

# Διερεύνηση των αντιλήψεων μαθητών Γυμνασίου για τις γλώσσες προγραμματισμού

Ευριπίδης Βραχνός

Ζάννειο Πειραματικό Γυμνάσιο Πειραιά  
evrachnos@gmail.com

## Περίληψη

Στη δευτεροβάθμια εκπαίδευση η εισαγωγή των μαθητών στον προγραμματισμό υπολογιστών γίνεται συνήθως με γλώσσες οι οποίες βασίζονται στη γεωμετρία της χελώνας, όπως είναι το Scratch και η Logo. Με την αλματώδη ανάπτυξη των εφαρμογών του παγκόσμιου ιστού και των κινητών συσκευών έχει παρουσιαστεί η ανάγκη για τη μετάβαση σε γλώσσες προγραμματισμού με πολύ απότομη καμπύλη μάθησης. Οι γλώσσες αυτές επιτρέπουν στους μαθητές να αναπτύξουν πολύ σύντομα εφαρμογές που θα μπορούν να εκτελούνται στον παγκόσμιο ιστό ή στο κινητό τους τηλέφωνο και ανταποκρίνονται καλύτερα στα ενδιαφέροντά τους. Ωστόσο δεν υπάρχει κάποια έρευνα σχετικά με τις αντιλήψεις που έχουν σχηματίσει οι μαθητές για όλες αυτές τις διαφορετικές προσεγγίσεις, διότι ελάχιστοι μαθητές έρχονται σε επαφή με περισσότερες από δυο γλώσσες προγραμματισμού στη σχολική της ζωή. Οι μαθητές του ομίλου Αλγοριθμικής στο Ζάννειο Γυμνάσιο έχουν αναπτύξει εφαρμογές σε App Inventor, Python και C++. Επίσης έχουν προγραμματίσει Scratch και σε Logo στο πλαίσιο του μαθημάτος πληροφορικής του Γυμνασίου. Έχει σημασία λοιπόν η διερεύνηση των αντιλήψεων που έχουν για όλες αυτές τις γλώσσες. Στην έρευνα συμμετείχαν 16 μαθητές οι οποίοι κλήθηκαν να απαντήσουν σε μια σειρά από ερωτήσεις για τις γλώσσες προγραμματισμού με τις οποίες έχουν ασχοληθεί.

**Λέξεις κλειδιά:** όμιλος αριστείας, γλώσσες προγραμματισμού, έρευνα

## 1. Εισαγωγή

Οι μαθητές στην δευτεροβάθμια εκπαίδευση δεν διδάσκονται πολλές γλώσσες προγραμματισμού. Στο γυμνάσιο γίνεται μια εισαγωγή στον προγραμματισμό με το περιβάλλον Scratch ή τη γλώσσα Logo. Και στις δυο περιπτώσεις το προγραμματιστικό παράδειγμα είναι αυτό της γεωμετρίας της χελώνας. Στο Λύκειο ακολουθείται το διαδικασιακό παράδειγμα με την ψευδογλώσσα και στο Επαγγελματικό Λύκειο τα τελευταία χρόνια έχουν εισαχθεί πιο σύγχρονες γλώσσες προγραμματισμού όπως Python και η Java.

Στο Ζάννειο Γυμνάσιο τα τελευταία χρόνια λειτουργεί ο όμιλος Αλγοριθμικής στον οποίο οι μαθητές ασχολούνται μεταξύ άλλων με διάφορες γλώσσες προγραμματισμού. Για αυτό θεωρήσαμε ότι θα είχε ενδιαφέρον να μελετήσουμε τις αντιλήψεις των μαθητών για όλες τις γλώσσες προγραμματισμού με τις οποίες έχουν

ασχοληθεί έως τώρα. Οι μαθητές του ομίλου έχουν προγραμματίσει σε Logo και Scratch στο μάθημα της πληροφορικής και σε C++, Python και App Inventor στον όμιλο.

