδοντας στο αντίστοιχο πρόγραμμα ένα νόημα ευρύτερο του πραγματικού, σαν να επρόκειτο για έναν διάλογο μ' άνθρωπο (ανθρωπομορφική αντίληψη) (Pea, 1986). Η ανθρωπομορφική αντίληψη έχει διάφορες εκφράσεις, όπως αναφέρει και ο Pea. Οι μαθητές εκπλήσσονται από τα αποτελέσματα της εκτέλεσης και μια «προσομοίωση» της εκτέλεσής του βήμα-με-βήμα είναι πολλές φορές αναγκαία για να γίνει κατανοητή η ροή του προγράμματος.

Ανάλογη αντίληψη με αυτήν, είναι εκείνη κατά την οποία οι νεαροί μαθητές μεταφέρουν στο προγραμματιστικό περιβάλλον στρατηγικές επίλυσης προβλημάτων από άλλα περιβάλλοντα, χωρίς να κάνουν τις απαραίτητες τροποποιήσεις. Οι μαθητές λαμβάνουν υπ' όψη τους το γεγονός ότι όλα τα δεδομένα ενός προβλήματος εμπεριέχονται στο πρόγραμμα και θεωρούν αυτόματα (και βεβαίως λανθασμένα) ότι το πληροφορικό σύστημα (δηλαδή ο Η/Υ και το λογισμικό) θα δώσει τη σωστή απάντηση, χωρίς να λαμβάνουν ταυτόχρονα υπ' όψη τους και το γεγονός, ότι η λειτουργία του προγράμματος είναι γραμμική κι όχι «παράλληλη» (ο Pea προτείνει ακριβώς τον όρο «παραλληλισμός» για να χαρακτηρίσει αυτή την κατηγορία λαθών).

Επίσης, διαπιστώνεται ελλιπής κατανόηση της λειτουργίας του πληροφορικού συστήματος, καθώς και ότι οι λανθασμένες αντιλήψεις των μαθητών δεν είναι μόνο αποτέλεσμα της άγνοιας, αλλά και το αποτέλεσμα μιας προγενέστερης γνώσης, που αποδεικνύεται λανθασμένη ή και αρκετές φορές ακατάλληλη. Η αντίληψη του παραλληλισμού, την οποία αναφέραμε, μπορεί να οφείλεται στην ελλιπή κατανόηση των αρχών λειτουργίας των πληροφορικών συστημάτων κι ιδιαίτερα στο γεγονός ότι τα προγράμματα και μάλιστα εκείνα τα προγράμματα με μικρό πλήθος εντολών, που συνήθως κατασκευάζουν οι μαθητές, εκτελούνται πολύ γρήγορα: «it looks at the program all at once because it is so fast» σχολίασε ένας σπουδαστής όταν ρωτήθηκε σε μια σχετική έρευνα (Pea 1986). Η απλοϊκή μορφή του ανθρωπομορφισμού (ο μαθητής που προσπαθεί να μάθει τ' όνομα του Η/Υ) μπορεί να βρίσκει την πηγή της στην περιρρέουσα ατμόσφαιρα της καθημερινής ζωής, στην οποία οι Η/Υ (παλιότερα "ηλεκτρονικοί εγκέφαλοι") εμφανίζονται να πραγματοποιούν ολόένα και πιο εντυπωσιακά επιτεύγματα, άρα θα μπορούσε κανείς να συμπεράνει ότι οι ΗΥ διαθέτουν ένα είδος «ευφυΐας».

Το λάθος κατά το οποίο οι μαθητές αποδίδουν τιμές χωρίς να λαμβάνουν υπόψη τους τον τρόπο με τον οποίο λειτουργεί το σύστημα, οφείλεται ίσως στην εμπειρία από άλλα μαθήματα στα οποία βέβαια η σειρά με την οποία περιγράφονται τα δεδομένα δεν παίζουν ουσιαστικό ρόλο στην επίλυση ενός προβλήματος. Παρόλα αυτά, αν και εντοπίζονται οι πιο συνηθισμένες δυσκολίες / παρανοήσεις που αντιμετωπίζουν οι μαθητές, όταν μαθαίνουν πώς να προγραμματίζουν, εξακολουθούν να υπάρχουν σαφείς στρατηγικές για την αντιμετώπισή τους.

Οι μαθητές πρέπει να διδάσκονται έννοιες προγραμματισμού ανεξαρτήτως των συγκεκριμένων εφαρμογών και των γλωσσών προγραμματισμού (Stephenson, Gal-Ezer, Haberman & Verno 2005, Szlávi & Zsakó 2006). Όλοι τους έχουν διαφορετικές ανάγκες και δυσκολίες.

Ο Du Boulay (1989) περιγράφει πέντε βασικές περιοχές στις οποίες, κατά τη γνώμη του, εντοπίζεται η δυσκολία της εκμάθησης του προγραμματισμού (Ξυνόγαλος 2000, Ξυνόγαλος 2002, Ξυνόγαλος & Σατρατζέμη 2002), τον **Προσανατολισμό** (Ανακάλυψη της χρησιμότητας και των πλεονεκτημάτων του προγραμματισμού), τη **θεωρητική μηχανή** (γενικές ιδιότητες του μηχανήματος), τη **Σημειογραφία** (συντακτικών κανόνων και των σημασιολογικών κανόνων), τις **Δομές** (Οι Spohrer et al. (1989) τα αναφέρουν με τον όρο goal/plan (στόχος /σχέδιο)) και τον **Έλεγχο του προγραμματισμού** (Εκμάθηση της ικανότητας προσδιορισμού, ανάπτυξης, δοκιμής και εντοπισμού σφαλμάτων ενός προγράμματος, χρησιμοποιώντας τα διαθέσιμα εργαλεία). Η τελευταία περιοχή δυσκολίας παραμελείται συχνά στα μαθήματα προγραμματισμού, παρά τη σπουδαιότητά της. Έχει να κάνει με τις βοηθητικές δεξιότητες που είναι απαραίτητες για τον προγραμματισμό και αφορούν την ικανότητα προσαρμογής και ελέγχου ενός περιβάλλοντος στον υπολογιστή, που θα χρησιμοποιηθεί για τη συγγραφή κώδικα, τη μεταγλώττιση και τον εντοπισμό λαθών (debugging) ενός προγράμματος. Οι μαθητές, ορισμένες φορές, έχουν δυσκολία να εγκλιματισθούν στο περιβάλλον ανάπτυξης προγραμμάτων και να μάθουν να χρησιμοποιούν τα διαθέσιμα εργαλεία, πριν ακόμα αρχίσουν να εξετάζουν την ίδια τη γλώσσα προγραμματισμού.

Ο Du Boulay (1989) επίσης, θεωρεί ότι τα λάθη των μαθητών παρατηρούνται λόγω της **Κακής εφαρμογής της αναλογίας**, της **Υπεργενίκευσης** και του **Λάθους χειρισμού της πολυπλοκότητας και της αλληλεπίδρασης**.

Ο Rea έχει εντοπίσει ορισμένα πολύ σημαντικά, εννοιολογικά, ανεξαρτήτως της γλώσσας, "σφάλματα", στον τρόπο με τον οποίο οι αρχάριοι προγραμματίζουν και κατανοούν τα προγράμματα (Rea, 1986). Οι μαθητές θεωρούν ότι οι υπολογιστές "υπερβαίνουν τις πληροφορίες που δίνονται" στο πρόγραμμα. Επιπλέον, έχει παρατηρηθεί ότι αρκετοί αποτυγχάνουν να μεταφράσουν μια λύση που προκύπτει από την εκφώνηση του προβλήματος σε σωστό κώδικα (Weigend 2006). Ο λόγος μπορεί να είναι ότι οι μαθητές δεν εκπαιδεύονται να μετασχηματίζουν έννοιες που

αντιλαμβάνονται, σε κώδικα (Weigend 2006). Τέτοια εμπόδια, θα μπορούσαν να ξεπεραστούν βοηθώντας τους μαθητές να αναπτύξουν δεξιότητες επίλυσης προβλημάτων, εκτός από τη λογική αιτιολόγηση.

Οι τεχνητές γλώσσες έχουν περιορισμένο λεξιλόγιο σε σχέση με τις φυσικές. Αυτό δημιουργεί πρόβλημα στην κατανόηση των εντολών. Οι καθηγητές χρησιμοποιούν τη φυσική γλώσσα για να αποκωδικοποιήσουν και να γνωστοποιήσουν την έννοια των λειτουργιών προγραμματισμού (Du Boulay 1989). Από την άλλη πλευρά, έχει αποδειχθεί ότι η διάδραση διευκολύνει την κατανόηση των εννοιών προγραμματισμού (Τσολακίδης & Φωκίδης 2007, Βακαλούδη 2003). Οι συνηθέστερες πρακτικές στα σχολεία περιλαμβάνουν περιβάλλοντα με οπτικοακουστική ανατροφοδότηση (Ασλανίδου 2010, Kelleher & Pausch 2005).

Οι δυσκολίες που συναντούν οι μαθητές κατά την εκμάθηση του προγραμματισμού, σχετίζονται, επίσης, με την παρανόηση των βασικών δομών, των κανόνων της γλώσσας προγραμματισμού, των ιδιοτήτων της «μηχανής» που ελέγχει ο μαθητής και την ικανότητα καθορισμού, ανάπτυξης, ελέγχου και εντοπισμού - διόρθωση λαθών ενός προγράμματος με τα διαθέσιμα εργαλεία (Ξυνόγαλος & Σατρατζέμη 2001, Ξυνόγαλος 2002).

Πιο συγκεκριμένα, όσον αφορά στις ιδιότητες της «μηχανής», ο μαθητής (Ξυνόγαλος 2002, Δαγδιλέλης 1996) δε μπορεί να εντοπίσει εύκολα τη σχέση που υπάρχει ανάμεσα στο πρόγραμμα (γραμμένο σε μια γλώσσα προγραμματισμού) και τον μηχανισμό εκτέλεσής του, κάνει, κατά κανόνα, δικές του υποθέσεις για τον τρόπο με τον οποίο λειτουργεί εσωτερικά το υπολογιστικό σύστημα και δε μπορεί να αντιληφθεί το βαθμό λεπτομέρειας που απαιτεί ο προγραμματισμός και την αυστηρότητα με την οποία πρέπει να τηρούνται οι κανόνες του προγραμματιστικού συστήματος.

Όσον αφορά στη γλώσσα προγραμματισμού, ο μαθητής (Ξυνόγαλος 2000 και 2002, Ξυνόγαλος & Σατρατζέμη 2004, Ξυνόγαλος et al. 2005, Σατρατζέμη et al. 2006) δυσκολεύεται μερικές φορές από τις αγγλικές λέξεις που περιλαμβάνουν οι γλώσσες προγραμματισμού, είτε γιατί τις ερμηνεύει με λανθασμένο τρόπο, είτε γιατί έχουν διαφορετική σημασία στη γλώσσα προγραμματισμού από ότι στην τρέχουσα Αγγλική γλώσσα. Εξάλλου, η τυπική γλώσσα που χρησιμοποιείται είναι δύσκολα ερμηνεύσιμη από τους μαθητές, οι οποίοι αγνοούν τους μηχανισμούς με τους οποίους λειτουργεί το Πληροφορικό σύστημα. Όσον αφορά στο χειρισμό των δομών της γλώσσας

προγραμματισμού, ο αρχάριος προγραμματιστής συναντά μεγάλες δυσκολίες, κάποιες από τις οποίες οφείλονται στις λανθασμένες αντιλήψεις του.

Οι αντιλήψεις αυτές, με τη σειρά τους, οφείλονται, συχνά, στο γεγονός ότι οι αρχάριοι εκφράζουν τη λύση ενός προβλήματος σε κάποια γλώσσα προγραμματισμού, βασιζόμενοι στη φυσική γλώσσα που χρησιμοποιούν στις καθημερινές συνομιλίες τους.

Η συνηθέστερη λοιπόν κατηγορία λανθασμένων αντιλήψεων στους μαθητές, έχει την πηγή της στην καθημερινή ζωή. Οι συχνότερες δυσκολίες ενός αρχάριου προγραμματιστή εντοπίζονται στην ερμηνεία των εντολών σε μια γλώσσα προγραμματισμού, με σημείο αναφοράς τις καθημερινές συνομιλίες μεταξύ ανθρώπων - οι οποίες βέβαια είναι συχνά ελλειπτικές, μεταφορικές κλπ. Δυσκολεύεται να κατανοήσει τη δομή επιλογής (ποια η ροή του προγράμματος όταν είναι αληθής ή ψευδής μια συνθήκη), ιδιαίτερα όταν αυτή περιλαμβάνει στη συνθήκη σύνθετες λογικές εκφράσεις (AND, NOT, OR) και να παρακολουθήσει τη λειτουργία των εμφωλευμένων εντολών. Αντιμετωπίζει και μαθαίνει ευκολότερα νέες δομές, όταν διαθέτει καλύτερο μαθηματικό υπόβαθρο. Αδυνατεί να ορίσει με επιτυχία τη συνθήκη εξόδου από ένα βρόχο. Δε μπορεί να καθορίσει την εμβέλεια του βρόχου και να προσδιορίσει την αρχή και το τέλος τους, καθώς και τις εντολές που επαναλαμβάνονται. Δυσκολεύεται να κατανοήσει τον τρόπο με τον οποίο λειτουργούν οι μεταβλητές.

Ακόμη και με τις διαγραμματικές τεχνικές (διδασκτική προσέγγιση), όπου σ' ένα διάγραμμα ροής σχήματα αντιστοιχίζονται με εντολές/ενέργειες, έλεγχο συνθηκών, οι μαθητές δυσκολεύονται τόσο στην επιλογή των σχημάτων, όσο και στον τρόπο σύνδεσής τους. Στις παρακάτω υποενότητες καταγράφονται ανά κατηγορία οι δυσκολίες που αντιμετωπίζουν οι μαθητές (Τζιμογιάννης et al. 2019).

2.4.1 Χρήση μεταβλητών

Η έννοια και ο ρόλος της μεταβλητής (αλγοριθμικής και προγραμματιστικής) υπήρξαν τα αντικείμενα εκτεταμένων ερευνών (για παράδειγμα Schneiderman 1980, Mayer 1981, Bayman & Mayer 1983, Rouchier & Samurçay 1984, Samurçay 1989, Dagdilelis 1986, Rogalski 1989).

Ο αρχάριος προγραμματιστής δυσκολεύεται να κατανοήσει τον τρόπο με τον οποίο λειτουργούν οι μεταβλητές, την ανάγκη αρχικοποίησης των μεταβλητών και την ενημέρωση καθώς και τον έλεγχο των τιμών των μεταβλητών σε βρόχους.

Η μεταβλητή έχει πολλές φορές μια εκφορά (σύνταξη) η οποία είναι συγγενής με των μαθηματικών, αλλά με τελείως διαφορετικό νόημα.

Αλλά και όταν ο αρχάριος προγραμματιστής κατανοεί ότι η παραπάνω έκφραση αντιστοιχεί σε μια εκχώρηση και όχι σε μια εξίσωση, σε πολλές περιπτώσεις, δυσκολεύεται να αντιληφθεί το αποτέλεσμα της εκχώρησης. Θα πρέπει ίσως να επισημάνουμε ότι οι αντιλήψεις αυτές, ίσως να είναι πιο σπάνιες σε προγραμματιστικά περιβάλλοντα στα οποία η εκχώρηση και η εξίσωση εκφράζονται με διαφορετικά σύμβολα, όπως στην Pascal ή στη Γλωσσομάθεια. Τέλος, δυσκολεύονται πολλές φορές να αντιληφθούν το γεγονός ότι αρκετές φορές οι πράξεις επιτρέπονται μεταξύ μεταβλητών του ίδιου ή συγγενούς τύπου και, επίσης, ότι υπάρχουν πράξεις που είναι επιτρεπτές για μια κατηγορία μεταβλητών και απαγορευμένες για τις υπόλοιπες κατηγορίες.

2.4.2 Είσοδος δεδομένων

Κατά τη διαχείριση των μεταβλητών που χρησιμοποιούνται για είσοδο δεδομένων, δηλαδή κατά την εισαγωγή των δεδομένων από το πληκτρολόγιο, ο αρχάριος χρήστης ερμηνεύει την "ακινησία" της οθόνης σαν ένα είδος βλάβης ή προβλήματος και δυσκολεύεται να εξηγήσει την κίνηση της πληροφορίας (εισαγωγή από πληκτρολόγιο και αποθήκευση της σε μια θέση μνήμης Α).

Αυτό συμβαίνει είτε πρόκειται για ένα δεδομένο, είτε για περισσότερα.

2.4.3 Δομές επιλογής

Οι (Τζιμογιάννης et al. 2019) επισημαίνουν προβλήματα στη χρήση της δομής επιλογής από τους μαθητές. Μερικές φορές δυσκολεύονται να κατανοήσουν ότι οι εντολές που βρίσκονται αμέσως μετά τη δομή επιλογής εκτελούνται πάντα, ανεξαρτήτως της αποτίμησης της συνθήκης. Θεωρούν κάποιες φορές ότι εκτελούνται τόσο το μπλοκ εντολών **Αν...τότε** όσο και εκείνο του **Αλλιώς** ή άλλες φορές εκτελούν το μπλοκ εντολών του **Αν...τότε**, είτε η συνθήκη που περιλαμβάνει είναι αληθής, είτε είναι ψευδής. Με άλλα λόγια δεν κατανοούν ότι ανάλογα με την ισχύ μιας συνθήκης εκτελείται ή δεν εκτελείται το εμφωλευμένο σύνολο εντολών. Πιστεύουν ότι αν μια συνθήκη σε μια δομή επιλογής είναι ψευδής, τότε σταματά η ροή του προγράμματος.

Κάποιοι μαθητές, επίσης, πιστεύουν ότι για να λειτουργήσει σωστά η δομή επιλογής απαιτείται πάντοτε η λογική έκφραση να είναι αληθής, όπως χρησιμοποιείται στη φυσική γλώσσα, ενώ αντίθετα αν μια λογική έκφραση αποτιμηθεί ως ψευδής, τότε η συνθήκη είναι λανθασμένη.

2.4.4 Δομές επανάληψης

Αν για κάποιο έμπειρο προγραμματιστή, τα στοιχεία της δομής μιας επαναληπτικής διαδικασίας περνούν σχεδόν απαρατήρητα, για τους αρχάριους, όπως οι μαθητές, η διαχείρισή τους αποτελεί ένα περίπλοκο πρόβλημα.

Οι συνηθέστερες δυσκολίες ενός αρχάριου προγραμματιστή είναι ότι προτιμά να χρησιμοποιεί ακολουθίες επαναλαμβανόμενων εντολών στα προγράμματά του, αντί για βρόχους, διαθέτει ανεπαρκή νοητά μοντέλα για τις επαναληπτικές δομές, δυσκολεύεται να επιλέξει την κατάλληλη, για κάθε πρόβλημα, επαναληπτική δομή, αδυνατεί να ορίσει με επιτυχία τη συνθήκη εξόδου από ένα βρόχο, δε μπορεί να καθορίσει την εμβέλεια του βρόχου και να προσδιορίσει την αρχή και το τέλος τους, καθώς και τις εντολές που επαναλαμβάνονται και δεν προσδιορίζει το πλήθος των επαναλήψεων.

Η χρήση των βρόχων και γενικότερα των επαναληπτικών διαδικασιών, παραμένει πολύ συχνά ένα θέμα το οποίο οι μαθητές δεν κατανοούν απόλυτα. Οι σχετικές πολυάριθμες αναφορές (Soloway et al. 1982, Soloway et al. 1983, Bonar & Soloway 1985, Soloway 1986, Rogalski 1985, Dagdilelis 1986, Bonar & Soloway 1989, Soloway et al. 1989) είναι ενδεικτικές της σπουδαιότητας του θέματος.

Ορισμένες αντιλήψεις των μαθητών συνδεόμενες με τη λειτουργία και τη χρήση των βρόχων, αποτελούν ουσιαστικό ανασταλτικό παράγοντα στην πορεία τους (Dagdilelis 1989) και η υπέρβαση των αντιλήψεων αυτών είναι απαραίτητη για την πρόοδο του μαθητή.

2.4.5 Ταξινόμηση-αναζήτηση

Με ένα γενικό τρόπο, όπως αναφέρουν οι (Τζιμογιάννης Α. et al. 2019), οι μαθητές έχουν μια δυσκολία διάκρισης ανάμεσα στους δείκτες (pointers) και το περιεχόμενο της θέσης προς την οποία δείχνουν οι αντίστοιχοι δείκτες. Όσο πιο σύνθετες είναι οι εκφράσεις που χρησιμοποιούνται, τόσο πιο δυσνόητα είναι για τους αρχάριους

προγραμματιστές. Δυσκολίες επίσης φαίνεται να παρουσιάζει η κατανόηση πινάκων με διάσταση μεγαλύτερη του 1 (2,3,4 κ.λπ. διαστάσεις).

