

Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση

Τόμ. 1 (2021)

10ο Πανελλήνιο Συνέδριο «Διδακτική της Πληροφορικής»



Το μοντέλο της Ιεραρχικής Αξιολόγησης
Γνώσεων Προγραμματισμού: μια εμπειρική μελέτη

Στ. Ζιώγα, Β. Κατωγιάννη, Α. Μικρόπουλος, Ι.
Μπέλλου

doi: [10.12681/cetpe.3718](https://doi.org/10.12681/cetpe.3718)

Το μοντέλο της Ιεραρχικής Αξιολόγησης Γνώσεων Προγραμματισμού: μια εμπειρική μελέτη

Στ. Ζιώγα¹, Β. Κατωγιάννη², Α. Μικρόπουλος³, Ι. Μπέλλου³
ziogastella68@gmail.com, vasilikikat2000@gmail.com
amikrop@uoi.gr, ibellou@uoi.gr

¹ Εκπαιδευτικός Πληροφορικής, ² ΤΜΗΥΠ Πανεπιστήμιο Ιωαννίνων,
³ ΠΤΔΕ Πανεπιστήμιο Ιωαννίνων

Περίληψη

Ο προγραμματισμός υπολογιστικών συστημάτων αποτελεί βασική και δύσκολη δεξιότητα. Οι εκπαιδευτικοί επιδιώκουν να βρουν τις καταλληλότερες τεχνικές αξιολόγησης ώστε να μπορούν αφενός να εκτιμήσουν με ακρίβεια τη γνώση των μαθητών, και αφετέρου να βελτιώσουν τη διδασκαλία τους. Η εργασία παρουσιάζει την εφαρμογή των ταξινομιών Bloom και SOLO στον προγραμματισμό ως πλαίσιο της σχεδίασης μαθησιακών δραστηριοτήτων και της αξιολόγησής τους αντίστοιχα. Παρουσιάζεται το μοντέλο της Ιεραρχικής Αξιολόγησης Προγραμματισμού (ΙΑΠ), και διερευνάται η αξιολόγηση των γνώσεων προγραμματισμού σε τελειόφοιτους μαθητές Λυκείου καθώς και η σύγκριση των αποτελεσμάτων της με τη συνηθισμένη εμπειρική αξιολόγηση από εκπαιδευτικούς. Τα αποτελέσματα επιβεβαιώνουν ευρήματα για την αξιολόγηση των γνώσεων προγραμματισμού και τις σχετικές δυσκολίες στη μάθηση του. Τα συμπεράσματα αναδεικνύουν ότι η ΙΑΠ αποτελεί ένα αναλυτικό εργαλείο ποιοτικής αξιολόγησης γνώσεων προγραμματισμού και ότι χρειάζεται επιπλέον διερεύνηση για τα δύο υψηλότερα επίπεδα της δομικής πολυπλοκότητας της.

Λέξεις κλειδιά: Προγραμματισμός, αξιολόγηση, ταξινόμια Bloom, ταξινόμια Solo, Ιεραρχική Αξιολόγηση Προγραμματισμού.

Εισαγωγή

Ο προγραμματισμός ως αντικείμενο διδασκαλίας προκαλεί εξαιρετικό ενδιαφέρον και αποτελεί ένα σύνθετο εννοιολογικό πεδίο. Συμβάλλει στην ανάπτυξη δεξιοτήτων υψηλού επιπέδου καθώς στη μεταφορά δεξιοτήτων επίλυσης προβλημάτων σε άλλα γνωστικά πεδία (Κόμης, 2001).

Οι περισσότερες έρευνες σχετικά με τον προγραμματισμό αφορούν στις διδακτικές τεχνικές, την κατανόηση αλλά και τις παρανοήσεις των μαθητών. Επικεντρώνονται στην επίδοση των μαθητών κατά τη διάρκεια της διδασκαλίας ή μετά, με εστίαση στα μαθησιακά αποτελέσματα, στα εμπόδια μάθησης και στις παρανοήσεις (Bennedsen & Caspersen, 2012). Λίγες μόνο μελέτες αφορούν την αξιολόγηση του προγραμματισμού και ιδιαίτερα σε μαθητές δευτεροβάθμιας εκπαίδευσης.

Η αξιολόγηση γνώσεων προγραμματισμού και οι δυσκολίες

Τα εισαγωγικά μαθήματα προγραμματισμού δίνουν έμφαση στην επίλυση προβλημάτων και στην απόκτηση γνώσεων και δεξιοτήτων σχετικά με μεθόδους αλγοριθμικής (Τζιμογιάννης, 2005). Διαδικασία περίπλοκη που συχνά οδηγεί σε παρανοήσεις και απογοητεύσεις. Ο Παπαδάκης (2016) στην επισκόπηση του παρουσιάζει τα προβλήματα της διδασκαλίας του προγραμματισμού στους μαθητές της Δευτεροβάθμιας εκπαίδευσης στην Ελλάδα και παραθέτει λύσεις για την αντιμετώπισή τους.

Ιδιαίτερης σημασίας είναι οι δραστηριότητες που χρησιμοποιούν οι εκπαιδευτικοί κατά τη διδασκαλία. Ο εκπαιδευτικός οφείλει να αξιοποιεί κατάλληλες τεχνικές και παραδείγματα ώστε οι μαθητές να αφομοιώσουν αφηρημένες έννοιες, καθώς δεν έχουν πραγματικές αναπαραστάσεις (Bey et al, 2010).

