

Η χρήση της δομής επιλογής στο AppInventor

Πίνακας περιεχομένων

1. Τίτλος διδακτικού σεναρίου	2
2. Εκτιμώμενη διάρκεια διδακτικού σεναρίου.....	2
3. Ένταξη του διδακτικού σεναρίου στο πρόγραμμα σπουδών	2
4. Σκοποί και στόχοι του διδακτικού σεναρίου	3
5. Περιγραφή του διδακτικού σεναρίου.....	4
6. Επιστημολογική προσέγγιση και εννοιολογική ανάλυση - Θέματα θεωρίας του διδακτικού σεναρίου	8
7. Χρήση Η.Υ. και γενικά ψηφιακών μέσων για το διδακτικό σενάριο («προστιθέμενη αξία» και αντίλογος, επιφυλάξεις, προβλήματα)	9
8. Αναπαραστάσεις των μαθητών/πρόβλεψη δυσκολιών στο διδακτικό σενάριο.....	10
9. Διδακτικό συμβόλαιο –Διδακτική μετατόπιση –Θεωρητικά θέματα –Διδακτικός θόρυβος.....	10
10. Χρήση εξωτερικών πηγών	11
11. Υποκείμενη θεωρία μάθησης.....	12
12. Επισήμανση μικρομεταβολών.....	13
13. Οργάνωση της τάξης – Εφικτότητα σχεδίασης.....	14
14. Επεκτάσεις / διασυνδέσεις των εννοιών ή των δραστηριοτήτων	14
15. Περιγραφή και ανάλυση φύλλων εργασίας.....	14
16. Αξιολόγηση	14
17. Το επιμορφωτικό σενάριο.....	14
18. Φύλλο εργασίας 1 –.....	15
Δραστηριότητα 1.....	15
Δραστηριότητα 2.....	18
Δραστηριότητα 3.....	20
19. Φύλλο εργασίας 2 –.....	21
Δραστηριότητα 1.....	21
Δραστηριότητα 2.....	23
Δραστηριότητα 3.....	25
20. Φύλλο εργασίας 3 –.....	26
Δραστηριότητα 1.....	26
Δραστηριότητα 2.....	28
21. Φύλλο εργασίας 4 –.....	29
Δραστηριότητα 1.....	29
22. Φύλλο εργασίας 1 - Επανάληψη - Εμπέδωση	32
23. Προτάσεις για περαιτέρω δραστηριότητες - προτεινόμενες εργασίες	33

Η χρήση της δομής επιλογής στο AppInventor

1. Τίτλος διδακτικού σεναρίου

Η χρήση της δομής επιλογής στο AppInventor

2. Εκτιμώμενη διάρκεια διδακτικού σεναρίου

Η προβλεπόμενη διάρκεια του σεναρίου είναι 4 διδακτικές ώρες

3. Ένταξη του διδακτικού σεναρίου στο πρόγραμμα σπουδών

Το διδακτικό σενάριο, εντάσσεται στο πρόγραμμα σπουδών της δευτεροβάθμιας βαθμίδας εκπαίδευσης (Γυμνάσια και τη μεταϋποχρεωτική Δευτεροβάθμια Εκπαίδευση που προσφέρεται από το Γενικό και το Τεχνολογικό Λύκειο).

Πιο αναλυτικά στα Νέα Προγράμματα Σπουδών Υποχρεωτικής εκπαίδευσης (Γυμνάσιο) ο πληροφορικός γραμματισμός (ICT Literacy) θεωρείται γνωστικό-μαθησιακό αντικείμενο αντίστοιχης σπουδαιότητας με τον γλωσσικό γραμματισμό (literacy), τα μαθηματικά και τον επιστημονικό γραμματισμό (scientific literacy). Ως εκ τούτου, η διδασκαλία των αντικειμένων της Πληροφορικής στο Γυμνάσιο και στο Δημοτικό δεν έχει ως στόχο την κατάρτιση των μαθητών σε εφήμερες τεχνολογικές δεξιότητες. Περιλαμβάνει δηλαδή πολλά περισσότερα από την απλή εξοικείωση με τους υπολογιστές και τις λειτουργίες συγκεκριμένων λογισμικών, και στοχεύει στην ανάπτυξη ικανοτήτων αυτόνομης αξιοποίησης των υπολογιστικών και δικτυακών εργαλείων για την επίλυση προβλημάτων, την ενημέρωση, την ψυχαγωγία και, γενικά, τη συμμετοχή τους στη σύγχρονη κοινωνία. Το νέο Π.Σ. παρέχει την ελευθερία στον καθηγητή να σχεδιάσει τη διδασκαλία του, ανάλογα με τις γνώσεις και τις ιδιαιτερότητες των μαθητών του και την υλικοτεχνική υποδομή που έχει στη διάθεσή του. Καλείται δηλαδή να μετατραπεί σε παραγωγό εκπαιδευτικού περιεχομένου που θα ικανοποιεί τις εκάστοτε ιδιαίτερες συνθήκες εφαρμογής του περιεχομένου αυτού.