Όπως είναι γνωστό, οι αναζητήσεις στοιχείων και οι ταξινομήσεις στοιχείων σε πίνακα (ή και σε άλλες δομές δεδομένων) είναι από τα πιο σημαντικά αλγοριθμικά θέματα. Όσον αφορά στους αρχάριους, ή μη πεπειραμένους προγραμματιστές, αναφέρεται (Τζιμογιάννης Α. et al. 2019), ότι είναι προτιμότερο να ξεκινούν από αλγορίθμους διαισθητικά πιο «εύκολους», ή αλγορίθμους που στηρίζονται σε πρότερες γνώσεις τους. Έτσι για παράδειγμα, ο αλγόριθμος επιλογής (εντοπισμός του ελάχιστου στοιχείου και τοποθέτησή του στην κατάλληλη θέση) είναι πιο κατανοητός. Επίσης, πάντα σύμφωνα με σχετικές έρευνες, φαίνεται ότι ο αλγόριθμος παρεμβολής (insertion sort) ή εισαγωγής (που βασίζεται στην παρεμβολή ενός νέου στοιχείου σε ένα ήδη ταξινομημένο πίνακα, οριακά με 1 στοιχείο) θεωρείται σχετικά κατανοητός. Αντίθετα, οι αλγόριθμοι ευθείας ανταλλαγής στοιχείων είτε «φυσαλίδας» είναι πολύ λιγότερο συμβατοί με τις αυθόρμητες αντιλήψεις και τη διαίσθηση των μαθητών, παρόλο που είναι εκείνοι που διδάσκονται στο σχολείο.

2.4.6 Υποπρογράμματα

Τα υποπρογράμματα, είναι από τα πλέον δύσκολα, στην κατανόησή τους, θέματα που διδάσκονται στο σχολείο (Τζιμογιάννης Α. et al. 2019). Κατ' αρχάς, η ίδια η αναγκαιότητα χρήσης υποπρογραμμάτων προβάλλει, όταν ο προγραμματιστής (η μαθήτρια ή ο μαθητής) εμπλακεί με προβλήματα αρκετά πολύπλοκα, ώστε η χρήση υποπρογραμμάτων να είναι πραγματικά απαραίτητη, αλλιώς η χρησιμότητά τους παραμένει αμφίβολη. Αλλά ακόμη και τότε, η διάκριση ανάμεσα σε διαδικασίες και συναρτήσεις, τα διάφορα είδη «περάσματος των παραμέτρων», η διάκριση ανάμεσα σε τοπικές και καθολικές μεταβλητές και η έννοια της εμβέλειας μιας μεταβλητής, απαιτούν αρκετή διανοητική προσπάθεια για την κατανόησή τους. Συνεπώς οι μαθητές δυσκολεύονται να κατανοήσουν την κλήση υποπρογραμμάτων από το κυρίως πρόγραμμα (π.χ. από ποιο σημείο θα γίνει η κλήση; Τι συμβαίνει όταν καλείται το υποπρόγραμμα με διαφορετικές τιμές; κ.ά.).

Πέραν των παραπάνω δυσκολιών, αναφέρονται (Τζιμογιάννης Α. et al. 2019), επίσης δυσκολίες που παρουσιάζονται στην ορθή σύνταξη των σχετικών τμημάτων κώδικα. Φαίνεται, εξάλλου, ότι οι μαθητές έχουν γενικότερα μια ελλιπή κατανόηση του τρόπου με τον οποίο το σύστημα διαχειρίζεται την εσωτερική μνήμη που απαιτείται για τις

μεταβλητές και τις παραμέτρους, επομένως δυσκολεύονται να πραγματοποιούν τα θέματα των υποπρογραμμάτων. Προφανώς η ορθολογική επιλογή εκείνων των τμημάτων ενός προγράμματος, που οφείλουν να γίνουν υποπρογράμματα, είναι μια διαδικασία που απαιτεί εμπειρία από τον προγραμματιστή και διανοητική προσπάθεια. Η οπτική αναπαράσταση και η προσομοίωση της βηματικής εκτέλεσης ενός προγράμματος με υποπρογράμματα μέχρι στιγμής, αποτελούν εργαλεία για την υποστήριξη της διδασκαλίας.

Στις επόμενες ενότητες, αφού πρωτίστως παρουσιαστούν τα προγραμματιστικά περιβάλλοντα που χρησιμοποιούν τον ήχο (διέγερση ακουστικής αντίληψης) για την κατανόηση της υπολογιστικής σκέψης, θα μελετηθεί πώς ο ήχος μπορεί να συνεπικουρεί της οπτικής αναπαράστασης, ώστε να αποτελέσει ένα συμπληρωματικό εργαλείο για τη διδασκαλία του προγραμματισμού.

2.5 Περιβάλλοντα προγραμματισμού που χρησιμοποιούν ήχους

Έχουν αναπτυχθεί εφαρμογές που χρησιμοποιούν ήχο για την ανάπτυξη προγραμματιστικών περιβαλλόντων όπως το Scratch, το Tynker, το App Inventor, το Sonic Pi και το Max / MSP, για την κατανόηση της υπολογιστικής σκέψης όπως το Tune Pad και για την προσομοίωση ενεργειών όπως το Peep, τα οποία παρουσιάζονται στη συνέχεια. Στον Πίνακα 2 παρουσιάζονται όσα από αυτά χρησιμοποιούνται στην εκπαίδευση.

Πίνακας 2 Πίνακας εκπαιδευτικών περιβαλλόντων προγραμματισμού στα οποία χρησιμοποιείται Ήχος

Εκπαιδευτικό Περιβάλλον	Βαθμίδα Εκπαίδευσης	Χώρα
Scratch	Δημοτικό/Γυμνάσιο	Εξωτερικό/Ελλάδα
Tynker	Γυμνάσιο	Εξωτερικό
App Inventor	Γυμνάσιο/ Λύκειο	Εξωτερικό/Ελλάδα
Sonic Pi	Δημοτικό/Γυμνάσιο	Εξωτερικό/Ελλάδα
Tune Pad	Λύκειο/Τριτοβάθμια	Εξωτερικό
Max/MSP	Τριτοβάθμια	Εξωτερικό/Ελλάδα

Ένα εκπαιδευτικό εργαλείο προγραμματισμού, που χρησιμοποιεί και ακουστικά σήματα, είναι το **Scratch**. Το Scratch χρησιμοποιείται στην εκπαίδευση και την

ψυχαγωγία και είναι κατάλληλο για μαθητές σε ηλικία έναρξης οκτώ ετών ή μεγαλύτερης ηλικίας (Scratch Wiki 2015). Είναι ένα πολύ δημοφιλές προγραμματιστικό περιβάλλον για μια εισαγωγή στον προγραμματισμό. Όταν οι άνθρωποι μαθαίνουν να προγραμματίζουν στο Scratch μαθαίνουν σημαντικές στρατηγικές για την επίλυση προβλημάτων και τον σχεδιασμό έργων (Scratch MIT).

Οι μαθητές μπορούν εύκολα να δημιουργήσουν διαδραστικές ιστορίες, κινούμενα σχέδια, παιχνίδια στον υπολογιστή, μουσική και ψηφιακή τέχνη. Οι αισθητήρες μπορούν επίσης να χρησιμοποιηθούν με την πλακέτα pico, ένα κομμάτι υλικού, το οποίο επιτρέπει στα προγράμματα του Scratch να αλληλεπιδράσουν με τον έξω κόσμο. Το σύστημα προσφέρει υποστήριξη για μουσικά μπλοκ που ελέγχουν τα σετ βρόχων, την αναπαραγωγή κλπ. (Kirn 2018). Ευρήματα μελετών αποκαλύπτουν ότι τα εκπαιδευτικά παιχνίδια προσθέτουν το στοιχείο διασκέδασης στη μάθηση και οι μαθητές αξιολόγησαν το παιχνίδι, ως έναν αποτελεσματικό τρόπο για να μάθουν τον προγραμματισμό. Οι μαθητές θα μπορούσαν εύκολα να συσχετίζουν στοιχεία παιχνιδιού με δύσκολες προγραμματικές δομές και να ενισχύσουν τις δεξιότητες προγραμματισμού τους (Mathrani, Christian & Ponder-Sutton, 2016).

Ως μοντέλο προγραμματιστικό, εντάσσεται στην οικογένεια των logo-like περιβαλλόντων: είναι δηλαδή ένας μικρόκοσμος στον οποίο κινείται και «ζει» μια οντότητα (στο Scratch εξ' ορισμού μια γάτα, έναντι της χελώνας στη Logo), η οποία υπακούει στις οδηγίες/εντολές του χρήστη/προγραμματιστή. Είναι ένας ανοιχτός μικρόκοσμος, ο οποίος επιτρέπει την ανάπτυξη πολλών projects από διάφορα επιστημονικά πεδία και ευνοεί επίσης τη διαθεματική προσέγγιση. Το Scratch ακολουθεί την παράδοση της Logo, αλλά διαφέρει από αυτήν. Η μεγάλη διαφορά από τη Logo συνίσταται στο ότι οι εντολές είναι υπό τη μορφή εικόνων-«αντικειμένων» οι οποίες μετακινούνται, αντί να συντάσσονται. Ως εκ τούτου αποκλείονται τα «συντακτικά» λάθη και η δομή του προγράμματος είναι ίσως πιο κατανοητή. Το περιβάλλον μέσα στο οποίο κινείται ο «ήρωας» είναι πολύ πιο παραστατικό και εύκολα διαχειρίσιμο, από το λιτό, μινιμαλιστικό περιβάλλον της «τυπικής» Logo. Οι παράγοντες αυτοί διευκολύνουν τον προγραμματισμό για τους αρχαίους και επιπλέον αποτελούν, για τους νεαρούς μαθητές, πρόσθετο κίνητρο για να ασχοληθούν με τις δραστηριότητες και τα projects που αναπτύσσονται στο περιβάλλον του.

Το Scratch χρησιμοποιείται περισσότερο από μαθητές Γυμνασίου, βάσει του προγράμματος σπουδών.

Το **Tynker** είναι μια εκπαιδευτική πλατφόρμα προγραμματισμού που στοχεύει στη διδασκαλία των παιδιών για την κατασκευή παιχνιδιών και προγραμμάτων. Αντί να πληκτρολογούν τον πηγαίο κώδικα, σύρουν οπτικά μπλοκ κώδικα και τα μεταφέρουν όλα μαζί (Tynker, 2021). Ο οπτικός σχεδιασμός και οι αρχές του βασίζονται στο Scratch. Το Tynker βασίζεται σε HTML5 και JavaScript και μπορεί να χρησιμοποιηθεί στο πρόγραμμα περιήγησης χωρίς προσθήκες, σε tablet και smartphone. Αποτελεί ένα ελκυστικό πρόγραμμα σπουδών για σχολεία. Διευκολύνει τα παιδιά να μάθουν την κωδικοποίηση δημιουργώντας παιχνίδια όπως το Hour of Code.

Το περιβάλλον του **App Inventor** έχει πολλές ομοιότητες με το περιβάλλον του Scratch, με τη διαφορά ότι οι εφαρμογές που δημιουργούνται, τρέχουν σε έξυπνες φορητές συσκευές. Επιπλέον, ενώ το Scratch, το οποίο απευθύνεται κυρίως στους μικρούς μαθητές, περιλαμβάνει μόνο βασικές προγραμματιστικές λειτουργίες, το App Inventor στον αντίποδα, έχει πολλαπλά επίπεδα ενεργειών, προκειμένου να ικανοποιήσει τους χρήστες με διαφορετικά επίπεδα δεξιοτήτων.

Αναφορικά με την εκμάθηση προγραμματισμού ένας ισχυρός παράγοντας είναι το γνωστικό φορτίο το οποίο απαιτεί το κάθε προγραμματιστικό περιβάλλον από τον χρήστη. Ο Sweller (2010) αναφέρει ότι προκειμένου να μειώσουμε το ενδογενές νοητικό φορτίο για αρχάριους στον προγραμματισμό χρήστες, αρκεί να μειώσουμε την ποσότητα της πληροφορίας που απαιτείται να χρησιμοποιήσουν, προκειμένου να επιλύσουν το πρόβλημα. Αυτό μπορεί να επιτευχθεί με την απομόνωση προγραμματιστικών ενοτήτων, ώστε οι χρήστες να μην είναι υποχρεωμένοι να θυμούνται αρκετές σχετικές πληροφορίες ταυτόχρονα.

Προγραμματιστικά περιβάλλοντα τύπου drag & drop, όπως το AI και το Scratch, αντικαθιστούν τον προς συγγραφή κώδικα, με οπτικά αντικείμενα τα οποία επιλέγονται μέσω ενός μενού επιλογών, μειώνοντας το νοητικό φορτίο που απαιτείται για τη συγγραφή κώδικα και ταυτόχρονα βοηθώντας τους χρήστες να επικεντρωθούν στην επίλυση ενός προβλήματος (Brennan, 2009).

Επίσης οι Resnick et al., (2009) αναφέρουν ότι προγραμματιστικά περιβάλλοντα αυτού του τύπου, θεωρούνται εύκολα στην εκμάθησή τους για όλες τις ηλικίες, με

διαφορετικά εκπαιδευτικά υπόβαθρα και ενδιαφέροντα, καθώς επιτρέπουν στους χρήστες να πειραματίζονται με προγραμματιστικές δομές, απλά ενώνοντας κομμάτια κώδικα με παρόμοιο τρόπο, με κείνον που συνδέουν τουβλάκια τύπου Lego.

Σύμφωνα με τους ίδιους ερευνητές, η παραπάνω προσέγγιση είναι ιδανική για αρχάριους στον προγραμματισμό χρήστες, καθώς τους προσφέρεται η δυνατότητα να επικεντρωθούν στη δομή των λύσεων, παρά στη σύνταξη προγραμματιστικών εντολών. Με δεδομένη τη δημοφιλή και γενικευμένη παρουσία των έξυπνων φορητών συσκευών στον κοινό, το App Inventor φαίνεται να διαθέτει σημαντικές δυνατότητες για την προσέλκυση μιας ολόκληρης γενιάς μαθητών, φοιτητών και εκπαιδευτικών στην πληροφορική και στην υπολογιστική σκέψη (Hsu, Rice & Dawley, 2012). Άλλωστε όπως αναφέρει και ο Mitchel Resnick (2009), επικεφαλής της ομάδας που ανέπτυξε το Scratch, η τεχνολογική ευχέρεια των μαθητών δεν πρέπει να περιορίζεται στην απλή χρήση των τεχνολογικών εργαλείων, αφού αυτοί θα πρέπει να είναι ικανοί όχι μόνο να διαχειρίζονται το περιεχόμενο, αλλά και να το κατασκευάζουν.

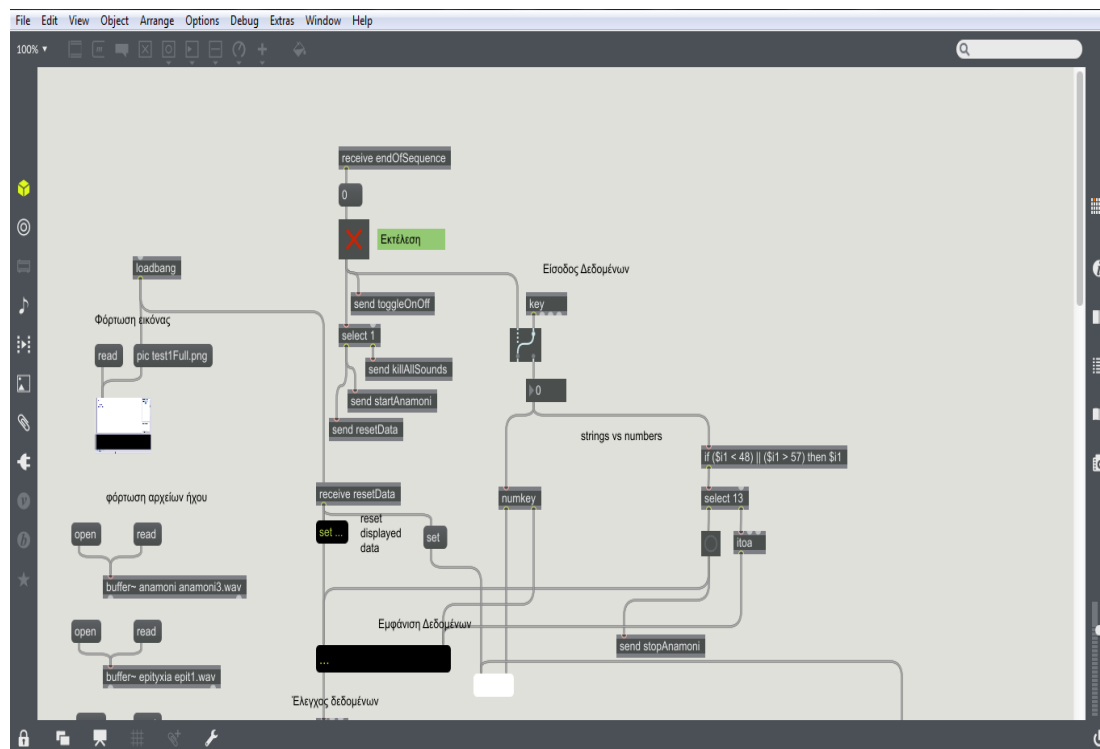
Το λογισμικό App Inventor με τις ανάλογες προσαρμογές από τον διδάσκοντα, ο οποίος λαμβάνει υπόψη του για παράδειγμα την πρότερη εμπειρία των μαθητών/τριών του, μπορεί να χρησιμοποιηθεί τόσο στο Γυμνάσιο όσο και στο Γενικό (Α΄ Λυκείου) και Επαγγελματικό Λύκειο. Επιπροσθέτως, παρότι δεν αναφέρεται ρητά στο πρόγραμμα σπουδών του Δημοτικού μπορεί ενδεχομένως και υπό ορισμένες προϋποθέσεις να αξιοποιηθεί και στις δύο μεγαλύτερες τάξεις, καθώς αναμένεται ότι οι μαθητές/τριες δεν θα συναντήσουν ιδιαίτερες δυσκολίες στην χρήση του, εφόσον στις μικρότερες τάξεις έχουν εξοικειωθεί με παρεμφερή προγραμματιστικά περιβάλλοντα, όπως το Scratch.

Το **Sonic Pi** είναι ένα εργαλείο δημιουργίας μουσικής με κώδικα (Sonic-pi, Aaron, Blackwell & Burnard 2016). Το περιβάλλον προγραμματισμού Sonic Pi είναι εύχρηστο και ο μαθητής μαθαίνει μουσική γράφοντας κώδικα και προγραμματισμό με την χρήση μουσικής. Το Sonic Pi περιλαμβάνει ένα πλήθος εντολών και δομών που επιτρέπουν στους μαθητές να αναπτύξουν μουσικές συνθέσεις, ενώ ταυτόχρονα κατανοούν αλγοριθμικές δομές. Σχεδιάστηκε αρχικά για να υποστηρίξει τόσο μαθήματα προγραμματισμού, όσο και τα μαθήματα μουσικής στα σχολεία. Αναπτύχθηκε από τον Sam Aaron στο εργαστήριο Computer University του Cambridge. Οι μαθητές εισάγονται στο εύχρηστο περιβάλλον προγραμματισμού **Sonic**

Pi, με το οποίο μπορούν αμέσως να δουν και να ακούσουν το αποτέλεσμα της προσπάθειάς τους. Με κίνητρο τη μουσική σύνθεση εισάγονται σε ολοένα και πιο σύνθετες έννοιες, που αποτελούν θεμέλιο για την ουσιαστική κατανόηση της Επιστήμης των Υπολογιστών. Οι μαθητές νοηματοδοτούν σύνθετες υπολογιστικές έννοιες, στο πλαίσιο της προσωπικής έκφρασης και δημιουργικότητας. Επίσης με το Soni Pi μπορεί να επιτευχθεί μια Εισαγωγή στον Ζωντανό Προγραμματισμό (live coding).

Ο Ζωντανός Προγραμματισμός αναφέρεται σε μια μέθοδο προγραμματισμού, που επιτρέπει τη συγγραφή κώδικα και την τροποποίηση ενός προγράμματος κατά τον χρόνο εκτέλεσής του (https://en.wikipedia.org/wiki/Live_coding). Ο Ζωντανός Προγραμματισμός αποτελεί αποτελεσματικό εργαλείο στη διδακτική του προγραμματισμού (Rubin 2013). Το Soni Pi είναι ευχάριστο και ενδείκνυται σε μαθητές Δημοτικού και Γυμνασίου.