Μεγάλος όγκος ερευνών επιβεβαιώνουν ότι η αξιολόγηση των γνώσεων προγραμματισμού δεν είναι εύκολη ούτε ακριβής. Το χαμηλό ποσοστό επιτυχίας των μαθητών στα μαθήματα προγραμματισμού, μπορεί να εξηγηθεί και από την έλλειψη καλών εργαλείων αξιολόγησης για την ταξινόμηση της δυσκολίας των μαθησιακών δραστηριοτήτων που αφορούν τη σύνταξη ενός προγράμματος. Ανεξάρτητα από τους τρόπους αξιολόγησης εντοπίζονται ορισμένες κοινές πεποιθήσεις που σχετίζονται με την αντίληψη ότι οι γνώσεις των μαθητών δεν μπορούν και δεν πρέπει να αξιολογούνται μόνο με τελική εξέταση. Οι μαθητές πρέπει να αξιολογούνται κατά τη διάρκεια διδασκαλίας του μαθήματος ώστε να κινητοποιούνται αλλά και να παρακολουθούν την πρόοδό και τις δυσκολίες τους (Ivanović et al., 2015):

Για την αξιολόγηση της δυσκολίας δραστηριοτήτων προγραμματισμού, συνήθως χρησιμοποιούνται δύο προσεγγίσεις, των ταξινομιών και των μετρήσεων. Η προσέγγιση των ταξινομιών είναι υποκειμενική, απαιτεί οι εκπαιδευτικοί που δημιουργούν τις δραστηριότητες προγραμματισμού, να είναι έμπειροι σχετικά με την ταξινόμηση των δραστηριοτήτων στα επίπεδα των ταξινομιών. Οι ταξινομίες που χρησιμοποιούνται είναι η αρχική ταξινόμια Bloom, η αναθεωρημένη ταξινόμια Bloom και η ταξινόμια SOLO. Η άλλη προσέγγιση, οι μετρήσεις, είναι μια ποσοτική προσέγγιση. Χρησιμοποιούνται μετρήσεις που προέρχονται από τη λύση που δίνεται από τον εκπαιδευτικό και που μπορεί να είναι πιο απλή από τις λύσεις που δίνουν οι μαθητές. Η λύση είναι ένας ενδεικτικός δείκτης δυσκολίας που βοηθά στην ταξινόμηση ερωτήσεων με βάση τις απαιτήσεις τους και πιθανόν να μην είναι κατάλληλη για ειδικά παραδείγματα προγραμματισμού (όπως η αναδρομή) (Elnaffar, 2016).

Γενικά η αλγοριθμική είναι ένα δύσκολο πεδίο για να διδαχτεί από τον εκπαιδευτικό και πολύπλοκη η αφομοίωση από τον μαθητή. Ο αλγόριθμος, επιπλέον, χαρακτηρίζεται από το πλήθος των λύσεων για ένα πρόβλημα. Αυτό το χαρακτηριστικό αποτελεί την πρώτη δυσκολία, επειδή είναι δύσκολο ο εκπαιδευτικός να προβλέψει όλες τις πιθανές λύσεις ενός προβλήματος προκειμένου να τις ενσωματώσει στη βάση της λύσης του (Bey et al., 2010).

Η δεύτερη δυσκολία είναι άμεσα συνδεδεμένη με την ίδια την αξιολόγηση. Οι ερευνητές θεωρούν ότι η αποτελεσματική διδασκαλία δεν αποδίδει ουσιαστικό ρόλο στην αξιολόγηση ως μέρος της μαθησιακής διαδικασίας. Η αξιολόγηση στην τάξη αποτελεί πηγή ασάφειας μεταξύ του αξιολογητή και του εκπαιδευόμενου, και μερικές φορές μεταξύ των ίδιων των αξιολογητών. Παρ' όλα αυτά, είναι μια διαδικασία ενσωμάτωσης στη διαδικασία μάθησης. Η αξιολόγηση αποτελεί οδηγό για την πρόοδο του εκπαιδευόμενου και παρεμβαίνει στο επίπεδο αλληλεπίδρασης μεταξύ εκπαιδευτικού και εκπαιδευόμενου για τη βελτιστοποίηση της μεταφοράς γνώσεων, δεξιοτήτων και πρακτικών. Έτσι, η σημασία της είναι πρωτεύουσα. Με αυτή την έννοια, έχουν χρησιμοποιηθεί πολλές μέθοδοι και εργαλεία στην αξιολόγηση, τα οποία είναι ανεπαρκή. Αυτή η ανεπάρκεια χαρακτηρίζεται είτε από αναποτελεσματικότητα, από αμφίβολο αποτέλεσμα είτε από μοναδικότητα, δηλαδή δεν μπορεί να εφαρμοστεί σε όλα τα πεδία (Bey et al., 2010).

Η εφαρμογή των ταξινομιών BLOOM και SOLO στον προγραμματισμό

Τόσο η αρχική ταξινόμια Bloom (1956) όσο και η αναθεωρημένη (Anderson, & Krathwohl, 2001) έχουν χρησιμοποιηθεί ως σημείο αναφοράς για έρευνα στον προγραμματισμό. Οι εκπαιδευτικοί που υλοποιούν την ταξινόμηση οφείλουν να γνωρίζουν το πλαίσιο διδασκαλίας του μαθήματος, δηλαδή να διδάσκουν οι ίδιοι που αξιολογούν ώστε να μπορούν να κατηγοριοποιούν τις δραστηριότητες στα κατάλληλα επίπεδα της ταξινόμιας

Η ταξινόμια SOLO των Biggs & Collis (198

2), κατηγοριοποιεί την νοητική δραστηριότητα μέσω των ποιοτικών και ποσοτικών χαρακτηριστικών των δραστηριοτήτων που απαιτείται να υλοποιήσουν οι μαθητές. Αρκετές μελέτες την χρησιμοποιούν στην αξιολόγηση απαντήσεων σε δραστηριότητες προγραμματισμού (Λαδιάς et al, 2018; Lister et al., 2006; Jimoγιannis, 2011).