Στο Γυμνάσιο και στις 3 τάξεις υπάρχει η ενότητα 'Προγραμματίζω τον υπολογιστή μου'. Βασικός στόχος της ενότητας αυτής είναι η σταδιακή εξοικείωση των μαθητών με τον προγραμματισμό μέσα από την αξιοποίηση διαθέσιμων εκπαιδευτικών περιβαλλόντων οπτικού προγραμματισμού. Ειδικότερα αναφέρεται ότι με την ολοκλήρωση της ενότητας ο μαθητής/τρια εκτός των άλλων θα πρέπει να είναι ικανός/ή:

- να αντιλαμβάνεται την αναγκαιότητα και τη χρησιμότητα των δομών επανάληψης και επιλογής να χρησιμοποιεί εντολές επανάληψης και επιλογής στα προγράμματα που αναπτύσσει.

Για την υλοποίηση των δραστηριοτήτων γίνεται αναφορά σε ποικίλα προγραμματιστικά

περιβάλλοντα όπως Scratch, BYOB, Microworld pro, Starlogo TNG, Turtle Art, Kodu, Storytelling Alice, gameMaker, GreenFoot, e-toy.

Αναφορικά με τα προγράμματα σπουδών του Γενικού και του Τεχνολογικού Λυκείου για τη διδασκαλία της πληροφορικής ένας από τους στόχους που τίθεται είναι να επεκτείνουν οι μαθητές τη γενική πληροφορική παιδεία τους με έμφαση στην ανάπτυξη ικανοτήτων και δεξιοτήτων στη χρήση και αξιοποίηση των υπολογιστικών και δικτυακών τεχνολογιών ως εργαλείων μάθησης και σκέψης. Άλλωστε στα πλαίσια αυτά προβλέπονται συνθετικές εργασίες με λογισμικό εφαρμογών γενικής χρήσης, εκπαιδευτικό λογισμικό και προγραμματιστικά περιβάλλοντα.

Το λογισμικό AppInventor παρότι δεν αναφέρεται ούτε στα πλέον πρόσφατα προγράμματα σπουδών (Δημοτικού & Γυμνασίου) κρίνουμε ότι μπορεί να χρησιμοποιηθεί καθώς πρόκειται για ένα ιδιαίτερα εύχρηστο αλλά και 'ισχυρό' προγραμματιστικό περιβάλλον που μπορεί να ανταποκριθεί και στις πιο απαιτητικές μαθησιακές δραστηριότητες σε όλες τις εκπαιδευτικές βαθμίδες καθώς κατασκευάστηκε για να διευκολύνει τη δημιουργία αλληλεπιδραστικών εφαρμογών από παιδιά και νέους αφού ο προγραμματισμός γίνεται με οπτικό τρόπο και η γλώσσα προγραμματισμού περιλαμβάνει διαισθητικό χειρισμό πολυμέσων.

4. Σκοποί και στόχοι του διδακτικού σεναρίου

Σκοπός του σεναρίου είναι οι μαθητές να γνωρίσουν την δομή επιλογής τόσο στην απλή όσο και στην σύνθετη μορφή καθώς και τις εμφωλευμένες επιλογές.

Στόχοι του σεναρίου είναι:

- να μπορούν οι μαθητές να είναι ικανός/ή να αντιλαμβάνεται την αναγκαιότητα και τη χρησιμότητα των δομών επιλογής
- να μπορούν οι μαθητές να χρησιμοποιούν την δομή απλής επιλογής στα προγράμματα που αναπτύσσουν.
- να μπορούν οι μαθητές να χρησιμοποιούν την δομή σύνθετης επιλογής στα προγράμματα που αναπτύσσουν.
- να μπορούν οι μαθητές να χρησιμοποιούν την εμφωλευμένη επιλογή στα προγράμματα που αναπτύσσουν
- να μπορούν να χρησιμοποιούν τα πλακίδια του AppInventor προκειμένου να προγραμματίζουν την λειτουργία της εφαρμογής με χρήση των δύο μορφών της δομής επιλογής.