Τα αποτελέσματα έδειξαν ότι οι μαθητές δείχνουν μια προτίμηση στην Python και το App Inventor έναντι της C++ και της Logo. Αυτό ήταν αναμενόμενο και για τις δυο γλώσσες. Από τη μια η Python έχει πολύ εύκολη σύνταξη και αρκετά απότομη καμπύλη μάθησης σε σχέση με τη C++ ενώ το App Inventor προσφέρει στους μαθητές τη δυνατότητα ανάπτυξης εφαρμογών που θα εκτελούνται στο κινητό τους, κάτι αρκετά ελκυστικό για την ηλικία τους. Ωστόσο θα πρέπει να έχουμε υπόψη μας κατά την ερμηνεία των αποτελεσμάτων της έρευνας ότι οι μαθητές του δείγματος συμμετείχαν σε έναν όμιλο πληροφορικής άρα έχουν ιδιαίτερο ενδιαφέρον για τον προγραμματισμό.

## **2. Όμιλος Αλγοριθμικής**

Ο όμιλος Αλγοριθμικής (Βραχνός & Ντούσκα, 2016) λειτουργεί στο Ζάννειο πειραματικό γυμνάσιο από το 2014, μια φορά την εβδομάδα μετά το σχολείο για δυο ώρες, και γίνεται πάντα στο εργαστήριο πληροφορικής. Έχει ως αντικείμενο την επιστήμη της πληροφορικής μέσα από την επίλυση αλγοριθμικών προβλημάτων και την υλοποίησή τους σε διάφορες γλώσσες προγραμματισμού.

Ο πρώτος στόχος του ομίλου είναι η ενασχόληση των μαθητών με προβλήματα τα οποία επιδέχονται αλγοριθμική λύση, όπως είναι ο κύβος του Rubik και το πρόβλημα των 8 βασιλισσών. Ο δεύτερος στόχος μας ήταν να γνωρίσουν οι μαθητές μια σύγχρονη γλώσσα προγραμματισμού που χρησιμοποιείται ευρέως στη βιομηχανία λογισμικού αλλά και στην έρευνα. Έτσι καταλήξαμε σε δυο γλώσσες τη C++ και την Python. Ο τρίτος στόχος μας ήταν να παράξουν οι μαθητές χρήσιμες εφαρμογές, να γίνουν οι ίδιοι/ες δημιουργοί εφαρμογών για κινητά τηλέφωνα ή ταμπλέτες, μέσα από την ανάπτυξη εφαρμογών για Android. Τον πρώτο χρόνο ασχοληθήκαμε με το App Inventor και τη C++ και τον 2<sup>ο</sup> με την C++ και τη γλώσσα Python.

## **3. Το περιβάλλον προγραμματισμού Scratch**

Ο προγραμματισμός με πλακίδια είναι αρκετά διαδεδομένος στη δευτεροβάθμια εκπαίδευση και ειδικότερα στις πολύ μικρές ηλικίες, με πιο δημοφιλές προγραμματιστικό περιβάλλον το Scratch. Ενώ ο προγραμματισμός με Scratch παρουσιάζει σημαντικά διδακτικά οφέλη στις μικρές ηλικίες (Meerbaum-Salant, Armoni & Ben-Ari, 2010), έχει και ένα σοβαρό μειονέκτημα. Μετά τον αρχικό ενθουσιασμό οι μαθητές αντιλαμβάνονται ότι δεν προγραμματίζουν σε μια “πραγματική” γλώσσα προγραμματισμού οπότε το ενδιαφέρον κάποιων μειώνεται αρκετά. Ο Lewis (2010) κατέληξε στο συμπέρασμα ότι οι μαθητές πρωτοβάθμιας και δευτεροβάθμιας εκπαίδευσης που προγραμματίζουν σε γλώσσες όπου χρειάζεται να

γράφουν κώδικα, όπως είναι η Logo, παρουσιάζουν μεγαλύτερη αυτοπεποίθηση στην επίλυση προβλημάτων από αυτούς που προγραμματίζουν σε Scratch.