Η εφαρμογή των δύο ταξινόμιών παρουσιάζει δυσκολίες στη διδασκική πράξη επειδή η ταξινόμηση των δραστηριοτήτων σε επίπεδα εξαρτάται από τα συγκεκριμένα παραδειγματα που διδάσκονται στην τάξη. Από βιβλιογραφική επισκόπηση διαπιστώθηκε ότι η χρήση της ταξινόμιας SOLO εξαρτάται σε μεγάλο βαθμό από το περιεχόμενο του μαθήματος και το βάθος κάλυψης της διδασκίας ύλης (Ζιώγα, 2018). Δεν μπορεί να εφαρμοστεί με συνέπεια από πολλούς εκπαιδευτικούς, συνήθως προσαρμόζεται, παραλείπονται κάποια επίπεδα (πρώτο προδομικό και πέμπτο εκτεταμένης θεώρησης), δημιουργούνται επιμέρους κατηγορίες, ή συνδυάζεται με την ταξινόμια Bloom. Η εφαρμογή της δημιουργεί ασάφειες.

Εμπειρική μελέτη

Το πλαίσιο της έρευνας

Η έρευνα για την αξιολόγηση των γνώσεων προγραμματισμού εστιάζει κυρίως στην τριτοβάθμια εκπαίδευση (Izu et al., 2016; Kiran, & Moudgalya, 2015; Lopez et al., 2008; Selby, 2015; Sheard et al., 2008; Thompson, 2007), στο γυμνάσιο και το δημοτικό (Jimoγιannis, 2011; Mallios, & Vassilakopoulos, 2015; Meerbaum-Salant, et al., 2010). Κατά την αξιολόγηση γνώσεων προγραμματισμού, διαπιστώνεται αμφισβήτηση ως προς την αξιοπιστία στη χρήση των ταξινόμιών Bloom και SOLO. Λαμβάνοντας υπόψη τα παραπάνω, η παρούσα μελέτη διερευνά αφενός το επίπεδο των γνώσεων προγραμματισμού τελειόφοιτων μαθητών Λυκείου και αφετέρου την εφαρμογή του μοντέλου Ιεραρχικής Αξιολόγησης Προγραμματισμού (ΙΑΠ) κατά την αξιολόγηση των γνώσεων των μαθητών.

Στην έρευνα εφαρμόστηκε το μοντέλο ΙΑΠ που προτείνουν οι Μπέλλου & Μικρόπουλος (2008) για την ποιοτική αξιολόγηση των γνώσεων στον προγραμματισμό. Επιλέχθηκε η εφαρμογή του μοντέλου αυτού αφενός επειδή είναι εύκολο για κάθε εκπαιδευτικό που διδάσκει να κατατάξει τις απαντήσεις των μαθητών στα επίπεδα του, αφετέρου επειδή μπορεί με μεγαλύτερη ακρίβεια να προσδιορίσει τις παρανοήσεις και να αξιολογήσει τις δυνατότητες των μαθητών στον προγραμματισμό. Το μοντέλο τροποποιεί την ταξινόμια SOLO, την εξειδικεύει στις προγραμματιστικές γνώσεις και περιλαμβάνει πέντε ιεραρχικά επίπεδα:

- 1^ο επίπεδο, προδομικό: ο μαθητής δεν έχει κατανοήσει το πρόβλημα, δεν γνωρίζει τη σύνταξη των εντολών ή δεν απαντά.
- 2^ο επίπεδο, επιμέρους κατανόησης: ο μαθητής έχει αντιληφθεί μερικώς το πρόβλημα και έχει αποδώσει ορθά γραμματικά και συντακτικά απλές εντολές.
- 3^ο επίπεδο, προσεγγιστικής κατανόησης: ο μαθητής έχει αντιληφθεί το πρόβλημα και έχει δώσει λύση ορθή με απλές εντολές και λογικές δομές.
- 4^ο επίπεδο, συνδυαστικό: ο μαθητής έχει σχεδιάσει δομημένη λύση. Χρησιμοποιεί σωστά εντολές και δομές αλλά επιλέγει την πλέον κατάλληλη.
- 5^ο επίπεδο, εκτεταμένης θεώρησης: ο μαθητής υλοποιεί το βέλτιστο αλγόριθμο για το πρόβλημα και τη συγκεκριμένη γλώσσα.