Προηγούμενες γνώσεις απαραίτητες για την χρήση/ολοκλήρωση της εισαγωγής στις βασικές προγραμματιστικές έννοιες των δομών επιλογής:

- Οι μαθητές γνωρίζουν το προγραμματιστικό περιβάλλον του AppInventor. Είναι εξοικειωμένοι με τις έννοιες των αντικειμένων, των ιδιοτήτων τους και τον χειρισμό

συμβάντων.

- Οι μαθητές, γνωρίζουν ήδη την δομή ακολουθίας καθώς και την έννοια της μεταβλητής.
- Οι μαθητές, γνωρίζουν τον τρόπο ορισμού και χρήσης των μεταβλητών στο AppInventor.
- Οι μαθητές, γνωρίζουν την λειτουργία των αριθμητικών και λογικών τελεστών και τον τρόπο σύνταξης τους στο AppInventor.
- Οι μαθητές είναι σε θέση να ανακτούν, αποθηκεύουν, πακετάρουν και να δημιουργούν απλά προγράμματα με τη χρήση του AppInventor.

5. Περιγραφή του διδακτικού σεναρίου

Η βελτιστοποίηση της διδασκαλίας εννοιών προγραμματισμού είναι ένα από τα διαχρονικά ζητήματα για τη διδακτική της Πληροφορικής (Γόγουλου, Α., Γουλή, Ε., & Γρηγοριάδου, Μ. 2009). Η δομή επιλογής παρουσιάζει αρκετές δυσκολίες στην οικοδόμησή της (Κόμης, 2005). Ωστόσο, φαίνεται πως με τη χρήση διερευνητικών τρόπων διδασκαλίας και προγραμματιστικών περιβαλλόντων που επιτρέπουν την αυτενέργεια των μαθητών οι δυσκολίες αυτές ξεπερνιούνται (Αλεξοπούλου & Κυνηγός, 2008; Γλέζου κ.α., 2005; Ελευθεριώτη, κ.α., 2010). Οι μαθησιακές δυσκολίες των μαθητών βασίζονται κυρίως στην κατανόηση και λειτουργική εφαρμογή της εκτέλεσης της εντολής καθώς και στον προσδιορισμό της τιμής της έκφρασης (αν και τότε η συνθήκη είναι αληθής ή ψευδής) (Τζιμογιάννης και Κόμης, 1999; DuBoulay, 1989). Επιπλέον, οι μαθητές αντιμετωπίζουν δυσκολίες με τη σύνταξη των δομών ελέγχου ή την κατανόηση πολύπλοκων εκφράσεων μέσα στη συνθήκη, όταν για παράδειγμα περιέχουν πολλούς λογικούς τελεστές (Κόμης, 2005).

Η έρευνα για τη διδακτική του προγραμματισμού έχει αποκαλύψει ποικίλα προβλήματα, μεταξύ των οποίων και η εγκυρότητα του μαθήματος στα μάτια των μαθητών (Kolikant, 2004) ενώ συχνά, ούτε οι χρησιμοποιούμενες στη διδασκαλία (εκπαιδευτικές) γλώσσες προγραμματισμού νομιμοποιούνται στα μάτια των μαθητών. Δηλαδή, πολύ συχνά οι μαθητές αμφισβητούν τη χρησιμότητα μιας εκπαιδευτικής γλώσσας όπως η Logo, σε αντιδιαστολή με εμπορικές γλώσσες (π.χ. C, C++, Java). Το μάθημα του προγραμματισμού μέσα από το σχεδιασμό ενός ηλεκτρονικού παιχνιδιού (Papert, 1998) διδάσκεται πιο αποδοτικά, ελκυστικά, διασκεδαστικά γιατί η μάθηση η βασιζόμενη στην τεχνολογία, μπορεί να είναι διασκεδαστική (Rickel et al, 2001). Προκαλεί την άμεση ικανοποίηση του μαθητή η οποία επιταχύνει τον κύκλο μάθησης (O'Kelly et al., 2006). Προσελκύει το ενδιαφέρον των μαθητών οι οποίοι αφιερώνουν πολλές ώρες σε ηλεκτρονικά παιχνίδια (Malone, 1981). Παρουσιάζει ενδιαφέρον γιατί συνδυάζει θέματα τεχνητής νοημοσύνης, προσομοίωσης και άλλων προσεγγίσεων όπως εφαρμογών κινητών τηλεφώνων κ.α. (Shin, et al., 2006). Μπορεί να έχει εκπαιδευτικό χαρακτήρα και αποτέλεσμα (Papert, 1998).

Βοηθά τους μαθητές να προσεγγίζουν την τεχνολογία με ένα πιο ευφάνταστο στυλ που ενθαρρύνει την ανάπτυξη δεξιοτήτων (Martocchio et al., 1992).