#### **4. Η γλώσσα προγραμματισμού Python**

Αρκετές έρευνες έχουν γίνει σχετικά με την επιλογή της γλώσσας για ένα εισαγωγικό μάθημα προγραμματισμού (Jayal et. al., 2011; Kaplan, 2010; Koulouri, 2014). Στις περισσότερες από αυτές συγκρίνονται οι δυο επικρατέστερες τάσεις, η πιο παλιά που υποστηρίζει την εισαγωγή της Java και η σύγχρονη που υιοθετεί την εισαγωγή της Python. Στις περισσότερες περιπτώσεις τα αποτελέσματα είναι αρκετά θετικά για την Python κυρίως όσον αφορά την απόδοση και τα ποσοστά επιτυχίας των φοιτητών στα μαθήματα αυτά.

Ένα πρόγραμμα σε Python δεν απέχει πολύ από την περιγραφή ενός αλγορίθμου σε ψευδογλώσσα, αφού η σύνταξή του είναι εξαιρετικά απλή. Δεν υπάρχει τμήμα δήλωσης μεταβλητών αφού η γλώσσα χρησιμοποιεί ένα δυναμικό σύστημα τύπων το οποίο σε συνδυασμό με τη χρήση του διερμηνευτή, διευκολύνει τον πειραματισμό των μαθητών. Έτσι μπορούμε να ορίσουμε μια συνάρτηση η οποία μπορεί να χρησιμοποιηθεί για οποιοδήποτε τύπο της γλώσσας, δεδομένου ότι οι πράξεις που περιέχει ορίζονται για μεταβλητές του τύπου αυτού, όπως φαίνεται παρακάτω:

Ένα άλλο πλεονέκτημα της Python είναι η υποστήριξη τριών διαφορετικών προγραμματιστικών υποδειγμάτων, του διαδικασιακού, του αντικειμενοστρεφούς και του συναρτησιακού, κάτι που δίνει πολλές επιλογές στον καθηγητή. Επίσης η Python έχει πολύ μεγάλη κοινότητα εκπαιδευτικών και προγραμματιστών, οι οποίοι διαθέτουν αρκετό υλικό (βιβλία, ασκήσεις, φύλλα εργασίας, tutorials, σημειώσεις) στο διαδίκτυο.

Συμπερασματικά η Python φαίνεται σαν μια εξαιρετική επιλογή για την εκπαίδευση και έχει υιοθετηθεί από πολλά πανεπιστήμια κυρίως στην Αμερική (Goldwasser & Letscher, 2015; Guo, 2014). Τα τελευταία χρόνια εισάγεται και στη δευτεροβάθμια εκπαίδευση.

#### **5. Ανάπτυξη Εφαρμογών με το App Inventor**

Το περιβάλλον προγραμματισμού App Inventor που χρησιμοποιείται από αρκετούς εκπαιδευτικούς κυρίως στο μάθημα Εφαρμογές Πληροφορικής της Α' Λυκείου, στηρίζεται, σε προγραμματισμό με πλακίδια (blocks) κάτι πολύ οικείο σε μαθητές γυμνασίου λόγω του Scratch (Chen & Huang, 2017). Η αξιοποίηση περιβαλλόντων προγραμματισμού με πλακίδια ως σκαλωσιά μάθησης για τη μετάβαση σε μια πραγματική γλώσσα προγραμματισμού υιοθετείται από πολλούς ερευνητές (Gray, Abelson, Wolber & Friend, 2012; Dorling, 2015).

Τον πρώτο χρόνο οι μαθητές σχεδίασαν και υλοποίησαν εφαρμογές με το App Inventor και στη συνέχεια έγινε η μετάβαση στη C++, κάτι ασυνήθιστο στη βιβλιογραφία αφού στις περισσότερες περιπτώσεις το App Inventor προετοιμάζει

τους μαθητές για την εισαγωγή στην Java, η οποία χρησιμοποιείται για την ανάπτυξη εφαρμογών για την πλατφόρμα Android. Ωστόσο υπάρχουν περιπτώσεις που αντί για τη Java επιλέγεται η C++ (Mishra, Balan, Iyer, & Murthy, 2014).