Δείγμα και διαδικασία

Η έρευνα διεξήχθη κατά το σχολικό έτος 2017-18 στο πλαίσιο του μαθήματος ΑΕΠΠ της Γ' Λυκείου όπου συμμετείχαν 33 μαθητές Θετικών Σπουδών και Οικονομίας - Πληροφορικής μετά από διδασκαλία 12 εβδομάδων. Οι δραστηριότητες ήταν ίδιες για όλους τους μαθητές και αφορούσαν έννοιες σχετικές με τις μεταβλητές, εντολές εκχώρησης, δομές επιλογής και επανάληψης. Η γραπτή αξιολόγηση έγινε στο τέλος του πρώτου τετραμήνου με διάρκεια τρεις διδακτικές ώρες, σε 10 δραστηριότητες, ταξινομημένες με βάση τα επίπεδα της αναθεωρημένης ταξινόμιας Bloom, με δύο δραστηριότητες για κάθε επίπεδο και μία για τα επίπεδα θυμάμαι και δημιουργώ. Οι δραστηριότητες επιλέχθηκαν σύμφωνα με τη διδαχθείσα ύλη και στις δύο ομάδες Προσανατολισμού ενώ εξαιρέθηκαν οι απαντήσεις των μαθητών που δεν απάντησαν σε όλες τις δραστηριότητες.

Οι απαντήσεις των μαθητών αξιολογήθηκαν με την ΙΑΠ από την εκπαιδευτικό που δίδασκε το μάθημα και με τον παραδοσιακό τρόπο βαθμολόγησης από έξι βαθμολογητές, εκπαιδευτικούς Πληροφορικής με εμπειρία από 12 έως 18 έτη. Ο καθένας τους βαθμολόγησε πέντε ή έξι γραπτά, σε εικοσαβάθμια κλίμακα. Κατόπιν, οι βαθμολογίες αντιστοιχήθηκαν με τα επίπεδα ΙΑΠ για να συγκριθούν οι δύο τρόποι αξιολόγησης. Η κλίμακα αντιστοιχισμός που χρησιμοποιήθηκε παρουσιάζεται στον Πίνακα 1. Ορίστηκε με βάση την εμπειρία των εκπαιδευτικών από βαθμολόγηση γραπτών σε επίπεδο εξετάσεων σχολείου. Η αξιολόγηση με την ΙΑΠ βασίστηκε σε υποδειγματικές απαντήσεις ανά ιεραρχικό επίπεδο (Ζιώγα, 2018).

Πίνακας 1. Αντιστοιχία της εικοσαβάθμιας κλίμακας στα επίπεδα ΙΑΠ

| Επίπεδα ΙΑΠ | Βαθμολογία |
|--|---------------|
| 1 ^ο προδομικό | 0 έως και 7 |
| 2 ^ο επιμέρους κατανόησης | 8 έως και 11 |
| 3 ^ο προσεγγιστικής κατανόησης | 12 έως και 16 |
| 4 ^ο συνδυαστικό | 17 έως και 19 |
| 5 ^ο εκτεταμένης θεώρησης | 20 |

Αποτελέσματα και συζήτηση

Γνώσεις προγραμματισμού

Πίνακας 2. Σύγκριση αξιολόγησης μεταξύ ΙΑΠ και παραδοσιακής αξιολόγησης (ΠΑ)

| Επίπεδο Bloom | 1 ^ο προδομικό (%) | | 2 ^ο επιμέρους κατανόησης (%) | | 3 ^ο προσεγγιστικής κατανόησης (%) | | 4 ^ο Συνδυαστικό (%) | | 5 ^ο Εκτεταμένης θεώρησης (%) | |
|---------------|------------------------------|------|---|------|--|------|--------------------------------|------|---|------|
| | ΙΑΠ | ΠΑ | ΙΑΠ | ΠΑ | ΙΑΠ | ΠΑ | ΙΑΠ | ΠΑ | ΙΑΠ | ΠΑ |
| Θυμάμαι | 10.1 | 17.2 | 16.2 | 15.2 | 27.3 | 21.2 | 45.5 | 15.2 | 1.0 | 31.3 |
| Κατανοώ | 24.2 | 41.4 | 7.1 | 3.0 | 18.2 | 8.1 | 48.5 | 5.1 | 2.0 | 42.4 |
| Εφαρμόζω | 18.2 | 32.3 | 9.1 | 7.1 | 12.1 | 0.0 | 60.6 | 1.0 | 0.0 | 59.6 |
| Αναλύω | 24.2 | 48.5 | 27.3 | 9.1 | 21.2 | 15.2 | 24.2 | 15.2 | 3.0 | 12.1 |
| Αξιολογώ | 33.3 | 55.6 | 14.1 | 9.1 | 7.1 | 3.0 | 38.4 | 5.1 | 7.1 | 27.3 |
| Δημιουργώ | 24.2 | 39.4 | 39.4 | 12.1 | 27.3 | 27.3 | 0.0 | 15.2 | 9.1 | 6.1 |

Για το προδρομικό επίπεδο η ανάκληση γνώσεων αποτελεί δεξιότητα στην οποία οι μαθητές σημειώνουν τις χαμηλότερες επιδόσεις (10,1%). Η ανάκληση τύπων δεδομένων και η σύνταξη δομών επανάληψης αποτελούν σημεία δυσκολίας για τους μαθητές. Οι χαμηλότερες επιδόσεις για το δεύτερο επίπεδο ΙΑΠ σχετίζονται με την ικανότητα να κατανοήσουν (7,1%) και να εφαρμόσουν (9,1%) στοιχειώδεις πολύ απλές λειτουργίες ενός προγράμματος. Όταν δίνεται ένα πρόγραμμα, εκχωρούν την ελάχιστη τιμή σε λάθος μεταβλητή. Επίσης, σε δομή επανάληψης «ΟΣΟ» χρησιμοποιούν στη συνθήκη λάθος μεταβλητές και τις αντίστοιχες οριακές τιμές. Ακόμα, διαπιστώνονται λάθη ως προς τη σειρά εκτέλεσης των εντολών αφού κάποιοι μαθητές θεωρούν ότι η επανάληψη προηγείται της αρχικοποίησης μιας μεταβλητής. Η δημιουργία προγράμματος αποτελεί τη μικρότερη δυσκολία για τους μαθητές που μπορούν να κάνουν μόνο απλούς ελέγχους με δομή επιλογής και να εμφανίζουν μηνύματα. Φαίνεται ότι στο επίπεδο της επιμέρους κατανόησης οι μαθητές έχουν τις καλύτερες επιδόσεις στη δημιουργία προγράμματος και τις χειρότερες στην κατανόηση γνωστών διαδικασιών.