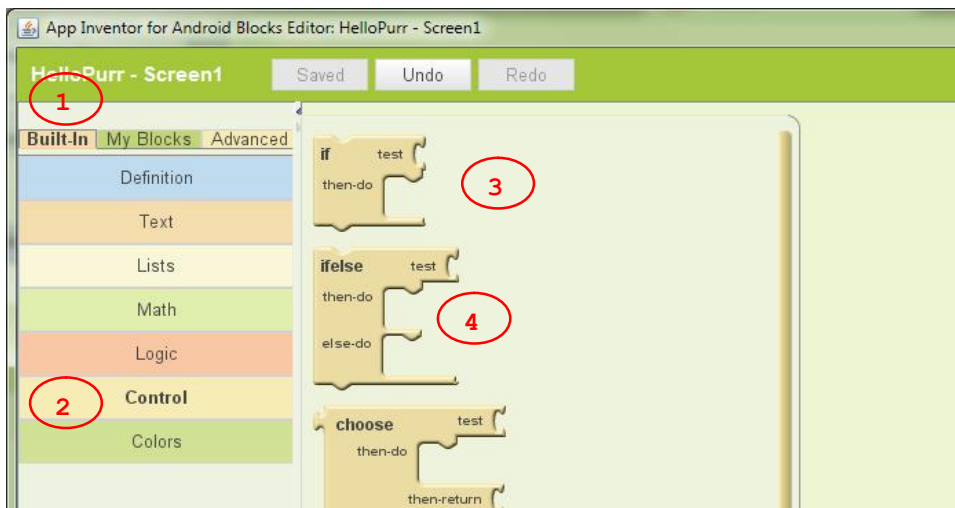
Το εκπαιδευτικό σενάριο που θα περιγράψουμε μπορεί να εφαρμοστεί τόσο στο Γυμνάσιο όσο και στο Γενικό και Τεχνολογικό Λύκειο, εφόσον η δομή επιλογής αποτελεί διδακτικό στα μαθήματα και των 2 σχολικών βαθμίδων. Σκοπός του είναι η διδασκαλία στους μαθητές βασικών έννοιών σχετικών με τις δομές επιλογής «Εάν..τότε» και «Εάν,..τότε ... αλλιώς». Οι μαθητές εργαζόμενοι ατομικά ή σε ομάδες 2-3 μαθητών (ανάλογα την διαθεσιμότητα του εργαστηρίου Πληροφορικής). Ο εκπαιδευτικός τους μοιράζει πολλαπλά φύλλα εργασίας (ανάλογα την δομή επιλογής: απλή, σύνθετη και εμφωλευμένη). Οι μαθητές μέσω των φύλλων εργασίας και των δραστηριοτήτων τους προσπαθούν να γνωρίσουν και να εμβαθύνουν στην έννοια της δομής επιλογής. Ο εκπαιδευτικός δεν περιορίζεται στο μοίρασμα δραστηριοτήτων και στην παθητική παρακολούθηση αλλά συμμετέχει ενεργά στη μαθησιακή διαδικασία, παρακολουθώντας και ελέγχοντας τις απαντήσεις των μαθητών, συμβουλεύοντας τους κριτικά κατά τη φάση ανάπτυξης των αλγορίθμων ενθαρρύνοντας τους να συνεχίσουν την διερευνητική τους προσπάθεια μέσω στοχευόμενων ερωτήσεων και προβληματισμών. Αναγκαίο είναι ο εκπαιδευτικός να έχει ετοιμάσει τα προγράμματα σε διάφορες εκδοχές τους (π.χ. μόνο το σχεδιαστικό κομμάτι καθώς και το ολοκληρωμένο πρόγραμμα) και να έχει φροντίσει ιδίως τα προγράμματα με το σχεδιαστικό τμήμα να τα έχει ήδη τοποθετήσει σε ένα δημόσιο αποθετήριο προσβάσιμο από τους μαθητές είτε τοπικά στους ΗΥ του εργαστηρίου Πληροφορικής. Αυτονόητο είναι οι μαθητές να μην αφιερώνουν χρόνο στη σχεδίαση των εφαρμογών αλλά η προσπάθεια τους να επικεντρωθεί στην υλοποίηση του προγραμματικού τμήματος των δραστηριοτήτων.