Το **Max / MSP** είναι ένα περιβάλλον προγραμματισμού υψηλού επιπέδου, ως διαδραστικός χρήστης, που χρησιμοποιείται κυρίως από συνθέτες, μουσικούς, καλλιτέχνες. Χρησιμοποιείται για την κατασκευή διαδραστικών εφαρμογών μουσικής και πολυμέσων σε πραγματικό χρόνο και παρέχει άμεσο οπτικοακουστικό εφέ (Cirriani & Giri 2010). Διδάσκεται στην τριτοβάθμια εκπαίδευση. Με πάνω από τριάντα χρόνια ιστορίας, το πρόγραμμα Max είναι σπονδυλωτό, με τις περισσότερες ρουτίνες (διαδικασίες) να υπάρχουν ως κοινόχρηστες βιβλιοθήκες. Μια διεπαφή προγραμματισμού εφαρμογών (API) επιτρέπει την ανάπτυξη νέων ρουτινών από τρίτους, που ονομάζονται εξωτερικά αντικείμενα (objects). Έτσι, το Max έχει μια μεγάλη βάση χρηστών-προγραμματιστών, μη συνεργαζόμενους με την Cycling '74, οι οποίοι βελτιώνουν το λογισμικό με εμπορικές και μη εμπορικές επεκτάσεις στο πρόγραμμα. Εξαιτίας αυτού του σχεδιασμού επεκτασιμότητας, ο οποίος ταυτόχρονα αντιπροσωπεύει τόσο τη δομή του προγράμματος, όσο και τη γραφική διεπαφή χρήστη (GUI), το Max έχει περιγραφεί ως το lingua franca για την ανάπτυξη λογισμικού διαδραστικών μουσικών θεμάτων (Place & Lossius 2006).



Σχήμα 2. Το περιβάλλον προγραμματισμού Max / MSP

Το **TunePad** είναι μια tablet εφαρμογή σύνθεσης ήχου, που ελέγχεται από περιβάλλον προγραμματισμού και βασίζεται σε μπλοκ. Το TunePad έχει σχεδιαστεί για να εισαγάγει τους μαθητές στην υπολογιστική σκέψη και να τους προετοιμάσει για περιβάλλοντα κωδικοποίησης βάσει κειμένου. Αυτός ο σχεδιασμός προσελκύει ενεργά τους μαθητές, διότι τους δίνεται η δυνατότητα της επικοινωνίας, αφού τα μπλοκ προγραμματισμού ελέγχουν τους ήχους που παίζονται. Προσανατολίζει και δεσμεύει τους μαθητές προς την υπολογιστική σκέψη, οι οποίοι μπορεί να μην είχαν δοκιμαστεί άλλη φορά, δίνοντας την ευκαιρία σε περισσότερους ανθρώπους να εισέλθουν στον τομέα της επιστήμης των υπολογιστών (Gorson, Patel, Beheshti, Magerko & Horn 2017).

Το **Peep** είναι ένας «ηχητικός επόπτης» δικτύου (εποπτεύονται οι λειτουργίες του δικτύου με χρήση ήχου), που αντικαθιστά την οπτική παρακολούθηση με μια σειρά φυσικών ήχων (Gilfix & Couch). Κάθε τύπος ήχου αντιπροσωπεύει ένα συγκεκριμένο είδος συμβάντος του δικτύου. Αυτό το σύστημα συνδυάζει πληροφορίες της κατάστασης ενός δικτύου από πολλαπλές πηγές δεδομένων, αναμειγνύοντας τα ακουστικά σήματα σε μία μόνο ροή ήχου, σε πραγματικό χρόνο. Με τη χρήση του Peep, μπορεί κανείς εύκολα να ανιχνεύσει κοινά προβλήματα δικτύου, όπως υψηλό φορτίο, υπερβολική κίνηση και ανεπιθύμητα μηνύματα ηλεκτρονικού ταχυδρομείου,

συγκρίνοντας τους ήχους που παίζουν, με αυτούς ενός δικτύου που λειτουργεί κανονικά. Αυτό επιτρέπει στον διαχειριστή του συστήματος να επικεντρωθεί σε πιο σημαντικά πράγματα, ενώ παρακολουθεί το δίκτυο μέσω της περιφερειακής ακοής.

Από τις παραπάνω αναφορές μετά την οπτικοποίηση, το επόμενο βήμα προφανώς θα είναι μια διαδικασία μετάδοσης πληροφοριών μέσω ήχων με τη χρήση παραμέτρων που ανταποκρίνεται στο ανθρώπινο αυτί, όπως η ένταση και η συχνότητα (Cerf 2018).

Η ακοή παίζει θεμελιώδη ρόλο κατά την επεξεργασία πληροφοριών, καθώς είναι μια από τις κύριες πηγές πληροφοριών στον εγκέφαλό μας. Το αυτί επιλέγει τους γύρω ήχους, αλλά και εκείνους της δικής μας φωνής τους μετατρέπει σε ηλεκτρικά ερεθίσματα και τους μεταδίδει στον εγκέφαλο, ο οποίος τους επεξεργάζεται και τους αναλύει.

Ως αντίληψη ορίζεται μια από τις νοητικές λειτουργίες με την οποία ο άνθρωπος είτε με τις αισθήσεις είτε με τη λογική λαμβάνει και ερμηνεύει τα εξωτερικά ερεθίσματα. Είναι μια ερμηνευτική διαδικασία, κατά την οποία δέχεται επιρροή από τη φύση των αντιληπτικών ερεθισμάτων από τη φύση των προσδοκιών των ατόμων, τις προηγούμενες γνώσεις τους και εμπειρίες.

Η αντίληψη (perception) καλείται στην ψυχολογία μια σύνθετη και πολύπλοκη λειτουργία τού ψυχοφυσικού οργανισμού, που αυτόματα (μέσω δυναμικής αλληλοσυσχέτισης των μηνυμάτων της εμπειρίας) μέσα στα αρμόδια εγκεφαλικά κέντρα, αναγνωρίζει, ερμηνεύει άμεσα και αποδίδει νόημα, με τη βοήθεια της νόησης (συλλαμβάνοντας απόψεις, ιδιότητες, ποιότητες ή σχέσεις των αντικειμένων μεταξύ τους), της διανοητικής ανάπτυξης, του συναισθήματος (διάθεσης) και της προηγούμενης εμπειρίας (μάθησης) του ατόμου (el.wikipedia.org).

Η ακουστική αντίληψη μπορεί να οριστεί ως η ικανότητα να λαμβάνουμε και να ερμηνεύουμε τις πληροφορίες που φτάνουν στα αυτιά μας, χρησιμοποιώντας τα κύματα της ηχητικής συχνότητας, που μεταδίδονται μέσω του αέρα ή με άλλο τρόπο. Η ανάπτυξη των ακουστικών δεξιοτήτων αποτελεί το βασικό θεμέλιο πάνω στο οποίο ο μαθητής χτίζει πολλές από τις δεξιότητες ακρόασης και επικοινωνίας. Η εξέλιξη αυτή συμβαίνει συνήθως σε τέσσερα στάδια, την **Επίγνωση** (Ευαισθητοποίηση των ήχων, αναγνώριση του ήχου), την **Προσοχή** (Ο μαθητής επικεντρώνεται ενεργά στον ήχο τον

οποίο ακούει), την **Ανταπόκριση** (Ανταπόκριση με οποιονδήποτε τρόπο στον ήχο που άκουσε ο μαθητής) και την **Κατανόηση** (Σύνδεση ήχων με νόημα).

Η ακουστική αναπαράσταση (Auditory display) συνδέει την ψυχοακουστική με τη γνωστική λειτουργία ανάλογα με τα χαρακτηριστικά του ήχου. Οι πιο σημαντικές πτυχές στον σχεδιασμό της, αφορούν το αν ο ακροατής μπορεί να ακούσει τον ήχο και να διακρίνει τις συγκεκριμένες αλλαγές συγκεκριμένων παραμέτρων. Βασικά πρέπει να γνωρίζουμε αρκετά, στο πώς ακούνε και σκέφτονται οι μαθητές (Walker & Kramer, 2004).

Η ποιότητα του τόνου (Timbre) είναι ένας όρος, τόσο στην ψυχοακουστική, όσο και στην ακουστική αναπαράσταση, που συχνά χρησιμοποιείται για να προσδιορίζει τα χαρακτηριστικά ήχου. Η δυνατότητα διάκρισης ήχων διαφορετικών timbres ήταν σημαντική στη χαρτογράφηση δεδομένων σε ήχους. Το πλάτος (pitch) είναι η πιο συχνά χρησιμοποιούμενη διάσταση ακουστικής αναπαράστασης. Αυτό συμβαίνει επειδή είναι εύκολη η διαχείριση και γενικά οι αλλαγές στο πλάτος στην ομιλία γίνονται εύκολα αντιληπτές. Η ψυχοακουστική έχει καθορίσει μια δυναμική λειτουργία που περιγράφει τον τρόπο που οι ακροατές αντιλαμβάνονται τις αλλαγές στην ένταση του ήχου. Αυτό μπορεί να επιτρέψει σε έναν σχεδιαστή να δημιουργήσει γραμμικά βήματα στην ένταση, τα οποία μπορεί να είναι πιο κατάλληλα για να αντιπροσωπεύουν ορισμένους τύπους δεδομένων και να δημιουργούν μια αποτελεσματική και μη επιζήμια ακουστική αναπαράσταση, για οποιαδήποτε ηχητική ειδοποίηση (Fastl 2006).

Αυτή η έρευνα άπτεται του επιστημονικού κλάδου της ακουστικής αναπαράστασης στη διδασκαλία του προγραμματισμού. Θα μελετηθεί και θα αξιολογηθεί η αντιστοίχιση των εντολών/ενεργειών ενός προγράμματος με ηχητικές ειδοποιήσεις. Ήχοι που μέσω διέγερσης της ακουστικής αντίληψης του κάθε μαθητή θα ανιχνεύονται, διακρίνονται, αναγνωρίζονται, κατανοούνται με τελικό στόχο την αντίστοιχη κατανόηση των εντολών/ενεργειών προγραμματισμού.

3. Πρόταση μεθοδολογίας διδασκαλίας του προγραμματισμού χρησιμοποιώντας ηχητικές ειδοποιήσεις

Στο 2^ο κεφάλαιο εξετάσαμε πως εξελίχθηκε η διδασκαλία του προγραμματισμού, ποια προγραμματιστικά περιβάλλοντα χρησιμοποιούνται στη εκπαίδευση, ποιες οι μεθοδολογίες διδασκαλίας, ποια τα προβλήματα στην εκμάθησή του.

Έχουν διεξαχθεί ερευνητικά σενάρια διδασκαλίας για την αξιολόγηση της αντιληπτικής και κριτικής ικανότητας των μαθητών σε βασικές αλγοριθμικές δομές, όπως αναφέρει η έρευνα των (Hsu & Hwang 2017). Από την άλλη πλευρά σύμφωνα με την αναφορά (Milne & Rowe 2002), οι μαθητές στα γυμνάσια και λύκεια μαθαίνουν μόνο βασικά στοιχεία του προγραμματισμού λόγω των αναποτελεσματικών και, σε ορισμένες περιπτώσεις, ξεπερασμένων μεθόδων διδασκαλίας, καθώς και της απουσίας διασύνδεσης μεταξύ εκπαίδευσης και βιομηχανίας.

Θεωρήσαμε σκόπιμο και σημαντικό να πειραματιστούν οι μαθητές με ηχητικά περιβάλλοντα, ώστε να διεγείρουμε την προσοχή τους, την αντίληψή τους και με νέες μεθόδους ελκυστικές με ερεθίσματα ακουστικά και οπτικά και όχι να μείνουμε στάσιμοι σε μια λογική μαθησιακή διδασκαλία, αλλά με πολλά ερεθίσματα, στοχεύουμε να επιτύχουμε την καλύτερη κατανόηση.

Σε μια προσπάθεια βελτίωσης της εκμάθησης και κατανόησης του προγραμματισμού, η παρούσα διατριβή ασχολείται με μια μεθοδολογία για τη διδασκαλία του προγραμματισμού μέσω ηχητικών ειδοποιήσεων, που διεγείρουν την ψυχοακουστική αντίληψη του μαθητή, μια νέα διδακτική προσέγγιση με σκοπό να κεντρίσει το ενδιαφέρον των μαθητών.

Η αντίληψη της ποιότητας του ήχου εξαρτάται, μεταξύ άλλων, από την ένταση, τη χροιά και την ευκρίνεια (Fastl 2006). Η χρήση φυσιολογικών μέτρων ευαίσθητων στην προσοχή και τη διέγερση, σε συνδυασμό με τα συμπεριφορικά και υποκειμενικά κριτήρια, οδήγησε στον σχεδιασμό των ακουστικών προειδοποιήσεων που παράγουν την αίσθηση της διαφορετικότητας των εντολών προγραμματισμού (Burt et al. 1995, Fastl 2006). Γνωρίζοντας ότι η ακουστική και οπτική μνήμη αποτελούν το 90% της αισθητήριας μνήμης, μπορεί να υποτεθεί ότι οι μαθητές που χρησιμοποιούν τόσο την

οπτική όσο και την ακουστική τους αντίληψη, θα κατανοήσουν καλύτερα τις δομές προγραμματισμού (Fastl 2006, Yarbrough 2017). Ο στόχος είναι να βοηθηθούν οι μαθητές να κατανοήσουν τις διαφορετικές αλγοριθμικές διαδικασίες, όπως σχεσιακές και αριθμητικές λειτουργίες, επιτυχημένη και ανεπιτυχή εκτέλεση κώδικα, ανάθεση τιμής, φόρτωση δεδομένων κλπ., μέσω αμφίδρομης (οπτικής και ακουστικής) ανάδρασης. Υποθέτουμε ότι μέσω των οπτικών και φωνητικών αλληλεπιδράσεων οι μαθητές θα είναι σε θέση να κατανοήσουν αποτελεσματικότερα τις διδαχθείσες δομές προγραμματισμού.

3.1 Χρήση λογισμικών στην παρούσα διατριβή

Προς αυτή την κατεύθυνση το έργο (μεθοδολογία πειραματισμού σε ομάδες μαθητών) που παρουσιάζεται στην παρούσα διατριβή χρησιμοποιεί το περιβάλλον MAX / MSP με σκοπό τον σχεδιασμό σεναρίων διδασκαλίας για την παραγωγή ηχητικής ειδοποίησης και δημιουργεί προγράμματα σε Max / MSP για την προσομοίωση εντολών και ενεργειών με ήχο. Κάθε πρόγραμμα Max / MSP σχεδιάστηκε έτσι, ώστε να υπάρχει ταυτόσημη διεπαφή του χρήστη με το περιβάλλον της Γλωσσομάθειας, η οποία διδάσκεται, όπως προαναφέρθηκε, στη Δευτεροβάθμια Εκπαίδευση. Σε αυτή τη διατριβή η Γλωσσομάθεια αποτελεί τη βάση για την αξιολόγηση της επίδρασης των ακουστικών σημάτων στην κατανόηση του προγραμματισμού στον υπολογιστή η δε, ιδέα να δοκιμαστεί η ακουστική ανατροφοδότηση, χρησιμοποιώντας τους ήχους στο σύστημα που θα παρουσιαστεί, προέκυψε από το περιβάλλον Peer, που αποτέλεσε έναυσμα για τη χρήση ήχου, ως συμπληρωματικού εργαλείου στην καλύτερη κατανόηση του προγραμματισμού, καθώς ο κύριος στόχος μας ήταν να επιτύχουμε μια καλύτερη κατανόηση της διδασκαλίας του προγραμματισμού με τη χρήση ήχων.

3.2 Επιλογή ήχων

Ένα κρίσιμο συστατικό στοιχείο στον σχεδιασμό ενός εκπαιδευτικού προγραμματιστικού περιβάλλοντος είναι η επιλογή των χρησιμοποιούμενων ηχητικών ερεθισμάτων, τα οποία για να είναι ευνοϊκά στη μάθηση των μαθητών, πρέπει να αντικατοπτρίζουν με σαφή και συνοπτικό τρόπο την αλγοριθμική δράση (εντολή-λειτουργία) που αντιστοιχούν (Burt et al. 1995). Επιπλέον, θα πρέπει επίσης να αντικατοπτρίζουν την αισθητική της ομάδας στόχου, η οποία, στην περίπτωσή μας, αποτελείται από μαθητές Λυκείου.

Ως εκ τούτου, διεξήχθη μια προκαταρκτική μελέτη με στόχο την επιλογή των καταλληλότερων ηχητικών ερεθισμάτων, λαμβάνοντας υπόψη τα προαναφερθέντα κριτήρια (Genuit 2004, Burt et al. 1995, Fastl 2006). Οι μαθητές κλήθηκαν να επιλέξουν ήχους για διαφορετικές λειτουργίες ενός προγράμματος. Επιλέχθηκαν δύο διαφορετικές κατηγορίες ήχων, φυσικοί ήχοι (από το φυσικό περιβάλλον) και τεχνητοί ήχοι, για να αντιπροσωπεύουν πέντε (5) διαδικασίες - λειτουργίες :

- α) εισαγωγή δεδομένων από το πληκτρολόγιο
- β) επιτυχής ανάθεση δεδομένων σε μια μεταβλητή
- γ) σφάλμα
- δ) αριθμητική πράξη και
- ε) σχεσιακή (συγκριτική) πράξη.

Για κάθε διαδικασία, επιλέχτηκε ένα ζευγάρι ηχητικών ερεθισμάτων (ένα φυσικό και ένα τεχνητό), έτσι ώστε οι ήχοι να έχουν κοινά ακουστικά χαρακτηριστικά, όσον αφορά στην **ένταση** (πόσο δυνατά ή χαμηλά ακούγεται ένας ήχος), τη **διάρκεια** (εάν ένας ήχος είναι σύντομος ή εάν διαρκεί πολύ), το **τονικό ύψος** (αναφέρεται στην "οξύτητα" του ήχου, το πόσο "λεπτός" είναι) και τη **χροιά** ή αλλιώς το **ηχόχρωμα**. Οι συμμετέχοντες στη μελέτη κλήθηκαν στη συνέχεια να επιλέξουν τις καλύτερες και κατάλληλες ειδοποιήσεις ήχου για τις συγκεκριμένες διαδικασίες-ενέργειες του προγραμματισμού, κατά την άποψή τους.

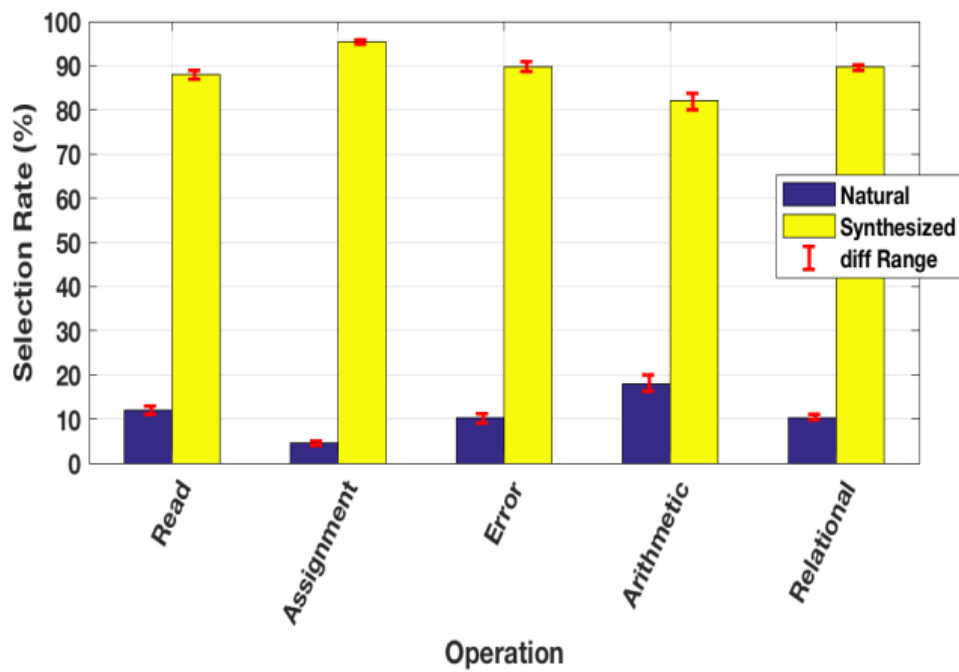
3.3 Πρωτόκολλο πειραμάτων α΄ φάσης (καταγραφή της διαδικασίας επιλογής ήχων – φυσικός ή τεχνητός ήχος)

Εκατόν σαράντα έξι (146) μαθητές λυκείου, εξήντα επτά (67) αγόρια, ηλικίας 16 έως 17 ετών, συμμετείχαν σε αυτήν την προκαταρκτική μελέτη. Οι συμμετέχοντες χωρίστηκαν σε οχτώ (8) ομάδες. Στις ομάδες παρουσιάστηκε, στο περιβάλλον Max / MSP, ένα ζεύγος ερεθισμάτων (ειδοποιήσεις ήχου) για καθεμία από τις πέντε (5) διαδικασίες κωδικοποίησης και ζητήθηκε στους μαθητές να επιλέξουν την προειδοποίηση που βάραινε καλύτερα σε κάθε διαδικασία σε μια εργασία 2AFC χωρίς επαναλήψεις. Κάθε ομάδα συμμετείχε στη δοκιμασία μόνο δύο (2) φορές, μία φορά στην αρχή και μία δεύτερη φορά στο τέλος μιας διδακτικής ώρας, για να αξιολογηθεί η απόκριση της επανάληψης του πειράματος. Η κατά προσέγγιση διάρκεια κάθε

δοκιμής ήταν πέντε (5) λεπτά. Μεταξύ των ομάδων, τόσο η σειρά των ηχητικών ερεθισμάτων, όσο και η δοκιμή των διεργασιών ήταν πλήρως τυχαία, αλλά παρέμεινε η ίδια εντός των ομάδων, για πρακτικούς λόγους.