Στο τρίτο επίπεδο οι μαθητές παρουσιάζουν τις χαμηλότερες επιδόσεις στην ικανότητα αξιολόγησης (7,1%). Μικρή δυσκολία εντοπίζεται στη μετατροπή μιας δομής επανάληψης σε άλλη με ελάχιστα μικρά συντακτικά λάθη σε αντίθεση με την τροποποίηση δομής επιλογής με αποτέλεσμα να μην ελέγχουν όλες τις περιπτώσεις (ελέγχουν μόνο την περίπτωση της αρνητικής τιμής στον υπολογισμό της απόλυτης τιμής). Στις ικανότητες της απομνημόνευσης και της δημιουργίας παρατηρούνται οι υψηλότερες επιδόσεις (27,3%). Για την απομνημόνευση τα σημεία δυσκολίας ιεραρχικά είναι η σύνταξη των τριών δομών επανάληψης, τα σχήματα στο διάγραμμα ροής και οι τύποι των δεδομένων, έτσι η περιγραφή γίνεται με παραδείγματα. Χαρακτηριστική περίπτωση είναι η εντολή εκχώρησης για τους τύπους των δεδομένων την οποία παρότι οι μαθητές γνωρίζουν, δεν γράφουν παραδείγματα τιμών με τη χρήση της. Για τη δημιουργία προγράμματος οι μαθητές χρησιμοποιούν δομή επανάληψης με συνθήκη που δεν ελέγχει όλες τις απαιτήσεις του προβλήματος δηλαδή τις οριακές τιμές για το σωστό υπολογισμό των ζητούμενων ή χωρίς κατάλληλο έλεγχο με δομή επιλογής για τη σωστή εμφάνιση μηνυμάτων. Στο επίπεδο της προσεγγιστικής κατανόησης οι επιδόσεις των μαθητών είναι υψηλότερες στην απομνημόνευση και τη δημιουργία προγράμματος και χαμηλότερες στην αξιολόγηση.

Για το τέταρτο συνδυαστικό επίπεδο, οι χαμηλότερες επιδόσεις των μαθητών παρατηρούνται στην παραγωγή προγράμματος με πληρότητα. Δεν υπάρχουν μαθητές που χρησιμοποιούν δομή επανάληψης συνδυάζοντας δύο συνθήκες και λογικό τελεστή. Συμπεραίνεται ότι δεν έχουν κατακτήσει την ικανότητα να δημιουργούν προγράμματα που απαιτούν σύνθεση και τα οποία δεν έχουν διδαχθεί στην τάξη. Οι μαθητές έχουν τη δυνατότητα διάσπασης του προβλήματος σε υποπροβλήματα με πληρότητα, δυσκολεύονται όμως να κάνουν έλεγχο εγκυρότητας δεδομένων επειδή απαιτεί εμφωλευμένη δομή. Είναι όμως σε θέση να χρησιμοποιούν εμφωλευμένες δομές επιλογής ώστε να υπολογίζουν σωστά πλήθη και ποσοστά. Οι παραπάνω μαθητές έχουν την ευχέρεια να επιλύουν προβλήματα που απαιτούν διαδικασίες επανάληψης με γνωστό ή άγνωστο πλήθος επαναλήψεων. Τις υψηλότερες επιδόσεις παρουσιάζουν οι μαθητές στην ικανότητα εφαρμογής (60,6%). Μεγαλύτερη δυσκολία εντοπίζεται στη σύνταξη συνθήκης με ΟΣΟ και στα συντακτικά λάθη στην εντολή εκχώρησης του ελάχιστου. Διαπιστώνεται ευχέρεια των μαθητών στο να υπολογίζουν πλήθη, αθροίσματα και ελάχιστα, μέγιστα, σε μια επαναληπτική διαδικασία. Άρα για το συνδυαστικό επίπεδο οι μαθητές παρουσιάζουν τις καλύτερες επιδόσεις στην εφαρμογή γνωστών διαδικασιών σε ένα πρόγραμμα ενώ τις χειρότερες στην δημιουργία.

Το επίπεδο της εκτεταμένης θεώρησης δεν ορίζεται για τις δραστηριότητες που σχετίζονται με το επίπεδο της εφαρμογής καθώς και για ερωτήματα των επιπέδων απομνημόνευσης, κατανόησης και αξιολόγησης. Οπότε για τις δεξιότητες αυτές δεν μπορούν να εξαχθούν

ακριβή και ασφαλή συμπεράσματα. Τις χαμηλότερες επιδόσεις παρουσιάζουν οι μαθητές στην ανάλυση ενός προβλήματος σε υποπροβλήματα (3%) καθώς δεν είναι σε θέση να λάβουν υπόψη όλους τους περιορισμούς του προβλήματος και να γράψουν έτσι τη βέλτιστη λύση που ελέγχει όλα τα ενδεχόμενα. Παρατηρούνται υψηλότερες επιδόσεις κατά τη δημιουργία ενός προγράμματος (9,1%) επειδή οι μαθητές γνωρίζουν πολύ καλά όλες τις δομές επανάληψης και μπορούν να γράψουν την πλέον κατάλληλη.