## 6. Προγραμματίζοντας σε C++

Για την επιλογή της C++ έπαιξε σημαντικό ρόλο το ενδιαφέρον που έδειξαν αρκετοί μαθητές για συμμετοχή στον πανελλήνιο διαγωνισμό πληροφορικής.

Επιλέχθηκε ένα υποσύνολο της γλώσσας έτσι ώστε να ελαχιστοποιηθεί ο συντακτικός θόρυβος και να αποφευχθούν οι προγραμματιστικές δομές οι οποίες δημιουργούν σοβαρές δυσκολίες στους μαθητές όπως προκύπτει από τη βιβλιογραφία. Τέτοια αντικείμενα είναι για παράδειγμα οι δείκτες (pointers) (Adcock et al., 2007, Craig and Petersen, 2016). Επίσης δεν έγινε καμία παρουσίαση αντικειμενοστρεφών εννοιών όπως κλάση, κληρονομικότητα κλπ. αφού ο σκοπός του ομίλου είναι η επίλυση αλγοριθμικών προβλημάτων και όχι η μοντελοποίηση του προβλήματος υπό τη σκοπιά της τεχνολογίας λογισμικού.

Όσον αφορά τους πίνακες χρησιμοποιήσαμε πίνακες VLA (Variable Length Array) που επιτρέπουν μια δήλωση της μορφής:

```
cin >> size;           # εισαγωγή από το πληκτρολόγιο του μεγέθους του πίνακα
int array[ size ];    # δημιουργία πίνακα ακεραίων size θέσεων.
```

Οπότε δεν χρειάστηκε να μιλήσουμε καθόλου για δείκτες. Επίσης η βιβλιοθήκη STL είναι χρήσιμη γιατί μας παρέχει διάφορες δομές δεδομένων όπως είναι η στοιβία και η ουρά και αλγορίθμους όπως η ταξινόμηση.

Οι μαθητές υλοποίησαν στην C++ ανά ζεύγη (pair programming) διάφορα έργα. Αυτό που παρουσίασε μεγαλύτερο ενδιαφέρον ήταν το παιχνίδι Πέτρα – Ψαλίδι – Χαρτί. Η προσέγγιση που ακολουθήθηκε ήταν η βαθμιαία ανάπτυξη του παιχνιδιού με συνεχείς επεκτάσεις, κάθε μια από τις οποίες υλοποιούσε και έναν συγκεκριμένο διδακτικό στόχο.

## 7. Μεθοδολογία της Έρευνας

Η έρευνα που παρουσιάζουμε σε αυτή την εργασία υλοποιήθηκε στο τέλος του σχολικού έτους 2016-2017. Ο βασικός στόχος της έρευνας ήταν να διερευνήσουμε τις προτιμήσεις τις στάσεις και τις αντιλήψεις των μαθητών σχετικά με τις γλώσσες προγραμματισμού με τις οποίες έχουν ασχοληθεί. Τα ερευνητικά ερωτήματα ήταν τα εξής:

- Ποια γλώσσα προγραμματισμού ή περιβάλλον προτιμούν οι μαθητές;
- Ποιο περιβάλλον ή γλώσσα προγραμματισμού τους δυσκόλεψε περισσότερο;

- Τι θα έπρεπε κατά τη γνώμη τους να διδάσκεται στο σχολείο;

Στην έρευνα συμμετείχαν 16 μαθητές των δυο τελευταίων τάξεων του γυμνασίου οι οποίοι είχαν παρακολουθήσει τον όμιλο αλγοριθμικής τα τελευταία δυο χρόνια. Το εργαλείο της έρευνας ήταν ένα ερωτηματολόγιο μιας σελίδας με 5 ερωτήσεις κλειστού τύπου. Δόθηκε στους μαθητές την ώρα του ομίλου και η συμπλήρωσή του, τους πήρε κατά μέσο όρο 15 λεπτά.