Ο εκπαιδευτικός προτού καν πραγματοποιήσει οποιαδήποτε αναφορά στο προγραμματιστικό περιβάλλον, προσπαθώντας να κερδίσει το ενδιαφέρον των μαθητών, μπορεί να αναφέρει παραδείγματα που σχετίζονται με τη καθημερινή ζωή των μαθητών όπως π.χ. ο δείκτης BMI και να τους ρωτήσει αν γνωρίζουν πως προκύπτει ο δείκτης με βάση τις τιμές που του δίνουμε. Επίσης μπορεί να αναφερθεί στον χαρακτηρισμό της βαθμολογίας ενός μαθητή, τότε μένει ένας μαθητής από απουσίες κλπ. Σκοπός είναι να προβληματίσει τους μαθητές στο γεγονός ότι πολύ συχνά για την επίλυση καθημερινών προβλημάτων, ο κάθε άνθρωπος θα πρέπει να λάβει υπόψη του κάποιες συνθήκες και να κληθεί να λάβει αποφάσεις με βάση τις τιμές αυτών των συνθηκών.

Για την διδασκαλία του σεναρίου θα χρησιμοποιήσουμε το AppInventor (AI) (εφευρέτης εφαρμογών). Το AI αποτελεί ένα νέο δωρεάν οπτικό περιβάλλον προγραμματισμού με πλακίδια (blocks), για τη δημιουργία εφαρμογών για κινητά τηλέφωνα με λειτουργικό σύστημα Android. Οι συγκεκριμένες εφαρμογές τρέχουν και σε προσομοιωτή (emulator). Πρόκειται για ένα δικτυακό περιβάλλον ανάπτυξης εφαρμογών το μεγαλύτερο πλεονέκτημα του οποίου είναι ότι

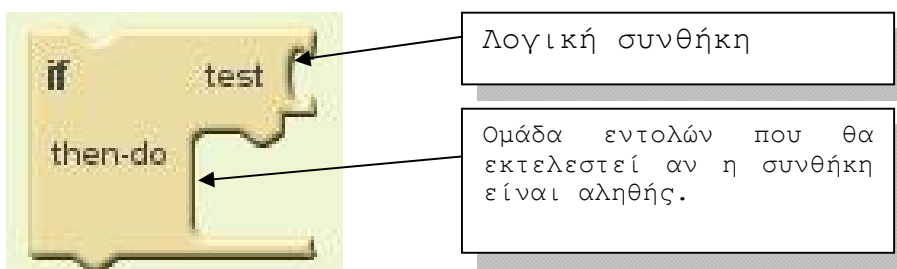
δεν απαιτεί ιδιαίτερες γνώσεις στον προγραμματισμό ή εναλλακτικά η εκμάθησή του λόγω της παιγνιώδους φύσης του είναι αρκετά εύκολη συγκρινόμενη μ' άλλα προγραμματιστικά περιβάλλοντα (Lohr, 2010).

Ας δούμε πως το AppInventor χειρίζεται την δομή επιλογής AN (IF). Καταρχήν να υπενθυμίσουμε ότι ο χειρισμός του προγραμματιστικού τομέα της εφαρμογής μας γίνεται από τον Block editor. Εφόσον λοιπόν έχουμε μεταβεί σε αυτόν, μπορούμε να δούμε τα εργαλεία που μας παρέχει το AppInventor για τον χειρισμό της δομής επιλογής. Πηγαίνοντας στην ενότητα Built-in (βήμα 1) και επιλέγοντας την κατηγορία Control (βήμα 2), μπορούμε να δούμε τις διαθέσιμες επιλογές που έχουμε στην διάθεση μας για τον χειρισμό όλων των δομών (επιλογή, επανάληψη) κ.α. Από τις επιλογές αυτές εμάς μας ενδιαφέρουν οι 2 πρώτες που αναφέρονται στην απλή (3) και στην σύνθετη δομή επιλογής (4), όπως φαίνεται στην εικόνα 1.



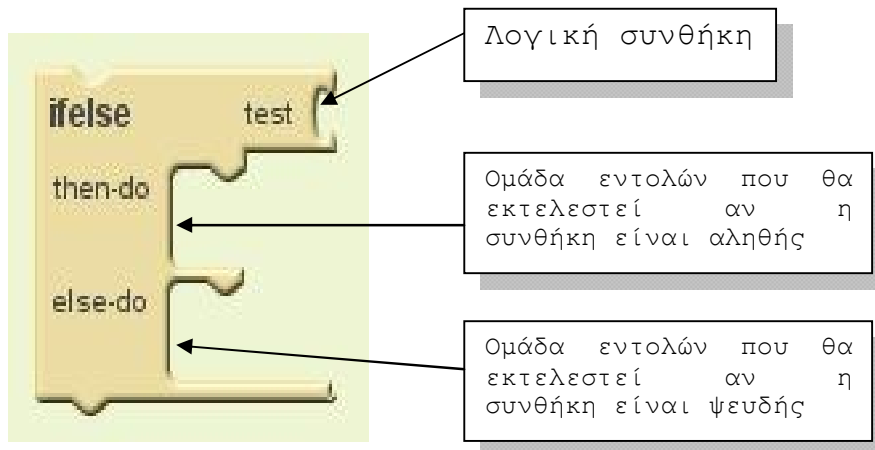
Εικόνα 1. Χειρισμός δομών επιλογής στο AppInventor

Η απλή δομή επιλογής **Αν** (συνθήκη=αληθινή) τότε Ομάδα εντολών **Τέλος_αν**, παρουσιάζεται στην εικόνα 2.