Τεχνικές αξιολόγησης γνώσεων προγραμματισμού

Ο Πίνακας 2 παρουσιάζει τη σύγκριση των δύο τεχνικών αξιολόγησης. Στο προδομικό επίπεδο, οι μαθητές που δεν μπορούν να ανακαλέσουν γνώσεις προγραμματισμού διαφοροποιούνται κατά μεγάλο βαθμό από τον τρόπο αξιολόγησης. Σε όλα τα επίπεδα Bloom όσον αφορά τους μαθησιακούς στόχους των δραστηριοτήτων παρατηρούνται χαμηλότερες επιδόσεις με την ΙΑΠ. Στο επίπεδο αυτό φαίνεται ότι η ΙΑΠ αποτελεί έναν περισσότερο αναλυτικό και στοχοθετημένο τρόπο αξιολόγησης που περιορίζει κατά μεγάλο βαθμό τα ποσοστά επιδόσεων των μαθητών για όλα τα επίπεδα της ταξινόμιας Bloom.

Για το δεύτερο επίπεδο της επιμέρους κατανόησης, η ανάκτηση γνώσεων είναι ανεξάρτητη από τον τρόπο αξιολόγησης. Σε όλα τα υπόλοιπα επίπεδα της ταξινόμιας Bloom, η βαθμολογία με την ΙΑΠ καταγράφεται υψηλότερη. Για το τρίτο επίπεδο της προσεγγιστικής κατανόησης, οι δύο τρόποι αξιολόγησης ταυτίζονται στη δημιουργία ολοκληρωμένων προγραμμάτων, ενώ στις δεξιότητες της απομνημόνευσης, της κατανόησης, της ανάλυσης και της αξιολόγησης υπερέχει η ιεραρχική αξιολόγηση.

Όταν το πέμπτο επίπεδο της εκτεταμένης θεώρησης δεν ορίζεται για όλα τα ερωτήματα της δραστηριότητας του αντίστοιχου επιπέδου Bloom, τότε θεωρείται ότι μαζί με το τέταρτο συνδυαστικό αποτελούν ένα επίπεδο. Στην περίπτωση αυτή, η ΙΑΠ πλεονεκτεί με μεγάλες ή μικρές διαφορές έναντι της παραδοσιακής αξιολόγησης ή συμπίπτει ανάλογα με την πολυπλοκότητα της δραστηριότητας. Τα δύο αυτά επίπεδα (τέταρτο και πέμπτο) εξετάζονται ξεχωριστά για τις δραστηριότητες της ανάλυσης και της δημιουργίας προγράμματος. Στη διάσπαση προβλήματος η ΙΑΠ δίνει υψηλότερη ή ίδια βαθμολογία με την παραδοσιακή αξιολόγηση.

Συμπεράσματα

Η παρούσα μελέτη διερευνά την αξιολόγηση γνώσεων προγραμματισμού τελειόφοιτων μαθητών Λυκείου σε θέματα που αφορούν τύπους δεδομένων, εντολές εκχώρησης, δομές επιλογής και επανάληψης. Παρουσιάζει επίσης την σύγκριση δύο τεχνικών αξιολόγησης.

Η ΙΑΠ σε σχέση με την παραδοσιακή αξιολόγηση στο προδομικό επίπεδο συνολικά δίνει πιο αναλυτικά αποτελέσματα, αφού μειώνει τις επιδόσεις των μαθητών σε όλα τα επίπεδα της ταξινόμιας Bloom. Για τα επίπεδα επιμέρους και προσεγγιστικής ταξινόμησης παρουσιάζει ταύτιση ή υπεροχή έναντι της παραδοσιακής αξιολόγησης. Δεν μπορούν να εξαχθούν ασφαλή συμπεράσματα για τα επίπεδα συνδυαστικό και εκτεταμένης θεώρησης. Φαίνεται ότι κατά την αξιολόγηση με την ΙΑΠ αναδεικνύονται με κάθε λεπτομέρεια οι δυσκολίες των μαθητών, καθώς ο εκπαιδευτικός έχει πιο ξεκάθαρη εικόνα για το επίπεδο του κάθε μαθητή.

Τα ευρήματα της έρευνας συμφωνούν με μελέτες σχετικά με τις δυσκολίες των μαθητών στη χρήση των δομών επανάληψης (Γρηγοριάδου et al, 2004), στη μετατροπή μιας δομής σε άλλη, και στους μη αριθμητικούς τύπους δεδομένων (Kiran & Moudgalaya, 2015). Παρερμηνείες εντοπίστηκαν σχετικά με τη δημιουργία συνθήκης και την αρχικοποίηση μεταβλητών ως προς την τιμή και τη θέση (Lahtinen et al., 2005). Οι υψηλότερες επιδόσεις καταγράφονται στο συνδυαστικό επίπεδο στην εφαρμογή γνωστών διαδικασιών (>60%), ενώ στο επίπεδο

προσεγγιστικής κατανόησης οι μαθητές σημειώνουν τις υψηλότερες επιδόσεις στο επίπεδο δημιουργίας (27,3%). Η εφαρμογή γνώσεων θεωρείται ευκολότερη για τους μαθητές από τη δημιουργία προγράμματος (Meerbaum-Salant et al, 2010; Lister et al., 2006; Lopez et al, 2008).