3.4 Αποτελέσματα της προκαταρκτικής μελέτης

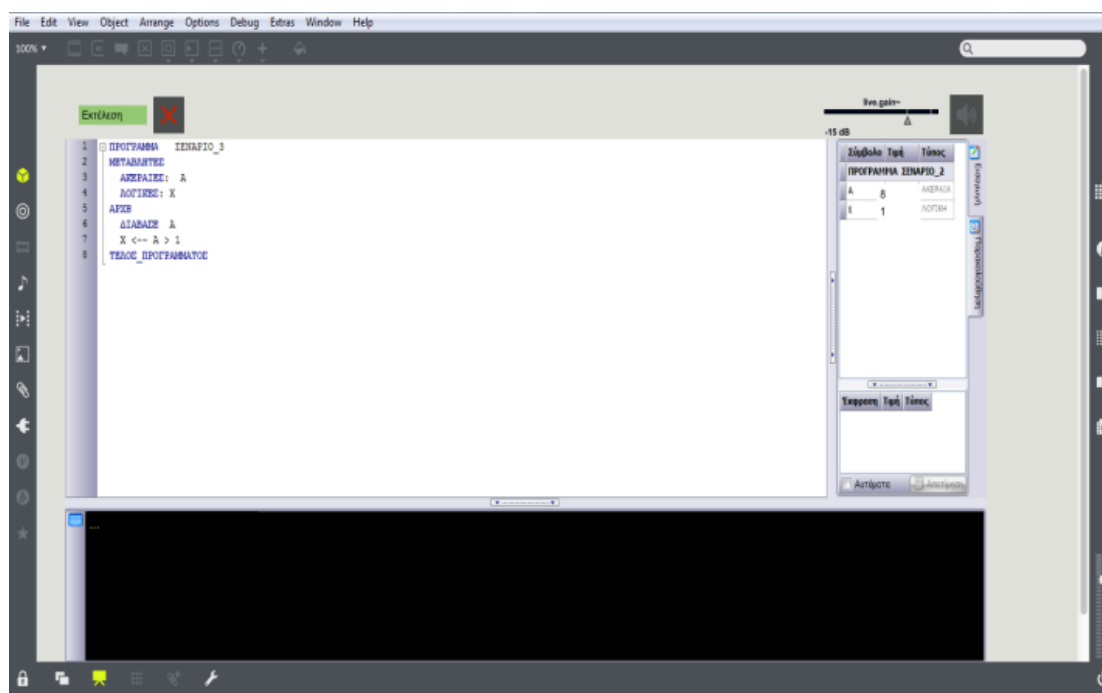
Τα αποτελέσματα της επιλογής του τύπου του ηχητικού ερεθίσματος της προκαταρκτικής μελέτης συνοψίζονται στο Σχήμα 3. Όπως φαίνεται, η μεγάλη πλειονότητα των μαθητών, που κυμαίνεται μεταξύ (82% και 96%), προτιμούν τους τεχνητούς ήχους ειδοποίησης για την προσομοίωση λειτουργιών σε ένα προγραμματιστικό περιβάλλον, από ότι τους φυσικούς ήχους. Δεν παρατηρήθηκαν σημαντικές αποκλίσεις, όσον αφορά στη σειρά των ηχητικών σημάτων ή της διαδικασίας παρουσίασης. Η προτίμηση των μαθητών παρέμεινε σχεδόν αμετάβλητη στις επαναλήψεις των δύο δοκιμών (βλ. Σχήμα 3, ράβδοι σφαλμάτων). Οι αποκλίσεις ήταν μικρότερες από 3% σε όλες τις διαδικασίες των δοκιμών. Αυτά τα αποτελέσματα υπογραμμίζουν, ότι οι προειδοποιήσεις ήχου προτιμώνται σε ένα εκπαιδευτικό λογισμικό προγραμματισμού. Οι ράβδοι αντιστοιχούν στον βαθμό επιλογής κάθε ήχου αντίστοιχου με κάθε διαδικασία, που αναγράφεται κατά μέσο όρο κατά τις επαναλαμβανόμενες δοκιμές. Οι μεταβολές μεταξύ των επαναλήψεων επισημαίνονται με ράβδους σφάλματος.



Σχήμα 3. Επισκόπηση αποτελεσμάτων των προκαταρκτικών δοκιμών.

4. Πειράματα β' φάσης - Μελέτη αξιολόγησης

Μετά την προκαταρκτική μελέτη πραγματοποιήθηκε ένα δεύτερο πείραμα για την αξιολόγηση της αποτελεσματικότητας των ηχητικών προειδοποιήσεων ως συμπληρωματικών εργαλείων για τη διδασκαλία του προγραμματισμού σε ηλεκτρονικούς υπολογιστές. Οι μαθητές συμμετείχαν σε έξι (6) σενάρια διδασκαλίας. Πενήντα τρεις (53) μαθητές/μαθήτριες Λυκείου, είκοσι τέσσερα (24) αγόρια, οι οποίοι προηγουμένως συμμετείχαν στην προκαταρκτική μελέτη, προσφέρθηκαν εθελοντικά να συμμετάσχουν στα τρία πρώτα σενάρια και ογδόντα (80) μαθητές/μαθήτριες, τριάντα οχτώ (38) αγόρια, στα επόμενα τρία σενάρια. Όλοι τους, μέχρι να πραγματοποιηθούν τα συγκεκριμένα πειράματα, παρακολουθούσαν το μάθημα προγραμματισμού στο σχολείο χρησιμοποιώντας το περιβάλλον Γλωσσομάθεια.



Σχήμα 4. Περιβάλλον Max / MSP - Εκτέλεση προγράμματος με ήχο (2ο πείραμα)

4.1 Πρωτόκολλο πειραμάτων β' φάσης

Στους συμμετέχοντες παρουσιάστηκαν έξι (6) σενάρια προγραμματισμού έτοιμα προς εκτέλεση, τόσο σε Γλωσσομάθεια, όσο και σε Max/ MSP. Η ενημερωμένη έκδοση κώδικα Max / MSP σχεδιάστηκε έτσι, ώστε να υπάρχει ταυτόσημη διεπαφή του

χρήστη με τη Γλωσσομάθεια. Η μόνη διαφορά ήταν ότι συμπληρώθηκε η οπτική ανατροφοδότηση με ειδοποιήσεις ήχου. Τα ηχητικά ερεθίσματα που χρησιμοποιήθηκαν είναι οι ειδοποιήσεις που επιλέχθηκαν μεταξύ των πλέον προτιμώμενων ήχων της προαναφερθείσας προκαταρκτικής μελέτης.

Οι συμμετέχοντες είχαν τη δυνατότητα να αλληλοεπιδρούν και με τα δύο περιβάλλοντα και στη συνέχεια τους ζητήθηκε να συμπληρώσουν ένα ερωτηματολόγιο, που αξιολογεί την επίδραση της ακουστικής ανάδρασης στην κατανόηση της λειτουργικότητας του κώδικα. Για κάθε διδακτικό σενάριο, οι αξιολογήσεις χρηστών συλλέχθηκαν σε κλίμακα 5AFC Linkert με τις ακόλουθες επιλογές: καμία επίδραση, μικρή επίδραση, μέτρια επίδραση, μεγάλη επίδραση, πολύ μεγάλη επίδραση. Το ερωτηματολόγιο ολοκληρώθηκε με μια ενότητα "γενικών παρατηρήσεων", όπου οι συμμετέχοντες μπορούσαν να μοιραστούν τις σκέψεις τους και τα σχόλιά τους, σχετικά με το περιβάλλον του συστήματος που πειραματίστηκαν.

4.2 Σενάρια διδασκαλίας

Τα επόμενα έξι (6) διδακτικά σενάρια, που περιγράφονται στα παρακάτω υποκεφάλαια πραγματοποιήθηκαν στη δεύτερη μελέτη.

Τα διδακτικά σενάρια αποτελούν συγκεκριμένα παραδείγματα προγραμμάτων, ένα παράδειγμα σε κάθε υποκεφάλαιο, που κάνουν χρήση αλγοριθμικών δομών, όπως δομή ακολουθίας, επιλογής, επανάληψης και λειτουργίες, όπως είσοδο, έξοδο, εκτέλεση πράξεων, έλεγχο συμβατότητας τύπου δεδομένων.

Σκοπός αυτών των σεναρίων διδασκαλίας είναι η καλύτερη κατανόηση της διαφοράς μεταξύ μιας αριθμητικής πράξης και μιας λογικής πράξης ή σχέσης σύγκρισης, της ισχύος μιας συνθήκης (αληθής ή ψευδής), των τύπων δεδομένων και των διαφορών των επαναληπτικών αλγοριθμικών δομών (πότε πραγματοποιείται μια επανάληψη, πότε τερματίζεται μια επαναληπτική διαδικασία), με τη χρήση ήχου.

4.2.1 Σενάριο 1: Εισαγωγή δεδομένων

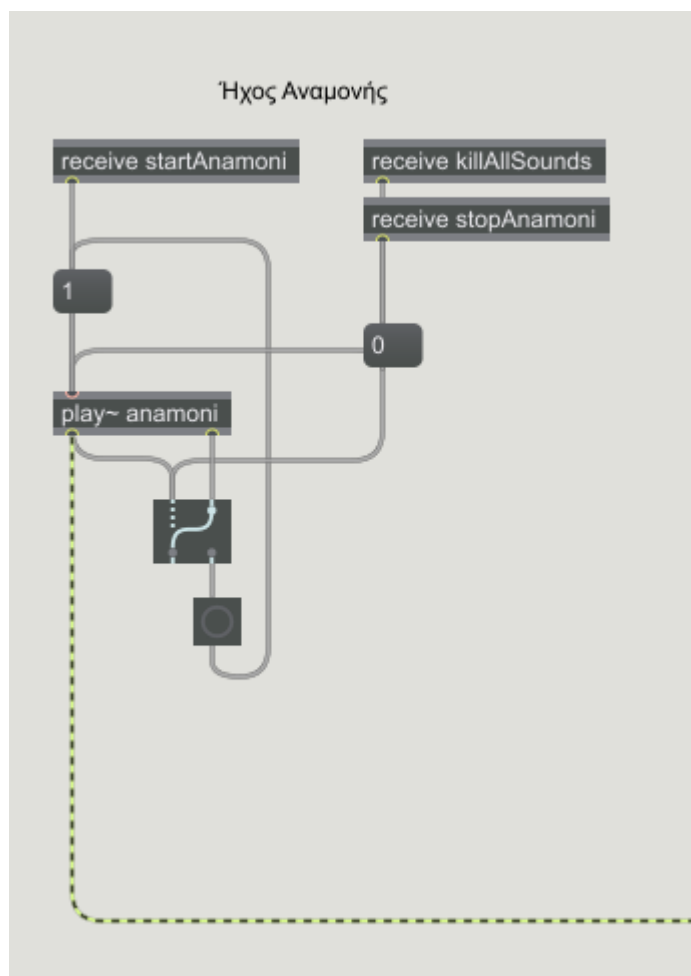
Ο κώδικας στο πρώτο σενάριο προγράμματος (Σχήμα 5α) εκτελεί την εισαγωγή μίας αριθμητικής τιμής σε μια προκαθορισμένη μεταβλητή, ακέραιου τύπου, που δίδεται από το πληκτρολόγιο με την εντολή εισόδου read (διάβασε).


```
! Data entry!  
Program Scenario_1  
Variables  
    Integer: A  
Begin  
    Read A  
End
```

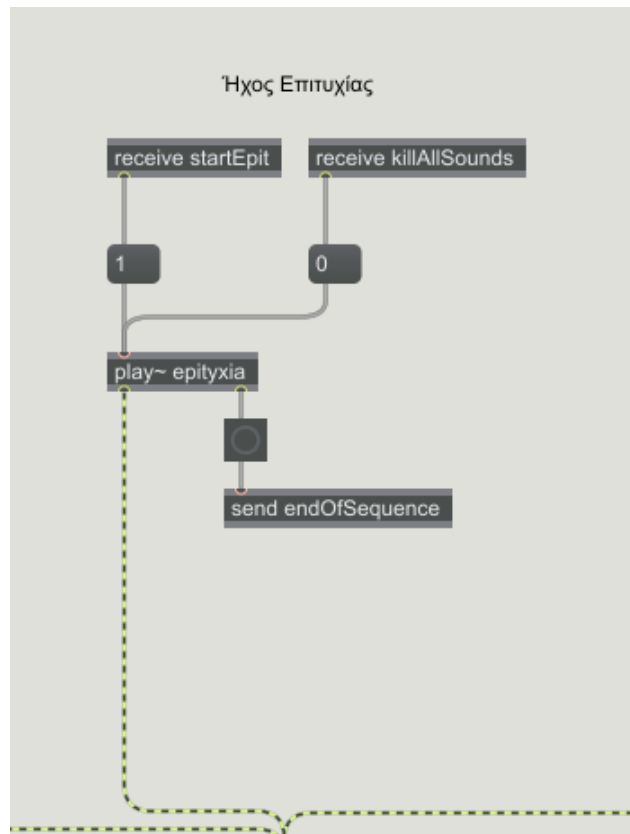
Σχήμα 5α. Καταχώριση δεδομένων

Όταν το σενάριο αυτό αξιολογείται στο περιβάλλον προγραμματισμού Γλωσσομάθεια, ο χρόνος αναμονής για την είσοδο χρήστη αναπαρίσταται με ένα δρομέα που αναβοσβήνει στην οθόνη του υπολογιστή. Στο περιβάλλον Max / MSP, εκτός από τον δρομέα που αναβοσβήνει, οι χρήστες ακούν μια ειδοποίηση ήχου που τους ενημερώνει για μια εκκρεμούσα ενέργεια. Εάν η είσοδος του χρήστη αποτελεί αριθμητικό δεδομένο, το περιβάλλον Γλωσσομάθεια εκτυπώνει την εντολή στην οθόνη και τελειώνει ο κώδικας, ενώ το περιβάλλον δοκιμής Max / MSP συμπληρώνει την οπτικοποίηση στη Γλωσσομάθεια, με μια ειδοποίηση ήχου που υποδεικνύει την επιτυχή ανάθεση. Αν η είσοδος χρήστη δεν είναι αριθμητική, το περιβάλλον της Γλωσσομάθειας εκτυπώνει ένα μήνυμα σφάλματος στην οθόνη, ενώ το περιβάλλον δοκιμής Max / MSP παράγει μια ειδοποίηση ήχου, επισημαίνοντας επιπλέον μια εσφαλμένη ενέργεια. Κάθε προειδοποίηση ήχου επιλέχθηκε στην προκαταρκτική μελέτη.

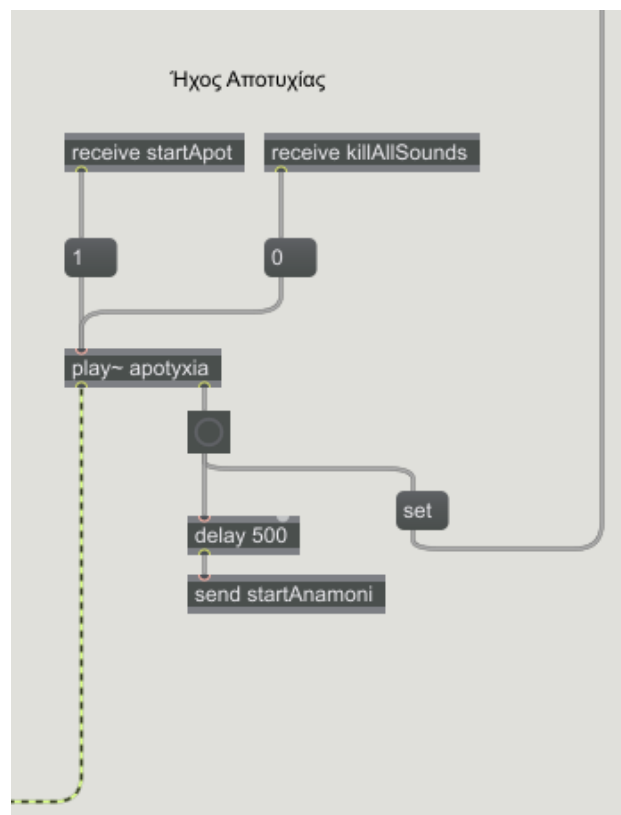
Στα παρακάτω σχήματα 5β, 5γ, 5δ, 5ε αποτυπώνονται αντίστοιχα οι κώδικες Max/MSP για τους ήχους αναμονής, επιτυχίας, αποτυχίας και ενεργοποίησης εκάστοτε ηχητικής ειδοποίησης.



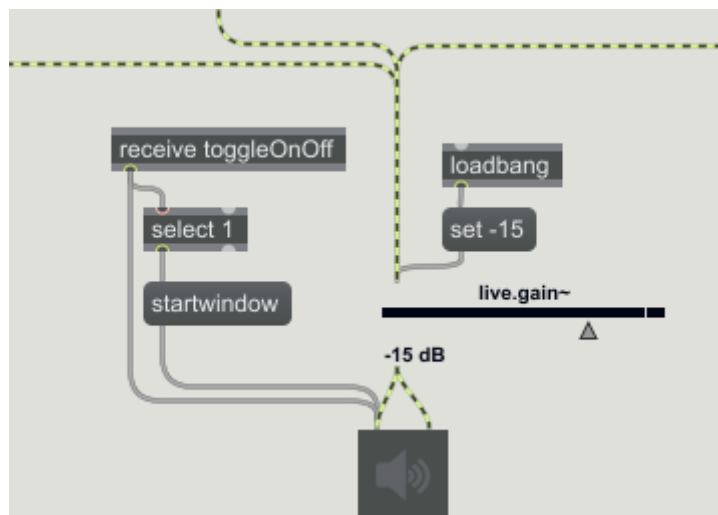
Σχήμα 5β. Κώδικας Max/MSP για τον ήχο αναμονής



Σχήμα 5γ. Κώδικας Max/MSP για τον ήχο επιτυχίας



Σχήμα 5δ. Κώδικας Max/MSP για τον ήχο αποτυχίας



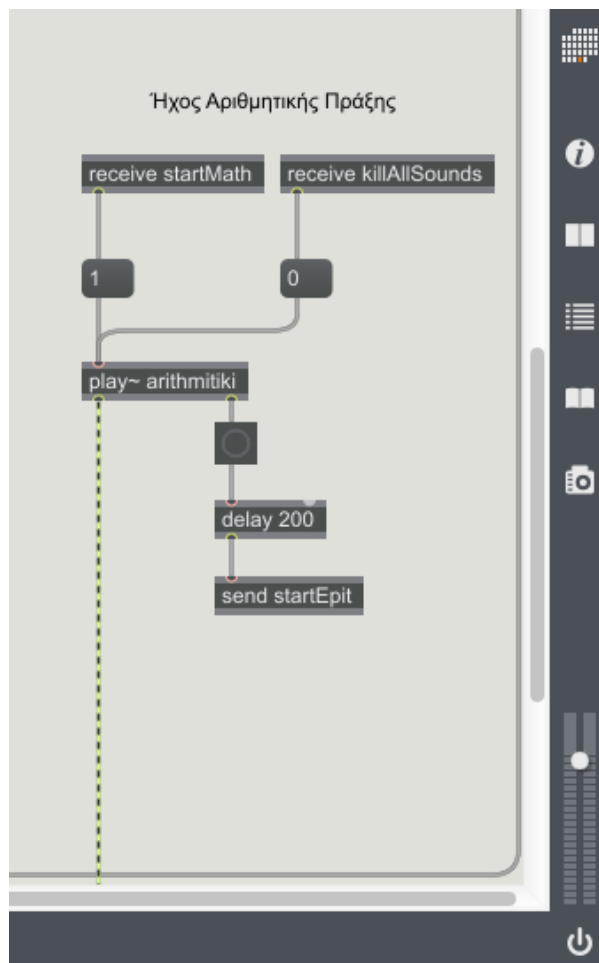
Σχήμα 5ε. Κώδικας Max/MSP ενεργοποίησης εκάστοτε ηχητικής ειδοποίησης

4.2.2 Σενάριο 2: Αριθμητική πράξη

Ο κώδικας στο δεύτερο σενάριο προγράμματος (Σχήμα 6α) εκτελεί την εκχώρηση μίας αριθμητικής τιμής σε μια προκαθορισμένη μεταβλητή ακολουθούμενη από μια απλή αριθμητική λειτουργία (προσθήκη σε μια σταθερά). Το πρώτο μέρος αυτού, συμπεριλαμβανομένης της ανάθεσης της εισόδου του χρήστη σε μια μεταβλητή, είναι ταυτόσημο με το σενάριο 1. Επομένως ο κώδικας λειτουργεί ακριβώς όπως περιγράφεται στην ενότητα 4.2.1 και οι προειδοποιήσεις τόσο του κώδικα της Γλωσσομάθειας όσο και του Max / MSP παραμένουν οι ίδιες. Όταν εκτελείται η αριθμητική λειτουργία, το περιβάλλον της Γλωσσομάθειας εκτυπώνει το αποτέλεσμα στην οθόνη του υπολογιστή, ενώ το περιβάλλον δοκιμής Max / MSP παράγει μια ειδοποίηση ήχου που δείχνει επίσης ότι έχει πραγματοποιηθεί μια αριθμητική λειτουργία (Σχήμα 6β).

```
! Arithmetic operation!
Program Scenario_2
Variables
    Integer: A
Begin
    Read A
    A ← A + 1
End
```