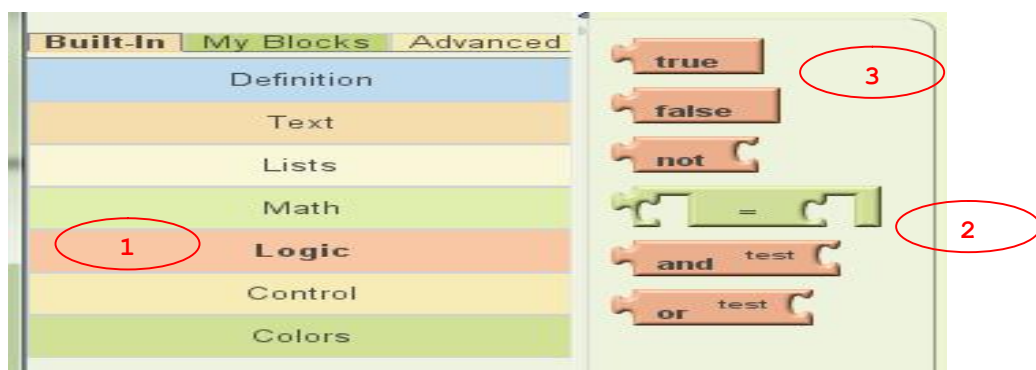
Εικόνα 2. Η δομή απλή επιλογής στο AppInventor

Η σύνθετη δομή επιλογής **Αν** (συνθήκη=αληθινή) τότε Ομάδα εντολών A **αλλιώς** Ομάδα εντολών B **Τέλος_αν**, παρουσιάζεται στην εικόνα 3.



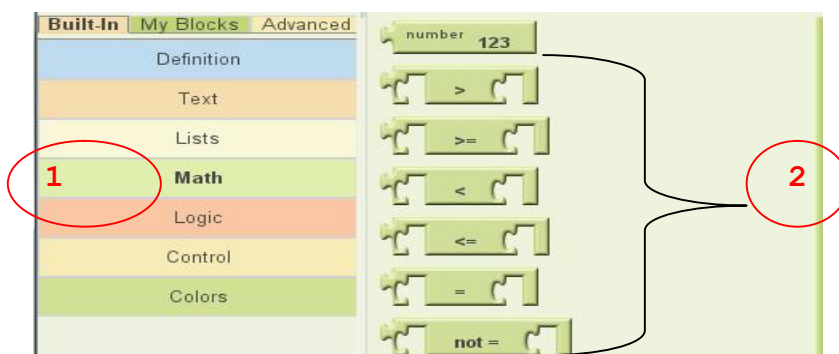
Εικόνα 3. Η δομή σύνθετης επιλογής στο AppInventor

Για τον έλεγχο της συνθήκης μπορούμε να μεταβούμε στην κατηγορία Logic (βήμα 1) και να χρησιμοποιήσουμε είτε τους λογικούς τελεστές **ΚΑΙ**, **Η**, **ΟΧΙ** για τον έλεγχο σύνθετων λογικών συνθηκών (βήμα 2) ενώ αν μια συνθήκη είναι αληθής ή ψευδής την συγκρίνουμε με τα πλακίδια true και false (βήμα 3).



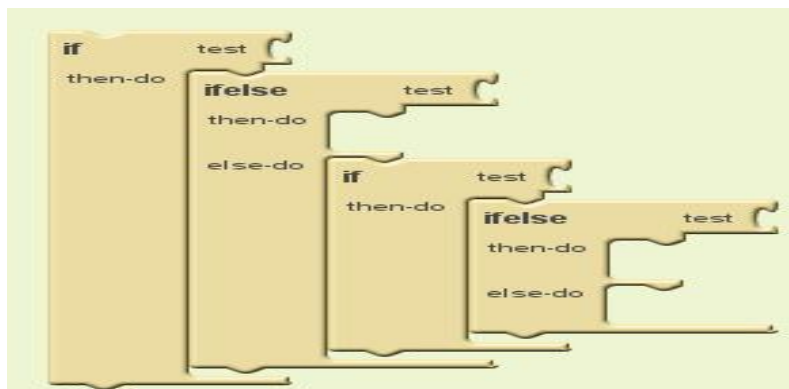
Εικόνα 4. Χειρισμός λογικών τελεστών στο AppInventor

Στο AppInventor υπάρχουν όπως είναι λογικό και τελεστές ανισότητας όπως μπορούμε να διαπιστώσουμε στην εικόνα 5.