Όταν δίνονται δύο δραστηριότητες στο ίδιο επίπεδο Bloom, οι μαθητές τείνουν να παρέχουν απαντήσεις στο ίδιο επίπεδο ΙΑΠ για την κατανόηση και την εφαρμογή, σε μικρότερο για την αξιολόγηση ενώ διαφοροποιούνται στην ανάλυση. Αυτό σχεδόν ταυτίζεται με την έρευνα των Sheard et al. (2008), οι οποίοι κατέληξαν στο συμπέρασμα ότι οι φοιτητές είναι σχετικά συνεπείς στο επίπεδο SOLO των απαντήσεων τους.

Στο επίπεδο επιμέρους κατανόησης, έδειξαν κακή αντίληψη της λειτουργίας της επανάληψης, τόσο για απλούς όσο και για εμφωλευμένους βρόχους (Izu et al., 2016). Ενώ επιβεβαιώνεται η απόψη των Petersen et al. (2011), ότι το επίπεδο εξαρτάται από τις διδασκικές δραστηριότητες που καλύπτονται κατά τη διδασκαλία.

Στην έρευνα των Izu et al. (2016) παραλείπεται η κατηγορία της εκτεταμένης θεώρησης επειδή δεν είναι δυνατόν να προσδιοριστεί εάν οι μαθητές μπορούν να επεκτείνουν τις τρέχουσες γνώσεις τους. Στη συγκεκριμένη έρευνα μπορεί να εφαρμοστεί μόνο για τα επίπεδα αναλύω και δημιουργώ της ταξινόμιας Bloom και όχι στα υπόλοιπα επίπεδα επειδή οι δραστηριότητες είναι πολύ απλές. Τα επίπεδα αυτά ίσως θα πρέπει να εξετάζονται ως ένα επίπεδο και να διαχωρίζονται μόνο στην περίπτωση της ανάλυσης και της δημιουργίας.

Στην έρευνα υπάρχουν τρεις βασικοί περιορισμοί. Ο μικρός αριθμός του δείγματος που είχε διδαχθεί από το ίδιο εκπαιδευτικό τις ίδιες δραστηριότητες και την διδασκία ύλη με την ίδια σειρά. Άλλος περιορισμός αφορά τους βαθμολογητές οι οποίοι δεν είχαν προσυμφωνήσει ώστε να διαμορφώσουν κοινό κριτήριο βαθμολόγησης. Ο τρίτος περιορισμός αφορά την αντιστοιχία της κλίμακας βαθμολόγησης με τα επίπεδα ΙΑΠ η οποία δεν είχε ελεγχθεί με δοκιμή. Η κλίμακα όπως έχει οριστεί ανταποκρίνεται σωστά. Τα επίπεδα της ΙΑΠ ενδεικνύονται είναι να είναι τέσσερα αφενός επειδή για όλα δεν ορίζονται τα επίπεδα Bloom και αφετέρου ο βαθμός 20 αντιστοιχεί για τους βαθμολογητές και στο τέταρτο και στο πέμπτο επίπεδο.

Οι μελλοντικές έρευνες πρέπει να περιλαμβάνουν λεπτομερέστερη εξέταση των σφαλμάτων και ακριβέστερη προσαρμογή της ΙΑΠ στην αξιολόγηση προγραμμάτων. Οι εκπαιδευτικοί μπορούν να εφαρμόσουν αυτό το μοντέλο αξιολόγησης στην αρχή κάθε τετράμηνου ώστε να αναγνωρίσουν τα λανθασμένα νοητικά μοντέλα, και κατόπιν να προσαρμόσουν την διδασκαλία τους ανάλογα με τις ανάγκες των μαθητών. Τέλος, ελπίζουμε να μπορέσουμε να επεκτείνουμε τα αποτελέσματα της έρευνας για την κατασκευή τυποποιημένων τεστ για δεξιότητες προγραμματισμού.

Τα ευρήματα της εργασίας δείχνουν ότι η εφαρμογή της ΙΑΠ μπορεί να ενισχύσει τις προοπτικές των εκπαιδευτικών αλλά και των ερευνητών για αλγοριθμικές εκτιμήσεις στον εισαγωγικό προγραμματισμό, να εμβαθύνει και να διευρύνει την αξιολόγηση των δραστηριοτήτων των μαθητών και να βοηθήσει να εντοπιστούν οι δυσκολίες στο σχεδιασμό προγραμμάτων, καθώς και τη βελτίωση τους.

Αναφορές

- Γρηγοριάδου, Μ., Γόγουλου, Α., & Γουλή, Ε. (2004). Μαθησιακές Δυσκολίες στις Επαναληπτικές Δομές. In *4ο Συνέδριο ΕΤΠΕ* (pp. 535-537). Πανεπιστήμιο Αθηνών.
- Ζιώγα, Σ. (2018). *Το μοντέλο της Ιεραρχικής Αξιολόγησης Γνώσεων Προγραμματισμού: μια εμπειρική μελέτη*. Μεταπτυχιακή Διπλωματική Εργασία. ΠΤΔΕ, Πανεπιστήμιο Ιωαννίνων.
- Κόμης, Β. (2001). Μελέτη βασικών εννοιών του Προγραμματισμού στο πλαίσιο μιας Οικοδομησιακής διδασκικής προσέγγισης. *Themes in Education 2*, (τομ. 2-3), 243-270.
- Λαδιάς, Δ., Μικρόπουλος, Α., Πλεσιώτης, Η., & Λαδιάς, Α., (2018). Εφαρμογή της ταξινόμιας SOLO στην αξιολόγηση της δόμησης των εντολών σε περιβάλλοντα οπτικού προγραμματισμού. *Έρκυνα Επιθεώρηση Εκπαιδευτικών-Επιστημονικών Θεμάτων* (15ο) pp: 43-52.