Σχήμα 6α. Αριθμητική πράξη



Σχήμα 6β. Κώδικας Max/MSP για ήχο που αντιστοιχεί με αριθμητική πράξη

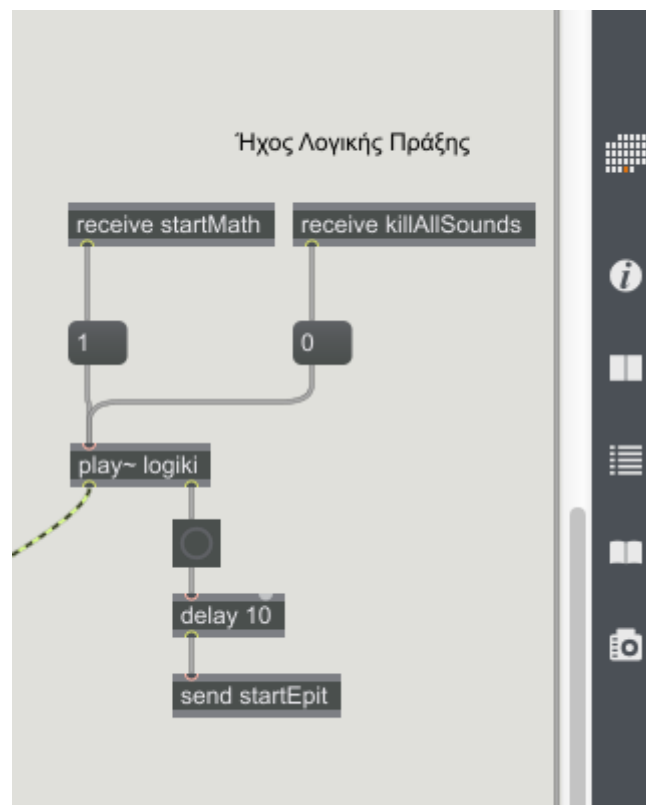
4.2.3 Σενάριο 3: Σχεσιακή πράξη (πράξη σύγκρισης- εκτέλεση λογικής πράξης)

Ο κώδικας στο τρίτο σενάριο προγράμματος (Σχήμα 7α) εκτελεί μια εκχώρηση αριθμητικής τιμής σε μια προκαθορισμένη μεταβλητή που ακολουθείται από μια απλή σχεσιακή λειτουργία (σύγκριση της μεταβλητής εισόδου με την αριθμητική τιμή 1). Το πρώτο μέρος αυτού, συμπεριλαμβανομένης της ανάθεσης της εισόδου του χρήστη σε μια μεταβλητή, είναι ταυτόσημο με το σενάριο 1. Επομένως ο κώδικας λειτουργεί ακριβώς όπως περιγράφεται στην ενότητα 4.2.1 και οι προειδοποιήσεις στον κώδικα της Γλωσσομάθειας και στο Max / MSP παραμένουν οι ίδιες.

```
! Relation operation!  
Program Scenario_3  
Variables  
    Integer: A  
    Boolean: X  
Begin  
    Read A  
    X ← A > 1  
End
```

Σχήμα 7α. Πράξη σύγκρισης

Όταν εκτελείται η σχεσιακή λειτουργία (πράξη σύγκρισης), το περιβάλλον της Γλωσσομάθειας εκτυπώνει το αποτέλεσμα Boolean (0: ψευδής, 1: αληθής) στην οθόνη του υπολογιστή, ενώ το περιβάλλον δοκιμής Max / MSP παράγει μια ειδοποίηση ήχου (Σχήμα 7β), που δείχνει επίσης ότι έχει πραγματοποιηθεί μια σχεσιακή (συγκριτική) λειτουργία.



Σχήμα 7β. Κώδικας Max/MSP για ήχο που αντιστοιχεί με λογική πράξη

4.2.4 Σενάριο 4: Δομή επιλογής (έλεγχος συνθηκών)

Αυτό το σενάριο είναι ένας συνδυασμός του 1ου και του 3ου σεναρίου για την καλύτερη κατανόηση της δομής επιλογής "αν ... τότε ... αλλιώς" (Σχήμα 8). Ο κώδικας εκτελεί μια εκχώρηση αριθμητικής τιμής (πραγματικής) σε μια προκαθορισμένη μεταβλητή που ακολουθείται από μια απλή σχεσιακή λειτουργία (σύγκριση της μεταβλητής εισόδου με την αριθμητική τιμή 9.5). Το πρώτο μέρος αυτού, συμπεριλαμβανομένης της ανάθεσης της εισόδου του χρήστη σε μια μεταβλητή, είναι ταυτόσημο με το σενάριο 1. Επομένως ο κώδικας λειτουργεί ακριβώς, όπως περιγράφεται στην Ενότητα 4.2.1 και οι προειδοποιήσεις του κώδικα στη Γλωσσομάθεια και στο Max / MSP παραμένουν οι ίδιες. Όταν εκτελείται η σχεσιακή λειτουργία, το περιβάλλον της Γλωσσομάθειας εκτυπώνει το αποτέλεσμα Boolean στην οθόνη του υπολογιστή (1: true ή 0: false) και ένα διαφορετικό μήνυμα ανάλογα. Αν το αποτέλεσμα Boolean είναι αληθές, το περιβάλλον δοκιμής Max / MSP παράγει επίσης μια επιτυχημένη ειδοποίηση ήχου (Σχήμα 5γ), διαφορετικά μια ειδοποίηση ήχου αποτυχίας (Σχήμα 5δ).

```

! Conditional statements!
Program Scenario_4
Variables
    Real: A
Begin
    Read A
    If A>9.5 then
        Write 'Past'
    Else
        Write 'Failure'
    End _ If
End

```

Σχήμα 8. Δομή επιλογής (έλεγχος συνθηκών)

4.2.5 Σενάριο 5: δομή επανάληψης - "όσο επανάλαβε (whiledo)"

Αυτό το σενάριο είναι ένας συνδυασμός του 1ου, του 3ου και του 4ου σεναρίου (Σχήμα 9). Ο κώδικας εκτελεί μια επαναλαμβανόμενη λειτουργία. Για να πραγματοποιηθεί η επανάληψη, πρέπει η συνθήκη να είναι αληθής. Συνεπώς σε κάθε

επανάληψη, το αποτέλεσμα Boolean είναι αληθές (δηλαδή 1: true), οπότε το περιβάλλον δοκιμής Max / MSP παράγει μια ηχητική ειδοποίηση επιτυχίας. Όταν το αποτέλεσμα Boolean είναι ψευδές (δηλαδή 0: false), ο βρόχος σταματά και το περιβάλλον δοκιμής Max / MSP παράγει μια ειδοποίηση ήχου αποτυχίας. Όσες φορές εκτελείται η επανάληψη ακούγεται τόσες φορές ο ήχος επιτυχίας και το loop ολοκληρώνεται με το άκουσμα της ηχητικής ειδοποίησης αποτυχίας. Ο ήχος επιτυχίας προσομοιώνεται με την ισχύ της συνθήκης (πραγματοποιείται η επαναληπτική διαδικασία), ενώ ο ήχος αποτυχίας αντίστοιχα με τον τερματισμό της επανάληψης, διότι η συνθήκη γίνεται ψευδής.

```
! Looping - "while...do"!
Program Scenario_5
Variables
    Integer: A
Begin
    A ← 1
    While A <= 5 Do
        Write A
        A ← A + 1
    End
End
```

Σχήμα 9. Δομή επανάληψης - "όσο επανάλαβε (whiledo)"

4.2.6 Σενάριο 6: δομή επανάληψης - "Αρχή επανάληψης μέχρις ότου (repeat...until)"

Το αντίθετο αποτέλεσμα, όσον αφορά στην ισχύ της συνθήκης, παρουσιάζεται ακριβώς σε αυτό το 6ο πείραμα (Σχήμα 10). Σε κάθε επανάληψη το αποτέλεσμα Boolean είναι ψευδές (0: false), οπότε το περιβάλλον δοκιμής Max / MSP παράγει μια ειδοποίηση ήχου αποτυχίας. Όταν το αποτέλεσμα Boolean είναι αληθές (1: true), ο βρόχος διακόπτεται και το περιβάλλον δοκιμής Max / MSP παράγει μια επιτυχημένη ειδοποίηση ήχου. Σ αυτήν την περίπτωση λοιπόν, όσες φορές εκτελείται η επανάληψη ακούγεται τόσες φορές ο ήχος αποτυχίας και το loop ολοκληρώνεται, με το άκουσμα της ηχητικής ειδοποίησης επιτυχίας. Ακριβώς το αντίθετο με τη λειτουργία του σεναρίου 4.2.5.


```
! Looping - "repeat...until"!
Program Scenario_6
Variables
    Integer: A
Begin
    A ← 1
    Repeat
        Write A
        A ← A + 1
    Until A > 5
End
```

Σχήμα 10. Δομή επανάληψης - "Αρχή επανάληψης μέχρις ότου (repeat...until)"

4.3 Αποτελέσματα του δεύτερου πειράματος

(Στατιστικά –Βαθμός επιρροής ήχου)

Οι αξιολογήσεις των συμμετεχόντων, σχετικά με την επίδραση των ηχητικών σημάτων στην κατανόηση του κώδικα, συνοψίζονται στον πίνακα 3 και στον πίνακα 4. Όπως φαίνεται στον πίνακα 3, περισσότερο από το 60% αυτών, έδειξε ότι η χρήση ηχητικών σημάτων είχε θετικό αποτέλεσμα στην ανάγνωση (εισαγωγή δεδομένου), στην επιτυχή ανάθεση δεδομένων σε μια μεταβλητή και στην εσφαλμένη εκτέλεση κώδικα (σφάλμα). Στον πίνακα 4, παραπάνω από το 85% των μαθητών που καταγράφηκε, αποδεικνύει ότι η χρήση των ειδοποιήσεων ήχου είχε θετικό αποτέλεσμα (μέτριο ή μεγάλο) στην κατανόηση του κώδικα για τις ακόλουθες λειτουργίες / ενέργειες:

- έλεγχος σχεσιακής λειτουργίας (if .. then ... else) και
- δομές επανάληψης (while ... do, repeat ... until).

Επιπλέον, φαίνεται ότι η χρήση των ειδοποιήσεων ήχου δεν είχε καμία επίδραση (ουδέτερη) στους χρήστες στην περίπτωση αριθμητικών και σχεσιακών πράξεων, γεγονός που θα μπορούσε να ερμηνευθεί με δύο διαφορετικούς τρόπους:

- είτε οι χρήστες προτιμούν γενικά τις ηχητικές ειδοποιήσεις για τα γεγονότα που σχετίζονται με εκτέλεση (ή ορθή ή εσφαλμένη) του κώδικα και ειδοποιήσεις για εκκρεμείς ενέργειες
- είτε οι συγκεκριμένες ερωτήσεις σχεδίασης και αξιολόγησης του πειράματος δεν ήταν κατάλληλες για τη δοκιμή της αποτελεσματικότητας των ειδοποιήσεων ήχου για τέτοιες πράξεις.

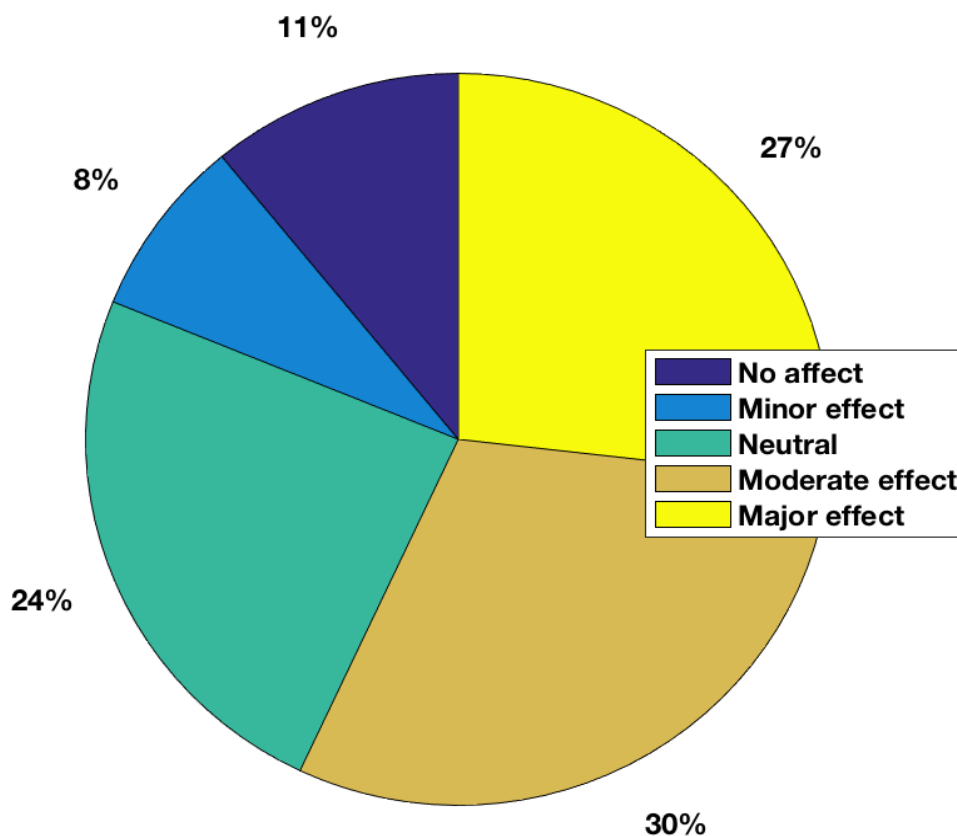
Θα πρέπει, επίσης, να σημειωθεί ότι από τους πενήντα τρεις (53) συμμετέχοντες, το 12% δήλωσε ότι η συμπληρωματική χρήση των ηχητικών σημάτων δεν είχε καμία επίδραση στην κατανόηση του κώδικα (Πίνακας 3) σε λιγότερο σημαντικά θέματα προγραμματισμού, όπως αυτά των τριών πρώτων σεναρίων. Οι υπόλοιποι μαθητές θεωρούσαν ότι τα ακουστικά συνθήματα είχαν αντίκτυπο στην εκμάθησή τους. Η παρατήρηση αυτή αντικατοπτρίζεται επίσης στο Σχήμα 11, το οποίο υποδηλώνει κατά μέσο όρο τις εκτιμήσεις των χρηστών σε σχέση με τα τρία πρώτα δοκιμασμένα σεναρία. Όπως φαίνεται, περίπου το 57% των μαθητών θεώρησε ότι το αποτέλεσμα ήταν μέτριο ή σημαντικό σε σύγκριση με το 19% που θεώρησε ότι το αποτέλεσμα ήταν ήσσονος σημασίας ή ανύπαρκτο.

Από την άλλη πλευρά, λιγότερο από το 4% των ογδόντα (80) συμμετεχόντων λένε ότι ο ήχος δεν τους επηρεάζει στην εκμάθηση, αλλά το αποτέλεσμα είναι το ίδιο, για πιο περίπλοκα θέματα προγραμματισμού, όπως οι δηλώσεις υπό όρους και επανάληψης (Πίνακας 4). Οι υπόλοιποι μαθητές θεωρούσαν ότι τα ακουστικά συνθήματα είχαν αντίκτυπο στην εκμάθησή τους. Η παρατήρηση αυτή απεικονίζεται επίσης στο σχήμα 12.

Είναι ενδιαφέρον ότι οι συμμετέχοντες έδειξαν ισχυρότερη προτίμηση για ειδοποιήσεις ήχου σχετικά με εκκρεμείς δραστηριότητες και με τις σωστές ή εσφαλμένες εκτελέσεις του κώδικα σε σχέση με άλλες λειτουργίες. Αυτό μπορεί να συσχετιστεί με το γεγονός ότι οι ηχητικές ειδοποιήσεις λειτουργούν καλά ως ενίσχυση της μνήμης, στο να ειδοποιούνται οι χρήστες για τυχόν συμβάντα που σχετίζονται με κώδικα και απαιτούν δράση. Η ερμηνεία μας για τις αξιολογήσεις των συγκεκριμένων πειραμάτων υποστηρίζεται και από ορισμένες γραπτές παρατηρήσεις των συμμετεχόντων. Για παράδειγμα, κάποιοι μαθητές έγραψαν: "Προτιμώ τους ήχους για επιτυχία και λάθος", "ο εκκρεμής ήχος ενέργειας με βοήθησε να καταλάβω ότι ήταν η στιγμή να εισάγουμε κάποια δεδομένα", "οι ηχητικές ειδοποιήσεις για αριθμητικές και σχεσιακές πράξεις δεν ήταν τόσο σημαντικές", "καταλαβαίνουμε, τότε εκτελείται μια επαναληπτική διαδικασία και τον διακριτικό χαρακτήρα μεταξύ της δομής "whiledo" και της δομής "repeatuntil".

Πίνακας 3. Στατιστικά στοιχεία των τριών πρώτων σεναρίων - εντολές κατανόησης με ήχους (αναλυτικά ανά διαδικασία)

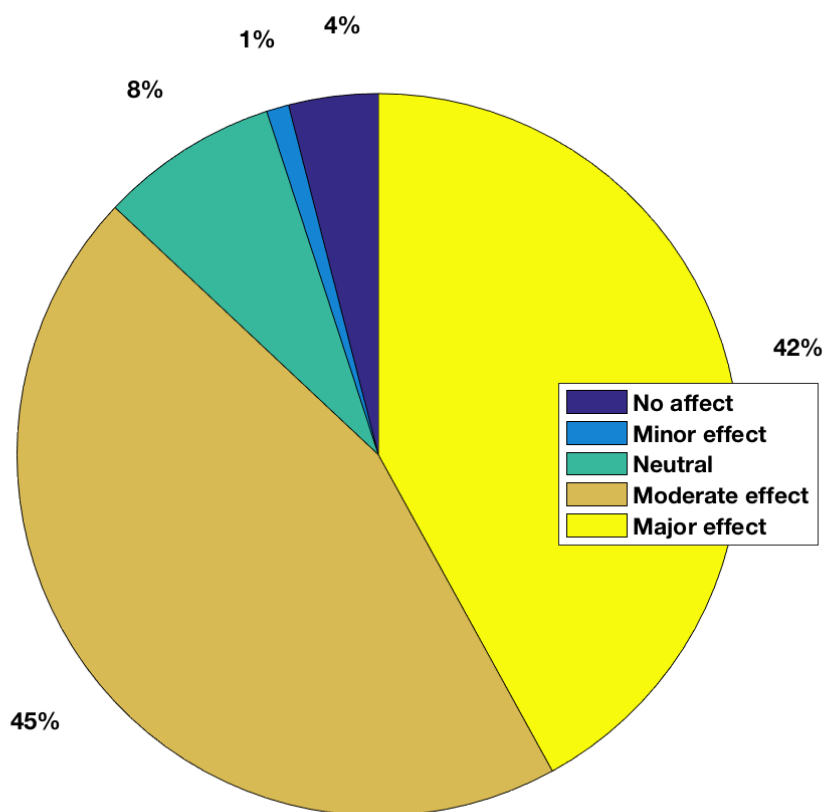
Operation	No affect	Minor affect	Neutral	Moderate affect	Major affect
Read	11,32%	0,00%	24,53%	30,19%	33,96%
Assignment	11,32%	1,89%	7,55%	35,85%	43,40%
Error	9,43%	3,77%	15,09%	39,62%	32,08%
Arithmetic	13,21%	15,09%	37,74%	16,98%	16,98%
Relational	9,43%	18,87%	35,85%	28,30%	7,55%



Σχήμα 11. Συνολική αξιολόγηση της χρήσης των ηχητικών προειδοποιήσεων σχετικά με την κατανόηση του κώδικα, κατά μέσον όρο στα τρία πρώτα σενάρια διδασκαλίας

Πίνακας 4. Στατιστικά στοιχεία των τριών τελευταίων σεναρίων - εντολές κατανόησης με ήχους (αναλυτικά ανά διαδικασία)

Operation	No affect	Minor affect	Neutral	Moderate affect	Major affect
If ... then ...else (Conditional statements)	2,50 %	0,00 %	10,00%	47,50%	40,00%
While....do (looping)	3,75 %	0,00%	8,75%	50,00%	37,50%
Repeat.... Until (looping)	5,00 %	2,50%	6,25%	38,75%	47,50%



Σχήμα 12. Συνολική αξιολόγηση της χρήσης των ηχητικών προειδοποιήσεων σχετικά με την κατανόηση του κώδικα, κατά μέσον όρο στα τρία τελευταία σενάρια διδασκαλίας

5. Τρίτο πείραμα - Μελέτη αξιολόγησης

Στη συνέχεια διεξήχθη ένα τρίτο πείραμα για την αξιολόγηση της αποτελεσματικότητας των ηχητικών προειδοποιήσεων. Εκατό (100) μαθητές/μαθήτριες, εκ των οποίων σαράντα επτά (47) αγόρια, διαγωνίστηκαν σε θέματα με επαναληπτικές διαδικασίες. Συμμετείχαν πενήντα (50) μαθητές/μαθήτριες, εκ των οποίων είκοσι τέσσερα (24) αγόρια, οι οποίοι δεν είχαν διδαχθεί τα πειράματα με ήχους και πενήντα (50) μαθητές, εκ των οποίων είκοσι τρία (23) αγόρια, οι οποίοι είχαν διδαχθεί και μάλιστα συμμετείχαν στα προηγούμενα πειράματα. Προκειμένου να είμαστε όσο το δυνατόν αμερόληπτοι, επιλέξαμε σκόπιμα και τις δύο ομάδες των πενήντα (50) μαθητών/μαθητριών να είναι ισοδύναμες, με βάση την προηγούμενη επίδοσή τους (και οι δύο ομάδες είχαν επιτύχει την ίδια μέση επίδοση στην αντίστοιχη πορεία, μέχρι το πείραμα αυτό).