Εικόνα 5. Χειρισμός τελεστών ανισότητας στο AppInventor

Τέλος με την ορθή τοποθέτηση των δομών επιλογής, μπορούμε να υλοποιήσουμε εμφωλευμένες δομές επιλογής, όπως φαίνεται στο παράδειγμα της εικόνας 6.



Εικόνα 6. Δημιουργία εμφωλευμένων δομών επιλογής στο AppInventor

6. Επιστημολογική προσέγγιση και εννοιολογική ανάλυση - Θέματα θεωρίας του διδακτικού σεναρίου

Το App Inventor (AI) ανακοινώθηκε για πρώτη φορά ως ένα μικρό έργο των εργαστηρίων της Google (Google Lab) στα τέλη του 2010. Στη συνέχεια, μεταφέρθηκε στο κέντρο για τη εκμάθηση της φορητής μάθησης του MIT (Mobile Learning Center, 2011) για δημόσια χρήση ως λογισμικό ανοικτού κώδικα. Το AI αποτελεί ένα νέο δωρεάν οπτικό περιβάλλον προγραμματισμού με πλακίδια (blocks), για τη δημιουργία εφαρμογών για κινητά τηλέφωνα με λειτουργικό σύστημα Android. Οι συγκεκριμένες εφαρμογές τρέχουν και σε προσομοιωτή (emulator). Αναπτύχθηκε στα εργαστήρια της Google από μια ομάδα με επικεφαλής τον καθηγητή του MIT Harold Abelson (Abelson, 2009) και βασίστηκε σε προηγούμενες μελέτες βασισμένες στη χρήση γραφικών περιβαλλόντων προγραμματισμού όπως το StarLogo TNG (Starlogo TNG, 2012) και τη βιβλιοθήκη Open Blocks (Open Blocks, 2012), ένα σύστημα προγραμματισμού που αναπτύχθηκε για εκπαιδευτικούς λόγους στο MIT.

Αναφορικά με την εκμάθηση προγραμματισμού μια μετανάλυση των Dehnadi et al., (2009) για την διερεύνηση των παραγόντων που επηρεάζουν την επιτυχία στον προγραμματισμό διαπίστωσε ότι 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 στο 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 εννοιών ώστε οι χρήστες να μην είναι υποχρεωμένοι να θυμούνται αρκετές προγραμματιστικές γνώσεις 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 την 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, διαφορετικά εκπαιδευτικά υπόβαθρα και ενδιαφέροντα καθώς επιτρέπουν στους χρήστες να πειραματίζονται με προγραμματιστικές δομές απλά ενώνοντας κομμάτια κώδικα με παρόμοιο τρόπο 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.

Το AppInventor αποτελεί ένα θαυμάσιο προγραμματιστικό περιβάλλον που δίνει την ευκαιρία στους χρήστες να "οικοδομούν πράγματα με νόημα" με σχετική ευκολία και λειτουργεί ως καταλύτης για πλούσιες μαθησιακές ευκαιρίες. Το σενάριο είναι βασισμένο στην εποικοδομητιστική θεωρία του Papert (1980) σύμφωνα με την οποία σημασία έχει να δίνεις στα παιδιά καλά πράγματα να κάνουν έτσι ώστε να μπορούν να μάθουν κάνοντας πολύ καλύτερα απ' ό,τι μπορούσαν πριν. Κατά συνέπεια το διδακτικό σενάριο είναι βασισμένο στη θεωρία μάθησης του εποικοδομητισμού (Constructivism).