## 8. Ανάλυση των αποτελεσμάτων

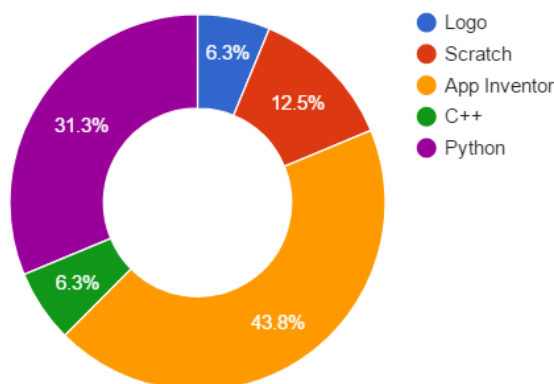
Παρακάτω δίνουμε τα συγκεντρωτικά αποτελέσματα των απαντήσεων των μαθητών για κάθε ερώτηση.

### Ερώτηση 1

*Ποια γλώσσα ή περιβάλλον προγραμματισμού προτιμάτε και γιατί;*

Οι περισσότεροι μαθητές προτίμησαν το App Inventor. Στην αιτιολόγησή τους όμως δεν σημείωσαν κάποια προγραμματιστικά χαρακτηριστικά ή ευκολίες του περιβάλλοντος αλλά μόνο τη δυνατότητα ανάπτυξης εφαρμογών για κινητά. Η ιδέα ότι μπορούν να αναπτύξουν τις δικές τους εφαρμογές και να τις φορτώσουν στο κινητό τους ή να τις μοιραστούν με τους συμμαθητές τους τους ενθουσίασε πολύ. Δεύτερη στις προτιμήσεις των μαθητών ήρθε η Python. Με βάση τα λεγόμενά τους τους έκανε εντύπωση η απλή σύνταξή της και η ευκολία επεξεργασίας συμβολοσειρών.

**Ποια γλώσσα ή περιβάλλον προγραμματισμού προτιμάτε;**



**Εικόνα 1.** Ποια γλώσσα προτιμούν οι μαθητές

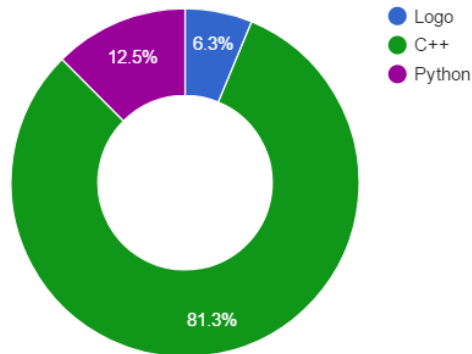
Εκτός από την οικειότητα που αισθάνονται με το συγκεκριμένο περιβάλλον προγραμματισμού, είναι φανερό ότι ο οπτικός προγραμματισμός διευκολύνει αρκετά

τους μαθητές από ότι ο παραδοσιακός προγραμματισμός στον οποίο οι μαθήτριες πρέπει να πληκτρολογήσουν κώδικα (Meerbaum-Salant, Armoni, & Ben-Ari, 2010; Uludag, Karakus, & Turner, 2011).

## Ερώτηση 2

*Ποια γλώσσα ή περιβάλλον προγραμματισμού σας δυσκόλεψε περισσότερο;*

**Ποια γλώσσα προγραμματισμού σας δυσκόλεψε περισσότερο;**



**Εικόνα 2.** Ποια γλώσσα δυσκόλεψε περισσότερο τους μαθητές

Εδώ η συντριπτική υπεροχή της C++ ήταν αναμενόμενη αφού πρόκειται για μια γλώσσα με πιο δύσκολη σύνταξη από την Python, που δεν παρέχει διερμηνευτή άρα πρέπει να γράφεις όλο το πρόγραμμα και να είναι συντακτικά σωστό πριν το εκτελέσεις. Ο προγραμματιστής δεν έχει την πολυτέλεια να δοκιμάσει και να πειραματιστεί με μεμονωμένες εντολές. Αυτό το χαρακτηριστικό αποτελεί σημαντικό πλεονέκτημα της Python σε ένα εισαγωγικό μάθημα προγραμματισμού ειδικά όταν απευθύνεται σε μικρές ηλικίες όπως αυτές των μαθητών Γυμνασίου.