5.1 Πρωτόκολλο τρίτου πειράματος

Η δοκιμή περιλάμβανε δέκα (10) διαδικασίες επαναληπτικού βρόχου και οι μαθητές έπρεπε να απαντήσουν για το πλήθος των επαναλήψεων κάθε διαδικασίας. Βαθμολογήθηκαν σε εκατονταβάθμια κλίμακα.

5.2 Δοκιμή

Η δοκιμή περιλάμβανε δέκα (10) τμήματα κώδικα. Κάθε τμήμα ήταν μια διαδικασία επαναληπτικού βρόχου, όπως "while do " ή " repeat until ". Οι μαθητές έπρεπε να διακρίνουν πότε μια διαδικασία επαναλήφθηκε και να καταγράψουν πόσες φορές επαναλήφθηκε. Επιπλέον, οι μαθητές έπρεπε να καταλάβουν ότι η διαδικασία "while do" **επαναλαμβάνεται** όταν μια λογική συνθήκη είναι αληθής, ενώ η διαδικασία " repeat until" **τερματίζεται** όταν η λογική συνθήκη είναι αληθής. Στη συνέχεια οι μαθητές έπρεπε να απαντήσουν, ποιος είναι ο αριθμός των επαναλήψεων κάθε διαδικασίας.

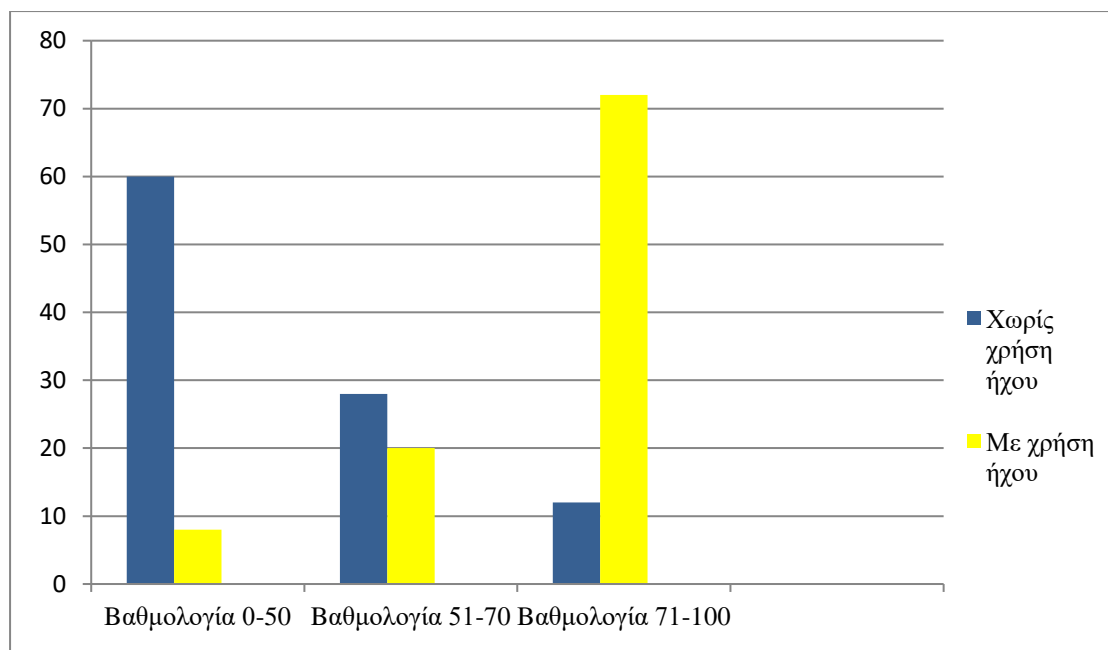
5.3 Αποτελέσματα δοκιμών

Τα αποτελέσματα των δοκιμών είναι πραγματικά εντυπωσιακά και απεικονίζονται στον Πίνακα 5 και στο Σχήμα 13. Το πολύ υψηλό ποσοστό του 92% των μαθητών που συμμετείχαν σε όλες τις έξι (6) δοκιμές, εμπλουτισμένες με σενάρια διδασκαλίας

ηχητικών σημάτων, βαθμολογήθηκαν πάνω από 50/100. Από την άλλη πλευρά, το αντίστοιχο ποσοστό των μαθητών που δεν συμμετείχαν στα σενάρια μας, ήταν μόνο το 40%. Τα αποτελέσματα είναι ακόμα πιο εντυπωσιακά, δηλαδή 72% έναντι 12% αντίστοιχα, αν επικεντρωθούμε στους μαθητές με υψηλή βαθμολογία (δηλαδή, στους μαθητές που βαθμολογήθηκαν πάνω από 70/100). Ομοίως επίσης, τα αποτελέσματα για τους μαθητές με χαμηλό βαθμό, (8% έναντι 60%, αντιστοίχως). Τα αποτελέσματα των δοκιμών έδειξαν ότι η χρήση του ήχου ήταν ένας σημαντικός παράγοντας για την κατανόηση της λειτουργίας εντολών κώδικα και υπογραμμίζουν τη σημασία του ρόλου των ηχητικών προειδοποιήσεων στη διδασκαλία βασικών θεμάτων προγραμματισμού, όπως οι δομές επανάληψης. Είναι σαφές ότι οι μαθητές που συμμετείχαν στα πειράματα με τις ενσωματωμένες ειδοποιήσεις ήχου, παρουσιάζουν πολύ καλύτερη κατανόηση όσον αφορά στους επαναληπτικούς βρόχους από τους μαθητές που δεν συμμετείχαν στο πείραμα.

Πίνακας 5. Στατιστικά στοιχεία βαθμολογίας

Βαθμολογία	Μαθητές που διδάχθηκαν προγραμματισμό <u>χωρίς</u> τη χρήση ήχου	Μαθητές που διδάχθηκαν προγραμματισμό <u>με</u> τη χρήση ήχου
0-50	60 %	8 %
51-70	28 %	20 %
71-100	12 %	72 %



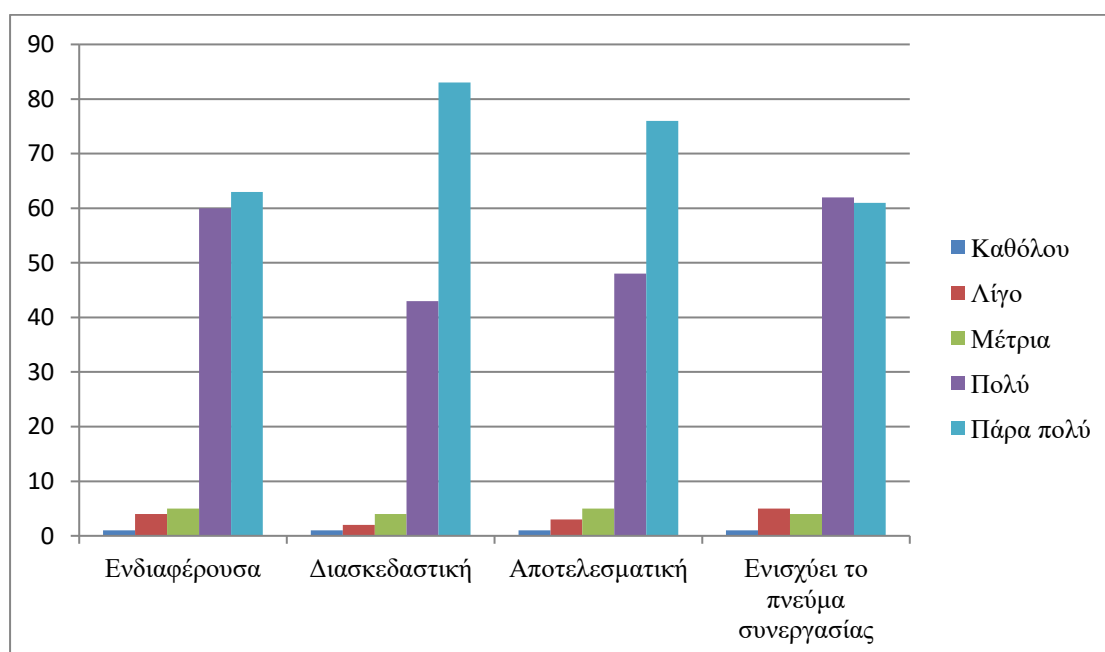
Σχήμα 13. Αξιολόγηση των μαθητών συμμετεχόντων και μη, στα διδακτικά σενάρια της έρευνας (Ποσοστά βαθμολογιών)

6. Αξιολόγηση και συμπεράσματα μαθητών

Τέλος μοιράστηκαν ερωτηματολόγια στους εκατό τριάντα τρεις (133) μαθητές/μαθήτριες, εκ των οποίων εξήντα δύο (62) αγόρια, που συμμετείχαν στα διδακτικά σενάρια, ώστε να αξιολογήσουν τη συμμετοχή τους στα πειράματα και να συμπληρώσουν τι αποκόμισαν από αυτό το νέο εγχείρημα. Τα αποτελέσματα περιγράφονται στον Πίνακα 6 και στο Σχήμα 14, όπου διαφαίνεται, ότι η συντριπτική πλειονότητα (περίπου 92 %) των μαθητών/μαθητριών θεωρεί ενδιαφέρουσα, διασκεδαστική, αποτελεσματική και συνεργατική όλη τη διαδικασία των πειραμάτων της έρευνας.

Πίνακας 6. Αξιολόγηση της διαδικασίας των πειραμάτων της διατριβής από τους μαθητές

Διαδικασία πειραμάτων	Καθόλου	Λίγο	Μέτρια	Πολύ	Πάρα πολύ
Ενδιαφέρουσα	1	4	5	60	63
Διασκεδαστική	1	2	4	43	83
Αποτελεσματική	1	3	5	48	76
Ενισχύει το πνεύμα συνεργασίας	1	5	4	62	61



Σχήμα 14. Αξιολόγηση και συμπεράσματα των συμμετεχόντων μαθητών στα διδακτικά σενάρια της έρευνας

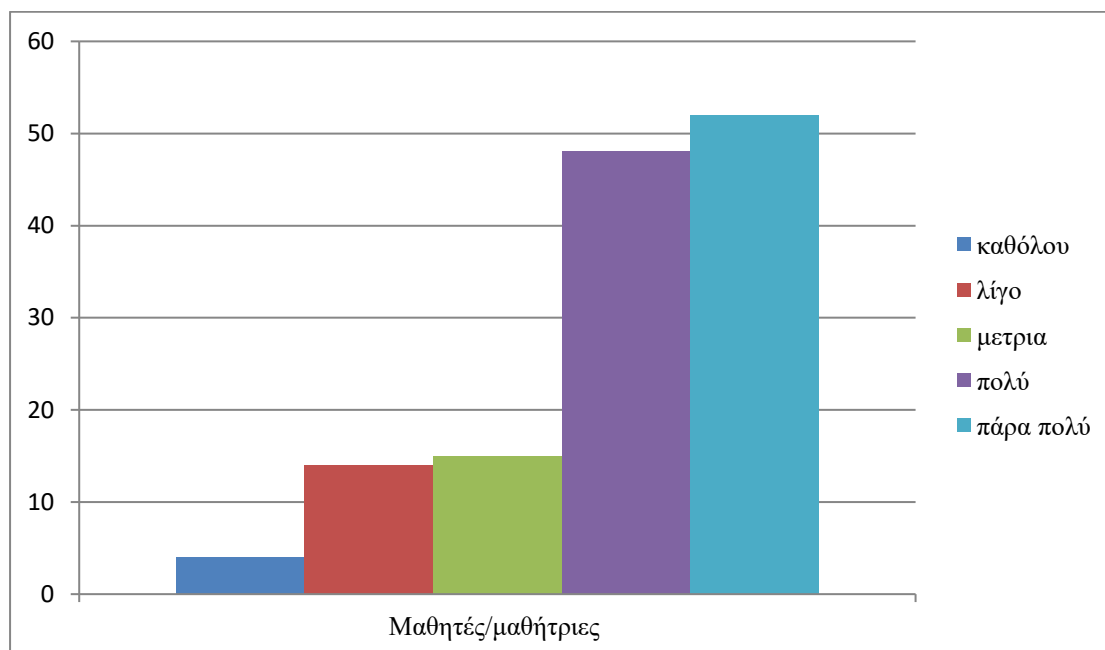
Στη συνέχεια στους παραπάνω μαθητές/μαθήτριες μοιράστηκαν δύο διερευνητικά ερωτηματολόγια, για να ελεγχθεί :

1. Αν οι μαθητές/μαθήτριες θεωρούν πιο αποτελεσματική διαδικασία να δημιουργούν εκείνοι/νες τις ηχητικές ειδοποιήσεις και
2. Σε ποιες διαδικασίες θεωρούν ότι χρειάζονται περισσότερο οι ηχητικές ειδοποιήσεις, για την καλύτερη κατανόηση του κώδικα.

Τα αποτελέσματα περιγράφονται στους Πίνακες 7 & 8, καθώς και στα Σχήματα 15 & 16 αντίστοιχα, όπου διαφαίνεται, ότι η πλειονότητα (περίπου 75 %) των μαθητών/μαθητριών θεωρεί προτιμότερο η δημιουργία των ηχητικών ειδοποιήσεων να γίνεται από τους μαθητές και ότι η πλειονότητα (περίπου 84 %) των μαθητών/μαθητριών θεωρεί σημαντική την παρουσία ηχητικών ειδοποιήσεων, σε διαδικασίες όπως δομές επανάληψης, αναζήτηση στοιχείου σε πίνακα, ταξινόμηση στοιχείων πίνακα και κλήση υποπρογραμμάτων.

Πίνακας 7. Δημιουργία ηχητικών ειδοποιήσεων από τους ίδιους/ίδιες τους /τις μαθητές/μαθήτριες

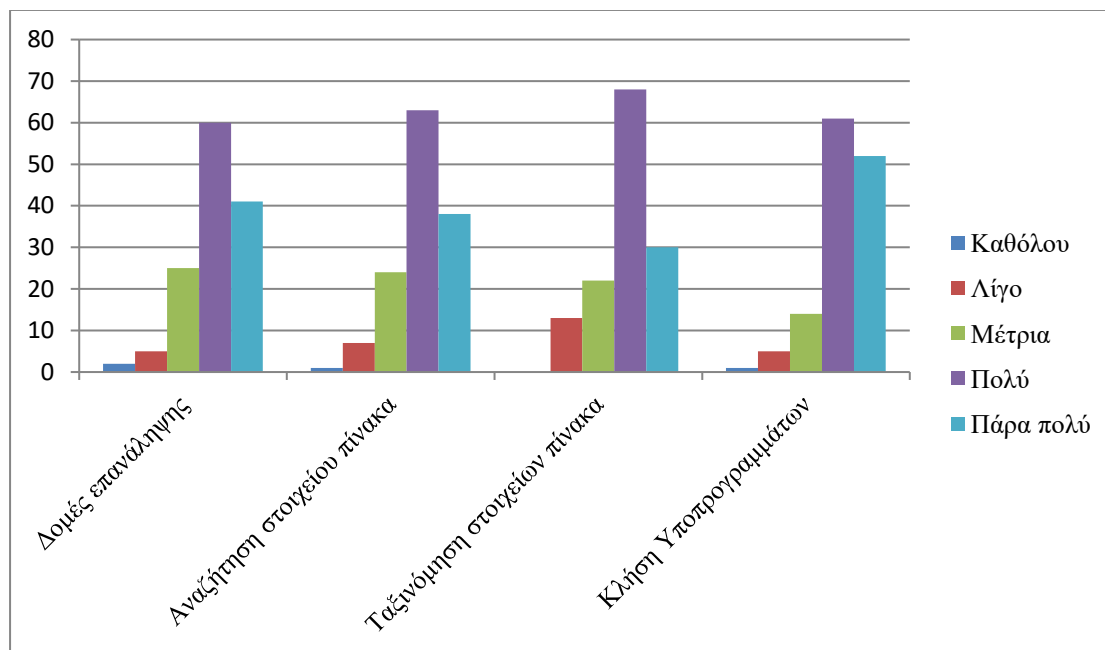
Δημιουργία ηχητικών ειδοποιήσεων από :	Καθόλου	Λίγο	Μέτρια	Πολύ	Πάρα πολύ
Μαθητές/μαθήτριες	4	14	15	48	52



Σχήμα 15. Συμμετοχή των μαθητών /τριών στη δημιουργία ηχητικών ειδοποιήσεων

Πίνακας 8. Διαδικασίες και ηχητικές ειδοποιήσεις

Διαδικασία	Καθόλου	Λίγο	Μέτρια	Πολύ	Πάρα πολύ
Δομές Επανάληψης	2	5	15	70	41
Αναζήτηση στοιχείου σε πίνακα	1	7	14	73	38
Ταξινόμηση στοιχείων πίνακα	0	13	12	78	30
Κλήση Υποπρογραμμάτων	1	5	7	68	52



Σχήμα 16. Αξιολόγηση των ηχητικών ειδοποιήσεων ανά διαδικασία

7. Συμπεράσματα & νέα πεδία διερεύνησης

Με αφετηρία το πρώτο σκέλος του **πρώτου** ερευνητικού ερωτήματος για το αν υπάρχουν σχετικά λογισμικά για την καλύτερη κατανόηση και εκμάθηση του αλγοριθμικού προγραμματισμού, έγινε έρευνα και παρουσιάστηκαν οι διδακτικές προσεγγίσεις (ενότητα 2.1) καθώς και τα εκπαιδευτικά εργαλεία που χρησιμοποιούνται στη Δευτεροβάθμια Εκπαίδευση (ενότητες 2.2, 2.5).

Στη συνέχεια απαντώντας στο δεύτερο σκέλος του **πρώτου** ερευνητικού ερωτήματος, για το ποιες είναι οι αδυναμίες των υπαρχόντων προγραμμάτων, παρουσιάστηκαν οι δυσκολίες στην κατανόηση του αλγοριθμικού προγραμματισμού και οι λανθασμένες αντιλήψεις των μαθητών που διατηρούνται λανθασμένες ακόμη και με την χρήση λογισμικών που περιγράφονται παραπάνω και θεωρούνται φιλικά στον χρήστη, τα οποία όμως διεγείρουν στην πλειονότητά τους μόνο την οπτική αντίληψη (ενότητα 2.3).

Στην παρούσα διατριβή ως απάντηση στο πρώτο σκέλος του **δεύτερου** ερευνητικού ερώτημα, για το ποια είναι η πρωτοτυπία της νέας πρότασης, συζητήθηκαν τα πιθανά οφέλη από τη χρήση ακουστικών ειδοποιήσεων (sound alert) ως συμπληρωματικών εργαλείων για τη διδασκαλία του προγραμματισμού στη Δευτεροβάθμια Εκπαίδευση (Κεφάλαιο 3). Η εργασία βασίζεται στην υπόθεση ότι οι ηχητικές αλληλεπιδράσεις υπολογιστή-χρηστών θα μπορούσαν να επηρεάσουν θετικά την ανάπτυξη των δεξιοτήτων επίλυσης προβλημάτων από τους μαθητές, να διεγείρουν την ψυχοακουστική τους αντίληψη και να βελτιώσουν την κατανόηση του κώδικα προγραμματισμού (αποτελέσματα που περιγράφονται στις ενότητες 4.3, 5.3), θέματα που αποτελούν την απάντηση στο δεύτερο σκέλος του **δεύτερου** ερευνητικού ερωτήματος, για το ποιος είναι ο στόχος και η συμβολή της νέας πρότασης.

7.1 Συμπεράσματα έρευνας

Παρουσιάστηκαν τρεις μελέτες. Η πρώτη εξέτασε τον τύπο των ήχων που θα ήταν προτιμότερος για μια τέτοια εργασία, ενώ η δεύτερη και η τρίτη, ανεξάρτητα η

καθεμιά από την άλλη, ερεύνησε το αν η χρήση ηχητικών σημάτων επηρεάζει στην κατανόηση κώδικα ή όχι.