- Μπέλλου, Ι., & Μικρόπουλος, Τ. (2008). Μέθοδος για την Ιεραρχική Αξιολόγηση Γνώσεων Προγραμματισμού. *4ο Πανελλήνιο Συνέδριο Διδακτική Της Πληροφορικής*, 111-120.
- Παπαδάκης, Σ. (2016). Η παραδοσιακή ή κλασική προσέγγιση στην διδασκαλία του Προγραμματισμού. Προβλήματα και λύσεις. *10ο Πανελλήνιο Συνέδριο Καθηγητών Πληροφορικής*.
- Τζιμογιάννης, Α. (2005). Προς ένα Παιδαγωγικό Πλαίσιο Διδασκαλίας του Προγραμματισμού στη Δευτεροβάθμια Εκπαίδευση. *3ο Πανελλήνιο Συνέδριο Διδακτική της Πληροφορικής* (pp. 99-111). Πανεπιστήμιο Πελοποννήσου, Κόρινθος.
- Anderson, L. W. & Krathwohl, D. R. (Eds.) (2001). *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. MA: Allyn & Bacon.
- Bennedsen, J., & Caspersen, M. E. (2012). Persistence of elementary programming skills. *Computer Science Education*, 22(2).
- Bey, A., Bensebaa, T., & Benselem, H. (2010). EASEL : Evaluation of Algorithmic Skills in Environment Learning, 4(6), 64-67. *World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering*
- Biggs, J., & Collis, K. (1982). *Evaluating the quality of learning: The SOLO taxonomy (Structure of the Observed Learning Outcome)*. New York, Academic Press.
- Bloom, B. (1956). *Taxonomy of Educational Objectives. Handbook 1: Cognitive Domain*. New York: David McKay.
- Elnaffar, S. (2016). Using software metrics to predict the difficulty of code writing questions. *IEEE Global Engineering Education Conference, EDUCON*, 10-13-April (April 2016), 513-518.
- Ivanović, M., Xinogalos, S., Pitner, T., & Savić, M. (2015). Different Aspects of Delivering Programming Courses: Multinational Experiences. *Proceedings of the 7th Balkan Conference on Informatics Conference*, 37:1-37:7.
- Izu, C., Weerasinghe, A., & Pope, C. (2016). A Study of Code Design Skills in Novice Programmers using the SOLO taxonomy. *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, 251-259.
- Jimoyiannis, A. (2011). Using SOLO taxonomy to explore students' mental models of the programming variable and the assignment statement. *Themes in Science & Technology Education*, 4(2), 53-74, 2011
- Kiran, E. L. N., & Moudgalya, K. M. (2015). Evaluation of programming competency using student error patterns. *International Conference on Learning and Teaching in Computing and Engineering*, 34-41..
- Lahtinen, E., Ala-Mutka, K., & Järvinen, H.-M. (2005). A study of the difficulties of novice programmers. *ACM SIGCSE Bulletin*, 37(3).
- Lister, R., Simon, B., Thompson, E., Whalley, J. L., & Prasad, C. (2006). Not Seeing the Forest for the Trees : Novice Programmers and the SOLO Taxonomy. *ACM SIGCSE Bulletin*, 38, 118-122.
- Lopez, M., Whalley, J., Robbins, P., & Lister, R. (2008). Relationships between reading, tracing and writing skills in introductory programming. *ICER '08*, 101-112.
- Mallios, N., & Vassilakopoulos, M. (2015). Evaluating Students Programming Skill Behaviour and Personalizing Their Computer Learning Environment Using "the Hour of Code" Paradigm, *International Association for Development of the Information Society*.131-135.
- Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2010). Learning computer science concepts with scratch. *ICER '10*, 69-76.
- Petersen, A., Craig, M., & Zingaro, D. (2011). Reviewing CS1 exam question content. *42nd ACM Technical Symposium on Computer Science Education*, 631-636.
- Selby, C. C. (2015). Relationships: Computational Thinking, Pedagogy of Programing, and Bloom's Taxonomy. *Proceedings of the Workshop in Primary and Secondary Computing Education*, 80-87.
- Sheard, J., Carbone, A., Lister, R., Simon, B., Thompson, E., & Whalley, J. L. (2008). Going SOLO to assess novice programmers. *ACM SIGCSE Bulletin*, 40(3), 209.
- Thompson, E. (2007). Holistic assessment criteria - Applying SOLO to programming projects. *Conferences in Research and Practice in Information Technology Series*, 66, 155-162.
- Whalley, J., Lister, R., Thompson, E., Clear, T., Robbins, P., Ajith Kumar, P. K., & Prasad, C. (2006). An Australasian study of reading and comprehension skills in novice programmers, using the bloom and SOLO taxonomies. *Conferences in Research and Practice in Information Technology Series*, 52, 243-252.