## Ερώτηση 3

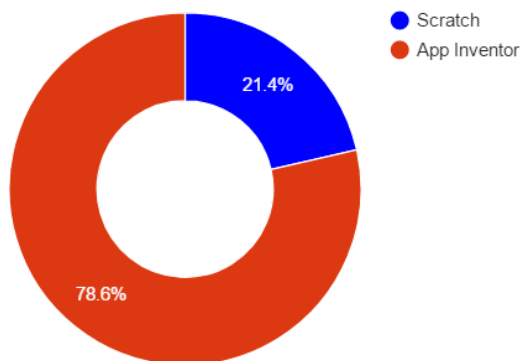
*Ποιο περιβάλλον οπτικού προγραμματισμού προτιμάτε καλύτερα;*

Και εδώ η συντριπτική πλειοψηφία επέλεξε το App Inventor. Παρουσιάζει ενδιαφέρον το γεγονός ότι οι μαθητές που προηγουμένως είχαν επιλέξει μη οπτικές γλώσσες όπως η Python και η C++ μεταξύ Scratch και App Inventor προτιμούν το 2°. Η αιτιολόγηση που έδωσαν ήταν ότι με το App Inventor μπορούν να αναπτύξουν μια αληθινή εφαρμογή που μπορεί να είναι χρήσιμη για τους ίδιους αλλά και τους συμμαθητές τους.

Οι μαθητές που προτίμησαν το Scratch δήλωσαν ότι το App Inventor τους δυσκόλεψε σε κάποια σημεία αφού έπρεπε να μάθουν πολλά νέα πράγματα. Επίσης θεωρούν ότι είναι πιο δύσκολο στη χρήση από ότι το Scratch. Εδώ αρκετοί μαθητές έκαναν λόγο

και για ένα μειονέκτημα του App Inventor την ταχύτητα απόκρισης που σε κάποιες περιπτώσεις είναι χαμηλή. Αυτό οφείλεται στον εξομοιωτή του ο οποίος απαιτεί πολλούς υπολογιστικούς πόρους.

**Ποιο περιβάλλον οπτικού προγραμματισμού προτιμάτε καλύτερα;**

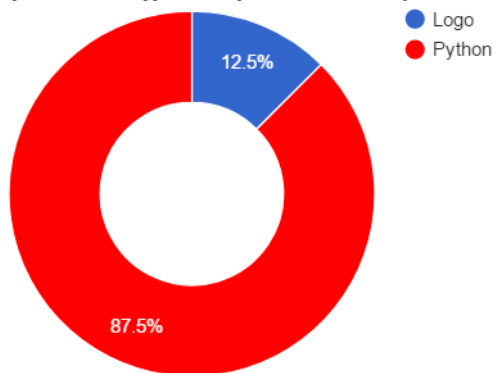


*Εικόνα 3. Ποιο περιβάλλον οπτικού προγραμματισμού προτιμούν οι μαθητές*

#### Ερώτηση 4

*Ποια γλώσσα προγραμματισμού θεωρείται κατάλληλη, από αυτές που γράφεται κώδικας, για να διδαχθεί στην εκπαίδευση; (Επιλέξτε μεταξύ Python, C++, Logo)*

**Ποια γλώσσα προγραμματισμού θεωρείται κατάλληλη, από αυτές που γράφεται κώδικας, για να διδαχθεί στην εκπαίδευση;**



*Εικόνα 5. Ποια γλώσσα είναι κατάλληλη για την εκπαίδευση*

Οι περισσότεροι μαθητές επέλεξαν τη γλώσσα Python για τους ίδιους λόγους που αναφέραμε προηγουμένως :