Στην πρώτη μελέτη εξετάστηκαν δύο διαφορετικές κατηγορίες ήχου: καταγεγραμμένα αποσπάσματα ήχων πουλιών, αναφερόμενα ως φυσικοί ήχοι στο κείμενο, και σύνθετοι ηλεκτρονικοί ήχοι από συνθεσάιζερ, που αναφέρονται ως συνθετικοί ήχοι ή τεχνητοί ήχοι στο κείμενο. Είχαν καθοριστεί πέντε (5) διαφορετικές υπολογιστικές διαδικασίες και / ή λειτουργίες:

- Ανάγνωση δεδομένων από πληκτρολόγιο
- Επιτυχής ανάθεση δεδομένων
- Σφάλμα
- Αριθμητική λειτουργία και
- Σχεσιακή λειτουργία (σχέση σύγκρισης).

Για κάθε διαδικασία επιλέχθηκε ένα ζευγάρι ηχητικών ερεθισμάτων και συνδυάστηκε με έναν ήχο από κάθε μία από τις δύο κατηγορίες. Τα ζευγάρια ήχου μοιράζονταν κοινές μουσικές και ψυχοακουστικές ιδιότητες. Οι συμμετέχοντες έδειξαν πολύ ισχυρή και σταθερή προτίμηση προς τους συνθετικούς (τεχνητούς) ήχους, απορρίπτοντας τους φυσικούς σχεδόν ομόφωνα σε όλες τις δοκιμασμένες διαδικασίες.

Στη δεύτερη μελέτη, οι συμμετέχοντες έπρεπε να αξιολογήσουν την επίδραση των ηχητικών ειδοποιήσεων σχετικά με την κατανόηση του κώδικα, δεδομένου ότι έλαβαν έξι (6) διδακτικά σενάρια προγραμματισμού. Οι μαθητές συνολικά αξιολόγησαν θετικά τη χρήση της ακουστικής ανατροφοδότησης. Στα τρία πρώτα σενάρια το 57% των συμμετεχόντων ανέφερε ότι οι ενδείξεις είχαν μέτρια ή μείζονα επίδραση, ενώ μόνο το 11% δεν έδειξε να επηρεάζεται από τη διαδικασία. Στα τελευταία τρία σενάρια το 87% των συμμετεχόντων ανέφερε ότι οι ενδείξεις είχαν μέτρια ή μείζονα επίδραση, ενώ μόνο το 4% δεν έδειξε καμία επίδραση.

Στην τρίτη μελέτη παρατηρήθηκε μια εξαιρετική βελτίωση στη βαθμολογία του test που δόθηκε στους μαθητές. Το 92% των πενήντα (50) μαθητών, που συμμετείχαν στα έξι (6) σενάρια διδασκαλίας με ήχο, βαθμολογήθηκαν πάνω από 50/100, ενώ μόνο το 40% των υπολοίπων πενήντα (50) μαθητών, που δεν συμμετείχαν σε σενάρια, βαθμολογήθηκαν πάνω από 50/100. Τα αποτελέσματα μας οδηγούν στο συμπέρασμα ότι οι μαθητές που μαθαίνουν τον δομημένο προγραμματισμό με τον ήχο επιτυγχάνουν καλύτερη μάθηση από εκείνους που μαθαίνουν με τη συμβατική διδασκαλία χωρίς ήχο - συμπέρασμα που αποτελεί και απάντηση στο **τρίτο** ερευνητικό ερώτημα, για το αν

μαθητές που μαθαίνουν τον δομημένο προγραμματισμό με ενσωματωμένο ήχο επιτυγχάνουν καλύτερη μάθηση/κατανόηση από εκείνους που μαθαίνουν με τη συμβατική διδασκαλία χωρίς ήχο (ενότητες 5.3, Κεφάλαιο 6).

Στα ερωτηματολόγια που μοιράστηκαν για να αξιολογήσουν όλη τη διαδικασία των πειραμάτων της έρευνας, στη συντριπτική πλειονότητά τους (περίπου 92%) οι μαθητές τη θεώρησαν ενδιαφέρουσα, διασκεδαστική, αποτελεσματική και συνεργατική.

Ορισμένοι συμμετέχοντες στα πειράματα ανέφεραν ότι ένας συνδυασμός ειδοποιήσεων ήχου και φωνητικών μηνυμάτων θα μπορούσε να είναι αποτελεσματικός στην κατανόηση του προγραμματισμού. Αυτό είναι, βεβαίως, μια διαδρομή που αξίζει να εξεταστεί καθώς προχωρεί αυτό το έργο.

Δεδομένων των παραπάνω, στη συνέχεια παρατίθενται οι προοπτικές και τα μελλοντικά πλάνα που προκύπτουν από την έρευνα και τα αποτελέσματά της.

7.2 Προοπτικές – Μελλοντικά πλάνα

Το πρώτο βήμα στα μελλοντικά πλάνα θα ήταν να συμπεριληφθούν οι ηχητικές προειδοποιήσεις σε περισσότερες διαδικασίες και λειτουργίες και να τις δοκιμάσουμε σε πιο πολύπλοκα σενάρια διδασκαλίας, όπως ενσωμάτωση ήχων στους προγραμματιστικούς μηχανισμούς που διδάσκονται στη Δευτεροβάθμια Εκπαίδευση, π.χ. αναζήτηση, ταξινόμηση, κλήση υποπρογραμμάτων, κατηγορίες λαθών, διαγράμματα ροής με στόχο να διαπιστώσουμε αν η προτεινόμενη μεθοδολογία μας θα μπορούσε να είναι αποτελεσματική σε κατανόηση δυσκολότερων εννοιών προγραμματισμού ή όχι.

Αναφορά γίνεται στις παρακάτω υποενότητες, εστιάζοντας στις δυσκολίες που έχουν παρατηρηθεί ανά ενότητα.

7.2.1 Εμφωλευμένες επαναληπτικές δομές

Θα ενσωματώσουμε ηχητικές ειδοποιήσεις σε πιο σύνθετους επαναλαμβανόμενους βρόχους, που ενδεχομένως περιλαμβάνουν πολλαπλές δηλώσεις υπό όρους. Στόχος αποτελεί η καλύτερη κατανόηση των εσωτερικών και εξωτερικών επαναλήψεων, ο χρόνος εκτέλεσής τους και γενικότερα η λειτουργία των εμφωλευμένων αλγοριθμικών δομών.

7.2.2 Δομές δεδομένων (Πίνακες, Δυναμικές δομές)

Θα σχεδιαστούν σενάρια με ηχητικές ειδοποιήσεις, τόσο σε μονοδιάστατους ή δισδιάστατους πίνακες, όσο και σε δυναμικές δομές δεδομένων, ώστε να κατανοηθεί η διαχείριση μνήμης στις στατικές δομές δεδομένων, τότε παρατηρείται η υπερχείλιση πίνακα και τότε η υποχείλιση πίνακα, η ώθηση και η απόθεση στοιχείου σε μια στοίβα, η είσοδος ή η έξοδος στοιχείου σε/από ουρά, η διαδικασία ανοίγματος ενός αρχείου ή ανάλογα το κλείσιμο ενός αρχείου, όσον αφορά στις δυναμικές δομές δεδομένων.

Ειδικότερα στους δισδιάστατους πίνακες μπορούν να ενταχθούν ηχητικές ειδοποιήσεις κατά την αλλαγή γραμμής ή στήλης αντίστοιχα.

Το περιβάλλον που θα σχεδιαστεί μπορεί επίσης να περιλαμβάνει βηματική εκτέλεση των εντολών, ώστε τέτοιες διαδικασίες όπως η διαχείριση δομών δεδομένων να γίνονται ακόμη πιο κατανοητές.

7.2.3 Αναζήτηση

Στους αλγόριθμους αναζήτησης στοιχείου/στοιχείων δυσκολίες παρατηρήθηκαν τόσο σε ταξινομημένους πίνακες όσο και σε αταξινομητους. Γι' αυτό και τα σενάρια θα ποικίλλουν. Θα εφαρμοστούν ηχητικά μηνύματα σε αναζήτηση στοιχείου ή στοιχείων σε ένα μη ταξινομημένο πίνακα δεδομένων ή αναζήτηση στοιχείου με δυαδική αναζήτηση σε ένα ταξινομημένο πίνακα.

7.2.4 Ταξινόμηση

Οι πιο γνωστοί αλγόριθμοι ταξινόμησης που εντάσσονται στο πρόγραμμα σπουδών στη Δευτεροβάθμια Εκπαίδευση είναι ο αλγόριθμος bubble sort (φουσαλίδα) και ο αλγόριθμος selection sort (με επιλογή). Θα ενταχθούν ηχητικά μηνύματα κατά την ταξινόμηση πίνακα μονοδιάστατου με τον αλγόριθμο φουσαλίδας ή τον αλγόριθμο ταξινόμησης με επιλογή, εστιάζοντας στην παρουσίαση του ήχου τόσο στη σύγκριση, όσο και στην αντιμετάθεση στοιχείων. Στη συνέχεια τα σενάρια μπορούν να επεκταθούν, σε περιπτώσεις που υπάρχουν παράλληλοι πίνακες, αλλά και σε ταξινομήσεις δισδιάστατων πινάκων.

7.2.5 Κλήση υποπρογραμμάτων

Ο δομημένος προγραμματισμός που βασίζεται στην τμηματοποίηση της επίλυσης ενός προβλήματος, αλλά και στην ιεραρχική σχεδίαση της, πρόβαλε δυσκολίες, όπως η

κατηγοριοποίηση των υποπρογραμμάτων, όσον αφορά στην κλήση τους, την ενεργοποίησή τους και την επιστροφή τιμών. Για την αντιμετώπιση αυτών των δυσκολιών, για την καλύτερη κατανόηση και διάκριση των δύο κατηγοριών υποπρογραμμάτων (διαδικασίας ή συνάρτησης), θα σχεδιαστούν σενάρια με παρουσία ηχητικής ειδοποίησης σε κάθε περίπτωση.

7.2.6 Κατηγορίες λαθών

Θα ελεγχθεί αν οι τρεις κατηγορίες λαθών που διδάσκονται, συντακτικά, λογικά, λάθη κατά την εκτέλεση (αντικανονικός τερματισμός προγράμματος), μπορούν να προσομοιώνονται με αντίστοιχες ηχητικές ειδοποιήσεις, ώστε οι μαθητές να αντιληφθούν τα λάθη που παρουσιάζονται στη διάρκεια ζωής ενός προγράμματος από τη συγγραφή μέχρι την εκτέλεση και να προβούν γρήγορα και ευκολότερα στον εντοπισμό και διόρθωση λαθών.

7.2.7 Διαγράμματα ροής

Τα σχήματα των διαγραμμάτων ροής μπορούν να αντιστοιχηθούν με ανάλογες ηχητικές ειδοποιήσεις, όπως επίσης και η ροή του διαγράμματος μπορεί να έχει διαφορετική ηχητική ειδοποίηση ανάλογα με τη φορά της.

7.2.8 Σενάρια διδασκαλίας με μουσικά δεδομένα

Επιπροσθέτως, θα δοκιμάσουμε σενάρια διδασκαλίας, σε περιβάλλον Max / MSP, τα οποία θα χρησιμοποιούν μουσικά δεδομένα. Για παράδειγμα θα αναζητούμε πόσες φορές βρίσκεται μια νότα ή ένα μοτίβο σε μια μελωδία (σειριακή αναζήτηση σε μουσικά δεδομένα).

7.2.9 Ολοκληρωμένο προγραμματιστικό περιβάλλον με ήχο

Όλα τα παραπάνω απαιτούν τη δημιουργία μιας βάσης δεδομένων, μιας ηλεκτρονικής βιβλιοθήκης ηχητικών μηνυμάτων για την επίτευξη προσομοίωσης εντολών με ήχο, την καταγραφή του ήχου των διαγραμμάτων ροής, την ερμηνεία και τον εντοπισμό - διόρθωση των λαθών κατά τον χρόνο εκτέλεσης, τα συντακτικά και λογικά σφάλματα χρησιμοποιώντας τον ήχο, ούτως ώστε να συνιστούν όλα μαζί ένα νέο ολοκληρωμένο εκπαιδευτικό εργαλείο.

Ο απώτερος στόχος μας λοιπόν είναι η σχεδίαση ενός νέου πλήρους εκπαιδευτικού εργαλείου, η δημιουργία ενός προγραμματιστικού περιβάλλοντος, όπως η

Γλωσσομάθεια, με ειδοποιήσεις ήχου, σε όλο το φάσμα των λειτουργιών. Παράλληλα θα υπάρχει οπτική ανατροφοδότηση που θα χρησιμοποιεί τον ήχο (προϋπόθεση η δημιουργία βάσης δεδομένων ακουστικών και φωνητικών μηνυμάτων) για την ερμηνεία και την προσομοίωση των εντολών ενός προγράμματος καθώς και των γραφημάτων ενός διαγράμματος ροής του, για την ερμηνεία συντακτικών λαθών και των σφαλμάτων κατά τον χρόνο εκτέλεσης ενός προγράμματος, ώστε οι μαθητές της δευτεροβάθμιας εκπαίδευσης να κατανοούν καλύτερα τη διδασκαλία του προγραμματισμού.

Θεωρούμε ότι ακόμα πιο καινοτόμος είναι η κατασκευή ηχητικών ειδοποιήσεων από τους ίδιους μαθητές και η αντιστοίχιση με ανάλογες ενέργειες-εντολές, για το σχηματισμό της βάσης δεδομένων.

7.2.10 Τελικό συμπέρασμα

Ολοκληρώνοντας την έρευνα, τα πειράματα και την αξιολόγησή τους, το τελικό συμπέρασμα είναι ότι η νέα διδακτική προσέγγιση που χρησιμοποιεί τον ήχο για την καλύτερη κατανόηση του αλγοριθμικού προγραμματισμού, μπορεί να χρησιμεύσει αφενός στην περαιτέρω έρευνα για τα οφέλη του ήχου και την ψυχοακουστική αντίληψη των εφήβων όσο στην παιδαγωγική διαδικασία. Επιπλέον παρέχει το εφόδιο στον εκπαιδευτικό αφενός να υπερκεράσει τις δυσκολίες που μέχρι τώρα αντιμετώπιζαν τόσο εκείνος στο να διδάξει τις έννοιες του προγραμματισμού, όσο και οι μαθητές, (στο να κατανοήσουν την επικοινωνία ανθρώπου με ένα υπολογιστικό σύστημα) και αφετέρου να ενισχύσει-βελτιώσει τη διάδραση στην τάξη.

Ειδικότερα σε απάντηση στο τρίτο σκέλος του **δεύτερου** ερευνητικού ερωτήματος για το ποια είναι τα παιδαγωγικά οφέλη της νέας προσέγγισης, η νέα πρόταση ενδυναμώνει τον ρόλο του καθηγητή που διεγείρει την ψυχοακουστική αντίληψη των μαθητών του και το ενδιαφέρον τους, δημιουργώντας ευρηματικά σενάρια και ενθαρρύνοντας τη συμμετοχή τους. Οι μαθητές δε, από απλοί δέκτες πληροφορίας μετατρέπονται σε βασικούς ενεργούς συντελεστές της εκπαιδευτικής διαδικασίας. Εύκολα και ευχάριστα θα έχουν πρόσβαση στο μαθησιακό υλικό, αναπτύσσοντας έτσι ένα οργανωμένο τρόπο σκέψης. Τέλος, η συμμετοχή τους στη διαμόρφωση του εκπαιδευτικού περιβάλλοντος μέσω αυτής της αλληλεπίδρασης με τον υπολογιστή, αποτελεί μια πρόκληση για το σύγχρονο σχολείο που στοχεύει στην ανάπτυξη ικανοτήτων και τεχνικών μάθησης. Ικανότητες και τεχνικές μάθησης που θα βοηθήσουν τους μαθητές να ανταποκριθούν

στις νέες μαθησιακές και κοινωνικές ανάγκες, δεδομένου ότι η σημερινή εποχή των ΤΠΕ απαιτεί ενεργητικές διδακτικές προσεγγίσεις και υψηλού επιπέδου δεξιότητες από μαθητές που να μπορούν να διαχειρίζονται περίπλοκες διαδικασίες και να τοποθετούνται υπεύθυνα και κριτικά απέναντί τους.

8. Πίνακας ορολογίας

Algorithmic programming: αλγοριθμικός προγραμματισμός

AND: λογικός τελεστής – σύζευξη

Arithmetic: Αριθμητικός

Assignment: Εκχώρηση τιμής

Auditory display: Ακουστική αναπαράσταση

Boolean result: Λογικό αποτέλεσμα (αληθής, ψευδής)

Computational thinking: υπολογιστική σκέψη

Cycling 74 MAX: Γραφικό περιβάλλον προγραμματισμού για ήχο και MIDI

Diff range: Εύρος διαφοράς

Divide et impera: διαίρει και βασίλευε

Drag and drop: μεταφέρω και αποθέτω

False: ψευδής

Lingua Franca: κοινή διάλεκτος

LIKERT : Διαδεδομένη προσέγγιση για την κλιμάκωση των απαντήσεων μιας έρευνας, που πήρε το όνομά της από τον εφευρέτη της ψυχολόγο Rensis Likert

(2AFC :κλίμακα 2 επιλογών, 5AFC: κλίμακα 5 επιλογών)

Major affect: πάρα πολύ

Minor affect: λίγο

Moderate affect: πολύ

Neutral: μέτρια

No affect: καθόλου

NOT: λογικός τελεστής - άρνηση

Operation: διαδικασία (ενέργεια)

OR: λογικός τελεστής- διάζευξη

Pitch: ακουστική συχνότητα

Psychoacoustic perception: ψυχοακουστική αντίληψη

Relational: Συγκριτικός

Selection rate: Ποσοστό επιλογής

Simulation: προσομοίωση

Sound alert: ηχητική ειδοποίηση

Structures: δομές

Tablet: φορητός υπολογιστής

Timbre: χροιά του τόνου

True: αληθής

9. Συντμήσεις – Αρκτικόλεξα – Ακρωνύμια

AFC: Alfa Forced Choice

AI: App Inventor

API: Application Programming Interface

GUI: Graphical User Interface

USB: Universal Serial Bus (Ενιαίος Σειριακός Δίαυλος: εξωτερικό μέσο αποθήκευσης)

ΑΕΠΠ: Ανάπτυξη εφαρμογών σε προγραμματιστικό περιβάλλον

Ι.Ε.Π.: Ινστιτούτο Εκπαιδευτικής Πολιτικής

ΕΚΠΑ: Εθνικό Καποδιστριακό Πανεπιστήμιο Αθηνών

ΠΑΔΑ: Πανεπιστήμιο Δυτικής Αττικής

Π.Ι: Παιδαγωγικό Ινστιτούτο

ΤΠΕ: Τεχνολογίες της Πληροφορίας και των Επικοινωνιών

ΦΕΚ: Φύλλο Εφημερίδας Κυβερνήσεως

10. Παραρτήματα

10.1 Ερωτηματολόγιο - Επιλογή ήχου

(A: φυσικός ήχος, B: τεχνητός ήχος)

Φύλο Αγόρι

Κορίτσι

1. Αναμονή για εισαγωγή δεδομένων

A B

2. Επιτυχία

A B

3. Σφάλμα

A B

4. Αριθμητικές πράξεις

A B

5. Λογικές Πράξεις

A B

10.2 Ερωτηματολόγιο – Κατανόηση εντολών

Φύλο Αγόρι

Κορίτσι

Πόσο πιστεύετε ότι επηρέασε ο ήχος στην καλύτερη κατανόηση των παρακάτω διαδικασιών :

1. Αναμονή για εισαγωγή δεδομένων

καθόλου λίγο μέτρια πολύ πάρα πολύ

2. Επιτυχία

καθόλου λίγο μέτρια πολύ πάρα πολύ

3. Σφάλμα

καθόλου λίγο μέτρια πολύ πάρα πολύ

4. Αριθμητικές πράξεις

καθόλου λίγο μέτρια πολύ πάρα πολύ

5. Λογικές Πράξεις

καθόλου λίγο μέτρια πολύ πάρα πολύ

Γενικό σχόλιο :

.....

.....

.....

.....

.....

10.3 Ερωτηματολόγιο – Κατανόηση δομών επανάληψης

Φύλο Αγόρι

Κορίτσι

Πόσο πιστεύετε ότι επηρέασε ο ήχος στην καλύτερη κατανόηση των παρακάτω
Αλγοριθμικών δομών;

1. Δομή επανάληψης «ΟΣΟ....»

καθόλου	λίγο	μέτρια	πολύ	πάρα πολύ
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2. Δομή επανάληψης «ΜΕΧΡΙΣ_ΟΤΟΥ....»