- Απλότητα σύνταξης εντολών
- Δυνατότητα πειραματισμού μεμονωμένων εντολών που παρέχεται από τον διερμηνευτή
- Λιγότερα συντακτικά λάθη, π.χ. δεν χρειάζεται κάθε εντολή να τελειώνει με ;
- Δεν υπάρχει τμήμα δήλωσης μεταβλητών
- Ευκολία ανάπτυξης εφαρμογών επεξεργασίας αλφαριθμητικών (π.χ. αλγόριθμος κρυπτογράφησης του Καίσαρα)

Επίσης κάποιοι μαθητές έγραψαν ότι με “*με αυτή τη γλώσσα σου δίνεται η εντύπωση ότι προγραμματίζεις πραγματικά*”.

### Ερώτηση 5

*Αναφέρετε τα πλεονεκτήματα και τα μειονεκτήματα της γλώσσας Python σε σχέση με τη C++ . Ποια γλώσσα από τις δυο προτιμάτε;*

Οι περισσότεροι μαθητές δεν ανέφεραν κανένα μειονέκτημα της Python εκτός από δυο που έγραψαν ότι η Python είναι πολύ πιο αργή από την C++. Οι συγκεκριμένοι μαθητές είχαν γράψει αρκετό κώδικα σε C++ και είχαν προκριθεί στην 2<sup>η</sup> φάση του διαγωνισμού πληροφορικής. Είχαν αποκτήσει μεγάλη εξοικείωση με τη γλώσσα και ήταν εξαρχής αρνητικοί στη μετάβαση στην Python. Κάποιοι μαθητές έγραψαν ότι τους δυσκόλεψαν αρκετά οι εσοχές που έπρεπε να αφήνουν για κάθε εμφωλευμένο μπλοκ εντολών. Από την άλλη τα πλεονεκτήματα της Python που ανέφεραν οι μαθητές σε σχέση με τη C++ συνοψίζονται στα παρακάτω χαρακτηριστικά σχόλια:

*“Δεν χρειάζεται δήλωση μεταβλητών”*

*“Ο διερμηνευτής με βοήθησε πολύ στην αρχή να καταλάβω μόνος μου πως δουλεύουν κάποιες εντολές.”*

*“Η Python είναι πιο απλή.”*

## 9. Συμπεράσματα

Οι μαθητές του ομίλου Αλγοριθμικής στο Ζάννειο Γυμνάσιο έχουν αναπτύξει εφαρμογές σε App Inventor, Python και C++. Επίσης έχουν προγραμματίσει Scratch και σε Logo στο πλαίσιο του μαθημάτος πληροφορικής του Γυμνασίου. Έχει σημασία λοιπόν η διερεύνηση των αντιλήψεων που έχουν για όλες αυτές τις γλώσσες.

Η έρευνα πεδίου έλαβε χώρα στο εργαστήριο πληροφορικής σε 16 μαθητές του ομίλου Αλγοριθμικής οι οποίοι είχαν αναπτύξει εφαρμογές σε App Inventor, Python, C++, Logo και Scratch, στους οποίους δόθηκε ένα ερωτηματολόγιο με ερωτήσεις κλειστού τύπου.

Οι μαθητές έδειξαν ιδιαίτερη προτίμηση στη γλώσσα Python λόγω της απλότητάς της και στο περιβάλλον App Inventor, επειδή δίνει τη δυνατότητα ανάπτυξης μιας



εφαρμογής την οποία μπορούν να φορτώσουν στο κινητό τους. Η μετατόπιση στον οπτικό προγραμματισμό ή σε πιο απτές και χρήσιμες εφαρμογές με παιγνιώδη χαρακτήρα προσφέρει ένα σημαντικό κίνητρο στους μαθητές για να ασχοληθούν.