καθόλου	λίγο	μέτρια	πολύ	πάρα πολύ
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

3. Δομή επανάληψης «ΓΙΑ»

καθόλου	λίγο	μέτρια	πολύ	πάρα πολύ
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

10.4 Ερωτηματολόγιο – Αξιολόγηση της διαδικασίας των πειραμάτων της διατριβής από τους μαθητές

Φύλο Αγόρι
 Κορίτσι

Αξιολογήστε την εκμάθηση του προγραμματισμού με τη χρήση ηχητικών σημάτων

1. Ενδιαφέρουσα

καθόλου	λίγο	μέτρια	πολύ	πάρα πολύ
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2. Διασκεδαστική

καθόλου	λίγο	μέτρια	πολύ	πάρα πολύ
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

3. Αποτελεσματική

καθόλου	λίγο	μέτρια	πολύ	πάρα πολύ
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

4. Ενισχύει το πνεύμα συνεργασίας

καθόλου	λίγο	μέτρια	πολύ	πάρα πολύ
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**10.5 Ερωτηματολόγιο – Συμμετοχή στη δημιουργία ηχητικών
ειδοποιήσεων**

Φύλο Αγόρι

Κορίτσι

**Πόσο πιστεύετε ότι η δημιουργία των ηχητικών ειδοποιήσεων είναι πιο
αποτελεσματική να γίνει από τους/τις μαθητές/μαθήτριες;**

καθόλου	λίγο	μέτρια	πολύ	πάρα πολύ
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

10.6 Ερωτηματολόγιο – Διαδικασίες & ηχητικές ειδοποιήσεις

Φύλο Αγόρι

Κορίτσι

Σε ποια από τις παρακάτω διαδικασίες χρειάζεται περισσότερο η παρουσία ηχητικών ειδοποιήσεων;

1. Δομές επανάληψης

καθόλου λίγο μέτρια πολύ πάρα πολύ

2. Αναζήτηση στοιχείου σε πίνακα

καθόλου λίγο μέτρια πολύ πάρα πολύ

3. Ταξινόμηση στοιχείων πίνακα

καθόλου λίγο μέτρια πολύ πάρα πολύ

4. Κλήση υποπρογραμμάτων

καθόλου λίγο μέτρια πολύ πάρα πολύ

11. Βιβλιογραφία

11.1 Ελληνόγλωσση Βιβλιογραφία

Αβούρης, Ν.,(2000). Εισαγωγή στην Επικοινωνία Ανθρώπου-Υπολογιστή Αθήνα: Δίαυλος

Αβούρης, Ν., Κατσάνος, Χ., Τσέλιος, Ν., Μουστάκας, Κ.,(2016). Εισαγωγή στην Αλληλεπίδραση Ανθρώπου-Υπολογιστή. 2^η Έκδοση

Αγγελιδάκης, Ν.(2015). Εισαγωγή στον προγραμματισμό με την Python. Ηράκλειο

Αλεξανδράκης, Ν.Π. (2001). *Εισαγωγή στον δομημένο προγραμματισμό και στη γλώσσα Pascal*. Athens.

Ασλανίδου, Σ. (2010). Εκπαιδευτική Τεχνολογία. Από την οπτικοακουστική στην ψηφιακή αγωγή. Θεσσαλονίκη: Κυριακίδη.

Βακαλούδη, Α. (2003), Διδάσκοντας και μαθαίνοντας με τις νέες τεχνολογίες θεωρία και πράξη, Αθήνα: Πατάκης.

Γρηγοριάδου, Μ., Γόγολου, Α., Γουλή, Ε., Γλέζου, Κ., Μπούμπουκα, Μ., Παπανικολάου, Κ., Τσαγκάνου, Γ., Κανίδης, Ε., Δουκάκης, Δ., Φράγκου, Σ., & Βεργίνης, Η. (2009). *Διδακτικές Προσεγγίσεις και Εργαλεία για τη διδασκαλία της Πληροφορικής*. Αθήνα: Νέων Τεχνολογιών.

Δαγδιλέλης Β., (1996) Διδακτική της πληροφορικής. Η διδασκαλία του προγραμματισμού: αντιλήψεις των σπουδαστών για την κατασκευή και επικύρωση προγραμμάτων και διδακτικές καταστάσεις για τη διαμόρφωσή τους, Διδακτορική διατριβή, Τμήμα Εφ. Πληροφορικής Πανεπιστήμιο Μακεδονίας.

Ξυνόγαλος, Σ. (2000) Προγραμματιστικοί Μικρόκοσμοι: Μια άλλη προσέγγιση της Διδασκαλίας του Προγραμματισμού, *Πρακτικά του Συνεδρίου «Πληροφορική και Εκπαίδευση» του Σ.Ε.Π.Δ.Ε.θ., Θεσσαλονίκη, Νοέμβριος 2000*.

Ξυνόγαλος Σ. & Σατρατζέμη Μ. (2001) Αρχές Σχεδίασης Ολοκληρωμένων Περιβαλλόντων Προγραμματισμού για Αρχάριους, *Πρακτικά 5ου Πανελληνίου Συνεδρίου με Διεθνή Συμμετοχή «Διδακτική των Μαθηματικών και Πληροφορική στην Εκπαίδευση», Θεσσαλονίκη 12-14 Οκτωβρίου 2001*.

Ξυνόγαλος Σ. (2002), Εκπαιδευτική Τεχνολογία: Ένας Διδακτικός Μικρόκοσμος για την Εισαγωγή στον Αντικειμενοστραφή Προγραμματισμό, Διδακτορική Διατριβή, Τμήμα Εφαρμοσμένης Πληροφορικής, Πανεπιστήμιο Μακεδονίας.

Ευνόγαλος Σ., Σατρατζέμη Μ., (2004) Η Εισαγωγή στον Αντικειμενοστραφή Προγραμματισμό: Διδακτικά Προβλήματα και Μεθοδολογίες για την Αντιμετώπισή τους, *4ο Πανελλήνιο Συνέδριο με διεθνή συμμετοχή «Οι Τεχνολογίες της Πληροφορίας και της Επικοινωνίας στην Εκπαίδευση»*, Αθήνα, 29 Σεπτεμβρίου – 3 Οκτωβρίου 2004, σελ. 133-142 (τόμος Β).

Ευνόγαλος Σ., Σατρατζέμη Μ., Δαγδιλέλης Β., Ευαγγελίδης Γ., (2005), "Η Διδασκαλία της Κληρονομικότητας στον Προγραμματιστικό Μικρόκοσμο objectKarel", *Πρακτικά 3ου Πανελληνίου Συνεδρίου Η Διδακτική της Πληροφορικής*, Κόρινθος, 7-9 Οκτωβρίου 2005.

Παπάζογλου, Π., Λιώνης, Σ.-Π. (2014). Ανάπτυξη εφαρμογών με το Arduino. Εκδότης: Τζιόλας

Πουλάκης, Ε. (2015) Προγραμματίζοντας με τον μικροελεγκτή Arduino. Ηράκλειο

Σατρατζέμη Μ., Ευνόγαλος Σ., Δαγδιλέλης Β., (2006), «Εκπαιδευτικά Προγραμματιστικά Περιβάλλοντα για τη Διδασκαλία του Αντικειμενοστραφούς Προγραμματισμού: μια επισκόπηση», *5ο ΠΑΝΕΛΛΗΝΙΟ ΣΥΝΕΔΡΙΟ ΜΕ ΔΙΕΘΝΗ ΣΥΜΜΕΤΟΧΗ -ΟΙ ΤΕΧΝΟΛΟΓΙΕΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΤΗΣ ΕΠΙΚΟΙΝΩΝΙΑΣ ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ (ΕΤΠΕ 2006)*, ΘΕΣΣΑΛΟΝΙΚΗ, 5 - 8 ΟΚΤΩΒΡΙΟΥ 2006, σελ. 899-906.

Τζιμογιάννης, Α., Τσιωτάκης, Π., Τσάκωνας, Π., Βραχνός, Π. (2019). Βασικές παρανοήσεις στη διδασκαλία προγραμματισμού.

Τσολακίδης, Κ. & Φωκίδης Μ. (2007). Εικονική πραγματικότητα στην εκπαίδευση. Αθήνα: Ατραπός-Περιβολάκι.

Χρονάκη, Α., & Κούριας, Σ. (2011) Παιδιά, Ρομπότ και Lego Mindstorms: Καταγράφοντας το ξεκίνημα μιας αλληλεπιδραστικής σχέσης. Στο Χ. Θ. Παναγιωτακόπουλος (επιμ.), *Πρακτικά του 2ου Πανελληνίου Συνεδρίου "Ένταξη και χρήση των ΤΠΕ στην Εκπαιδευτική Διαδικασία"* (σελ. 1009-1022). Πάτρα: ΕΤΠΕ.

11.2 Ξενόγλωσση Βιβλιογραφία

Aaron, S., Blackwell, A.F., Burnard, P. (2016). The development of Sonic Pi and its use in educational partnerships: Co-creating pedagogies for learning computer programming. *Journal of Music, Technology & Education*. **9** (1): 75–94. Retrieved 11 December 2019.

Bayman, P., Mayer, R. E. (1983). A Diagnosis of Beginning Programmers' Misconceptions of BASIC Programming Statements, *Communications of the ACM*, vol. 26, n° 9, pp. 677-679.

Bonar, J., Soloway, E. (1985). Preprogramming Knowledge: A Major Source of Misconceptions in Novice Programmers. In *Human-Computer Interaction* 1(2):133-161.

Bonar, J., Soloway, E. (1989). Preprogramming Knowledge: A Major Source of Misconceptions in Novice Programmers. *Studying the Novice Programmer*. E. Soloway and J. C. Spohrer. Hillsdale, NJ, Lawrence Erlbaum Associates: 325-353.

Brennan, K. (2009). Scratch-Ed: an online community for scratch educators. In *Proceedings of the 9th international conference on Computer supported collaborative learning-Volume 2* (pp. 76-78). International Society of the Learning Sciences.

Burt, J. L., Bartolome, D. S., Burdette, D. W., Comstock, J. R. Jr. (1995). A psychophysiological evaluation of the perceived urgency of auditory warning signals. *Ergonomics*, 38(11), 2327-2340.

Cerf, V.G. (2018). The sound of programming. *Communications of the ACM*, 61(4), 6.

Cipriani, A., Giri, M. (2010). *Electronic Music and Sound Design - Theory and Practice with Max/MSP - volume 1*, 2nd edition. Rome: ConTempoNet.

Dagdilelis, V. (1986). *Conceptions des eleves apropos des notions fondamentales de la programmation informatique en classe de Troisieme*, Memoire D.E.A., Universite Joseph FOURIER, Grenoble, France.

Dahl, O. J. Dijkstra E. W. and Hoare C. A. R. (1972). *Structured Programming*. Academic Press, New York.

Du Boulay, B. (1989). Some difficulties of learning to program. In Soloway, E., Spohrer, J.C. (Eds.), *Studying the Novice Programmer*, London, Lawrence Erlbaum Associates (pp. 283–299).

Fastl, H. (2006). Psychoacoustic basis of sound quality evaluation and sound engineering. *Proceeding of the International Congress on Sound and Vibration, 2006*. Vienna, Austria

Fluck, A., Webb, M., Cox, M., Angeli, C., Malyn-Smith, J., Voogt, J., & Zagami, J. (2016). Arguing for Computer Science in the School Curriculum. *Educational Technology & Society*, 19 (3), 38–46.

Genuit, K. (2004). *Sound quality in environment: Psychoacoustic mapping*. ASA 2004, San Diego, CA, USA.

Hsu, Y. C., Rice, K., & Dawley, L. (2012). Empowering educators with Google's Android App Inventor: An online workshop in mobile app design. *British Journal of Educational Technology*, 43(1), E1-E5.

- Hsu, T.-C., & Hwang, G.-J. (2017). Effects of a Structured Resource-based Web Issue-Quest Approach on Students' Learning Performances in Computer Programming Courses. *Educational Technology & Society*, 20 (3), 82–94.
- Kelleher, C. & Pausch, R. (2005). Lowering the Barriers to Programming: A Taxonomy of Programming Environments and Languages for Novice Programmers. ACM-Computer-Surveys.
- Kernighan, B.W., Ritchie, D. M. (1988). *The C Programming Language*. Prentice-Hall.
- Lockwood J., Mooney A. (2018) Computational Thinking in Secondary Education: Where Does It Fit? A Systematic Literary Review, *International Journal of Computer Science Education in Schools*
- Mathrani, A., Christian, S., & Ponder-Sutton, A. (2016). Play IT: Game Based Learning Approach for Teaching Programming Concepts. *Educational Technology & Society*, 19 (2), 5–17.
- Mayer, R. (1981). The Psychology of How Novices Learn Computer Programming, *Computing Surveys*, Vol. 13, n° 1, pp. 121-141.
- Milne, I. & Rowe, G. (2002). Difficulties in Learning and Teaching Programming—Views of Students and Tutors. *Education and Information Technologies* 7(1), 55-66.
- Noss, R. (1984). Children Learning Logo Programming. Interim Report No. 2 of the Chiltern Logo Project, Advisory Unit for Computer Based Education, Hatfield, United Kingdom.
- Papadakis, St., Kalogiannakis, M., Orfanakis, V., Zaranis, N. (2014). Novice Programming Environments. Scratch & App Inventor: a first comparison. In H. M. Fardoun and J. A. Gallud (Eds.) *Proceedings of the 2014 Workshop on Interaction Design in Educational Environments* (pp 1-7), New York: ACM.
- Papert, S. (1980) *Mindstorms, Children, Computers and Powerful Ideas*. Basic Books Inc. Harper Colophon Books.
- Pea, R.D. (1986). Language-independent conceptual “bugs” in novice programming. *Journal of Educational Computing Research* (pp. 2, 25–36).
- Place, T., Lossius, T. (2006). Jamoma: A modular standard for structuring patches in Max: Proc. of the International Computer Music Conference, pp. 143–146, New Orleans, US
- Prensky, M. (2001). *Digital Natives, Digital Immigrants*. MCB University Press, Vol. 9 No. 5
- Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., et al. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60–67
- Ringwood, J. V., Monaghan, K. & Maloco, J. (2005). Teaching engineering design through Lego Mindstorms. *European Journal of Engineering Education*, 30:1, (pp. 91 – 104)
- Robins, A., Rountree, J. & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, 13(2), 137-172.
- Rogalski J. (1985) Alphabétisation Informatique, *Bulletin de l'APMEP* n° 352, pp. 61-74.

Rogalski, J., (1989). Problem-Solving in Mathematics and in Informatics: Differences and Invariants. In J. Hoc, T Green, R. Samurçay & D. Gilmore (Eds.), *Psychology of Programming*, pp. 236-247.

Rouchier, A. & Samurçay, R. (1984). Concepts Informatiques et programmation: Une premiere analyse en classe de seconde des Lycees, Rapport de recherche CNRS.

Rubin, M.J. (2013) The effectiveness of live-coding to teach introductory programming SIGCSE '13: Proceeding of the 44th ACM technical symposium on Computer science education March 2013 Pages 651–656 <https://doi.org/10.1145/2445196.2445388>

Saeli, M., Perrenet, J., Jochems, W., Zwaneved, B., (2011). Teaching Programming in Secondary School: A Pedagogical Content Knowledge Perspective, *Informatics in Education*, (10), 73-88.

Samurçay, R. (1989). The Concept of Variable in Programming: its Meaning and Use in Problem-Solving by Novice Programmers, In E. Soloway, J. Sprohrer (Eds.) *Studying The Novice Programmer*, 161-178, Lawrence Erlbaum Associates.

Schneiderman, B. (1980). *Software Psychology, Human Factors in Computer and Information Systems*, Winthrop Publishers Inc', Cambridge.

Selby, C., Woollard, J. (2013). *Computational thinking: the developing definition*. University of Southampton (E-prints) 6pp.

Soloway, E. (1986) Learning to Program = Learning to Construct Mechanisms and Explanations, *Communications of the ACM*, Vol. 29, No. 9, pp. 850-858.

Soloway, E., Bonar, J. and Ehrlich, K. (1983). “Cognitive Strategies and Looping Constructs: An Empirical Study”, *Communications of the ACM*, 26(11), pp 853-860.

Soloway, E., Bonar, J. & Ehrlich, K. (1989) Cognitive Strategies and Looping Constructs: An Empirical Study. In *Studying The Novice Programmer*, Soloway, E., Sprohrer, J. (Eds.) Lawrence Erlbaum Associates, pp. 191-208.

Soloway, E., Ehrlich, K., Bonar, J. and Greenspan, J. (1982). “What de novices know about programming”, in A.N. Badre and B. Shneiderman (Eds.), *Directions in Human-Computer Interaction*, New York: Ablex.

Sprohrer, J., Soloway, E. & Pope, E. (1989) A Goal/Plan Analysis of Buggy Pascal Programs. In *Studying The Novice Programmer*, Soloway, E., Sprohrer, J. (Eds.) Lawrence Erlbaum Associates, pp. 355-400.

Stephenson, C., Gal-Ezer, J., Haberman, B. & Verno, A. (2005). *The New Educational Imperative: Improving High School Computer Science Education* (Rep. No. Final Report of the CSTA Curriculum Improvement Task Force, 2005).

Sweller, J. (2010). Element interactivity and intrinsic, extraneous, and germane cognitive load. *Educational psychology review*, 22(2), 123-138.

Szlávi, P. & Zsakó, L. (2006). Programming versus application In: Mittermeir, R.T. (Ed.), *ISSEP 2006, LNCS 4226* (p.p.48–58).

Toh, L. P. E., Causo, A., Tzuo, P. W., Chen, I. M., & Yeo, S. H. (2016). A Review on the Use of Robots in Education and Young Children. *Educational Technology & Society*, 19 (2), 148–163.

Walker, B. N., Kramer, G. (2004). Ecological Psychoacoustics and Auditory Displays: Hearing, Grouping, and Meaning Making

Weigend, M. (2006). From intuition to program. Programming versus application. In: Mittermeir.

Wirth, N. (1973). The Programming Language Pascal. Zurich.

11.3 Πηγές από το Διαδίκτυο

Algorithmos. Retrieved on 15 February, 2020 from: <http://www.algorithmos.gr/glossomatheia.html>

Alice. Retrieved on 15 February, 2021 [https://en.wikipedia.org/wiki/Alice_\(software\)](https://en.wikipedia.org/wiki/Alice_(software))

Beanz (2013). The magazine for kids, code, and computer science. Retrieved on 15 February, 2020 from: <https://www.kidscodex.com/resources/programming/education/>

Bruner, J. Retrieved on 25 February, 2021 from [Discovery learning - Wikipedia](#)

Ebooks.edu.gr. Retrieved on 15 February, 2020 from: <http://ebooks.edu.gr/courses/DSGL-C101/document/4c65902ff3dk/4e52d483egdp/4e52e406mj6l.pdf>

Edu.klimaka.gr. Retrieved on 15 February, 2021 from: <https://edu.klimaka.gr/mathimata/lykeiou/3179-programma-spoudwn-pliroforiki-g-lykeiou>

Gilfix, M., Couch, A. Peep (The Network Auralizer): Monitoring Your Network With Sound. Retrieved on 15 February, 2019 from : <https://www.usenix.org/legacy/events/lisa00/gilfix/gilfix.html/>

Gorson, J., Patel, N., Beheshti, E., Magerko, B. & Horn, M. (2017). TunePad: Computational Thinking Through Sound Composition. IDC '17: Proceedings of the 2017 Conference on Interaction Design and Children June 2017 Pages 484–489 <https://doi.org/10.1145/3078072.3084313>

Iep.edu.gr. Retrieved on 15 February, 2021 from: <http://iep.edu.gr/el/graf-b-yliko/geniko-lykeio>

Jerinic, L., Ivanovic, M., Puntik, Z., Budimac, Z., & Savic, M. (2014). e-Education in Teaching Programming - Forty Years of Promises? International Conference on e-Learning'14. Retrieved on 30 June, 2020 from: https://www.academia.edu/9828622/e-Education_in_Teaching_Programming_-_Forty_Years_of_Promises

Kirn, P. (2018). Roland and MIT want to use music to teach kids programming. Retrieved on 15 February, 2020 from: <http://cdm.link/2018/01/roland-mit-want-use-music-teach-kids-programming/>

Pi-schools.gr. Retrieved on 15 February, 2021 from: http://www.pi-schools.gr/content/index.php?lesson_id=1&ep=228

Piaget, Z. Retrieved on 25 February, 2021 from: [Ζαν Πιαζέ - Βικιπαίδεια \(wikipedia.org\)](#)

Bruner, J. Retrieved on 25 February, 2021 from [Discovery learning - Wikipedia](#)

Scratch Wiki (2015). Retrieved on 15 February, 2020 from: <https://wiki.scratch.mit.edu/wiki/ScratchX>

Scratch MIT. Retrieved on 3 February, 2021 from: <https://scratch.mit.edu/>

Sonic-pi. Retrieved on 15 February, 2020 from: <https://projects.raspberrypi.org/en/projects/getting-started-with-sonic-pi>

Spinet. Retrieved on 15 October, 2020 from: <http://spinet.gr/glossomatheia/>

Βιβλιογραφία

Szlavi, P. & Zsako, L. (2003) Methods of teaching programming. Retrieved on 25 February, 2020 from:

https://www.researchgate.net/publication/235925815_Methods_of_teaching_programming

Tynker. Retrieved on 3 February, 2021 from: <https://en.wikipedia.org/wiki/Tynker>

Yarbrough, D. (2017). Sound the alarm: how sounds affect our memory and emotions. Retrieved on 15 February, 2020 from:

https://www.voxmagazine.com/music/sound-the-alarm-how-sounds-affect-our-memory-and-emotions/article_153c4146-be25-11e7-b9ab-8b1620bcc28d.html