Τέλος η συντριπτική πλειοψηφία των μαθητών πιστεύουν ότι η γλώσσα προγραμματισμού Python και το περιβάλλον προγραμματισμού App Inventor θα πρέπει να διδάσκονται στην δευτεροβάθμια εκπαίδευση σε όλους τους μαθητές.

## ***Αναφορές***

Chen, P. & Huang, R. (2017). Design Thinking in App Inventor Game Design and Development: A Case Study. *IEEE 17th International Conference on Advanced Learning Technologies (ICALT)*, Timisoara, Romania, pp. 139-141.

Dorling, M., & White, D. (2015). Scratch: A Way to Logo and Python. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE '15)*. ACM, New York, NY, USA, 191-196.

Goldwasser, M., & Letscher, D. (2008). Teaching an object-oriented CS1 -: with Python. *SIGCSE Bulletin*. 40(3), 42-46.

Gray, J, Abelson, H., Wolber, D., & Friend, M. (2012). Teaching CS principles with App Inventor. In *Proceedings of the 50th Annual Southeast Regional Conference (ACM-SE '12)*. ACM, New York, 405-406.

Guo, P., (2014). *Python is now the Most Popular Introductory Teaching Language at Top U.S. Universities*. Survey published at the Communications of the ACM blog (CACM blog).

Jayal, A., Lauria, S., Tucker, A., & Swift, S. (2011). Python for teaching introductory programming: A quantitative evaluation. *Innovation in Teaching and Learning in Information and Computer Sciences*, 10(1), pp. 86–90.

Kaplan, R. (2010). Choosing a first programming language. In *Proceedings of the 2010 ACM conference on Information technology education (SIGITE '10)*. ACM, New York, NY, USA, 163-164

Koulouri, T., Lauria, S., & Macredie, R. (2014). Teaching Introductory Programming: A Quantitative Evaluation of Different Approaches. *ACM Transactions on Computing Education*. 14(4).

Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2010). Learning computer science concepts with scratch. In *Proceedings of the Sixth international workshop on Computing education research (ICER '10)*. ACM, New York, NY, USA, 69-76.

Mishra, S., Balan, S, Iyer, S, & Murthy, S. (2014). Effect of a 2-week scratch intervention in CS1 on learners with varying prior knowledge. In *Proceedings of the 2014 conference on Innovation & technology in computer science education (ITiCSE '14)*. ACM, New York, NY, USA, 45-50.

Lewis, C. (2010). How programming environment shapes perception, learning and goals: logo vs. scratch. In *Proceedings of the 41st ACM technical symposium on Computer science education (SIGCSE '10)*. ACM, New York, NY, USA, 346-350.

Uludag, S., Karakus, M., & Turner, S. (2011). Implementing IT0/CS0 with scratch, app inventor for android, and lego mindstorms. In *Proceedings of the 2011 conference on Information technology education (SIGITE '11)*. ACM, New York, NY, USA, 183-190

Βραχνός, Ε. & Ντούσκα, Σ. (2016). *Όμιλος Αλγοριθμικής στο Ζάννειο Πειραματικό Γυμνάσιο Πειραιά*. 8<sup>th</sup> Conference in Informatics in Education, σελ. 146-155.

### Abstract

The introduction of secondary education students into computer programming is usually done with languages that are based on the turtle geometry, such as Scratch and Logo. With the rapid growth of web-based and mobile applications, there has been a need for a transition to programming languages with a very steep learning curve. These languages allow students to quickly deploy applications that can be run on the web or on their mobile phones. However, there are a few research papers regarding pupils' perceptions about all these different approaches because most students use only one or two programming languages in their school life. The students of our Algorithmic club at Zanneio experimental high-school have developed applications in App Inventor, Python and C ++. They have also used Scratch and Logo as part of a computer science course. The survey involved 16 students who were asked to answer a series of questions in order to investigate their perceptions regarding programming languages.

**Keywords:** programming languages, perceptions, high-school students.