7. Χρήση Η.Υ. και γενικά ψηφιακών μέσων για το διδακτικό σενάριο («προστιθέμενη αξία» και αντίλογος, επιφυλάξεις, προβλήματα)

Για την διδασκαλία του σεναρίου θα χρειαστεί ένα εργαστήριο πληροφορικής που να έχει πρόσβαση στο διαδίκτυο για να μπορούν οι μαθητές να προσπελάζουν το διαδικτυακό περιβάλλον δημιουργίας του AppInventor. Επίσης θα πρέπει τοπικά στους σταθμούς εργασίας να έχει εγκατασταθεί μια οποιαδήποτε σχετικά σύγχρονη έκδοση της Java (www.java.com). Επιθυμητό είναι να υπάρχει ένας video projector (ή εφόσον πρόκειται για Γυμνάσιο και ένας διαδραστικός πίνακας) για να μπορεί ο εκπαιδευτικός να παρουσιάζει το περιβάλλον του AppInventor και ότι άλλο αυτός κρίνει απαραίτητο. Επίσης θα μπορούσε ο εκπαιδευτικός προκειμένου να κεντρίσει και να κερδίσει το ενδιαφέρον των μαθητών, να τους πει να κρατούν τα κινητά τους τηλέφωνα ή τις ταμπλέτες τους (αν διαθέτουν) μαζί με τα USB καλώδια που τα συνοδεύουν εντός του εργαστηρίου εφόσον όμως είναι τύπου smartphone και τρέχουν λειτουργικό σύστημα Android. Εναλλακτικά εφόσον το εργαστήριο έχει πρόσβαση σε ασύρματο δίκτυο, οι μαθητές απλά θα κρατούν μόνο τις συσκευές τους καθώς η σύνδεση θα γίνεται ασύρματα.

Όπως ήδη αναφέραμε το περιβάλλον AppInventor, δημιουργία του MIT υπό την αιγίδα του καθηγητή Hal Abelson, είναι το πλέον σύγχρονο και εύκολο περιβάλλον για μια εισαγωγή στον προγραμματισμό. Όπως χαρακτηριστικά αναφέρει ο Hal Abelson σε συνέντευξη του στους New York Times τον Ιούλιο του 2010, ο γενικότερος σκοπός ανάπτυξης του App Inventor είναι να επιτρέψει στους χρήστες συσκευών με εγκατεστημένο λογισμικό Android να μετατραπούν από

καταναλωτές σε δημιουργούς. Το συγκεκριμένο πρόγραμμα της Google και του MIT, ανέφερε ο Abelson (2010), έχει ως βασικό στόχο να παράσχει στους χρήστες, ιδίως τους νέους, ένα απλό εργαλείο το οποίο θα τους επιτρέψει να πειραματιστούν με το λογισμικό της έξυπνης κινητής συσκευής τους, με παρόμοιο τρόπο που οι άνθρωποι τόσες δεκαετίες κάνουν με τους προσωπικούς τους υπολογιστές. Επίσης, θεωρεί ότι με την πάροδο των χρόνων τα απλοποιημένα εργαλεία προγραμματισμού όπως η Basic, η Logo και το Scratch έχουν ανοίξει την πόρτα σε καινοτομίες όλων των ειδών.

Όπως επισημαίνει ο Kirkpatrick (2010), στην ενθουσιώδη παρατήρηση αν το App Inventor θα συμβάλει στην ανάπτυξη εφαρμογών για έξυπνες κινητές συσκευές όσο συνέβαλε το λογισμικό Quark στις εκδόσεις ή το Blogger στην ανάπτυξη των ιστολογίων η απάντηση από ειδικούς των Media είναι σαφέστατα θετική.

8. Αναπαραστάσεις των μαθητών/πρόβλεψη δυσκολιών στο διδακτικό σενάριο

Οι έρευνες που αφορούν στη διδακτική προγραμματιστικών δομών μέσω της χρήσης υπολογιστικών περιβαλλόντων εστιάζουν κυρίως: στο είδος των δομών που διδάσκονται, το υπολογιστικό περιβάλλον και τη γλώσσα προγραμματισμού που χρησιμοποιείται και την διδακτική προσέγγιση που ακολουθείται (Κόμης, 2001; DuBoulay, 1989, Brusilovsky et al., 1997, Γρηγοριάδου, κ.α., 2002). Η δομή επιλογής παρουσιάζει αρκετές δυσκολίες στην οικοδόμησή της (Κόμης, 2005). Ωστόσο, φαίνεται πως με τη χρήση διερευνητικών τρόπων διδασκαλίας και προγραμματιστικών περιβαλλόντων που επιτρέπουν την αυτενέργεια των μαθητών οι δυσκολίες αυτές ξεπερνιούνται (Αλεξοπούλου & Κυνηγός, 2008; Γλέζου κ.α., 2005; Ελευθεριώτη, κ.α., 2010).

Οι μαθησιακές δυσκολίες των μαθητών βασίζονται κυρίως στην κατανόηση και λειτουργική εφαρμογή της εκτέλεσης της εντολής καθώς και στον προσδιορισμό της τιμής της έκφρασης (αν και τότε η συνθήκη είναι αληθής ή ψευδής) (Τζιμογιάννης και Κόμης, 1999; DuBoulay, 1989). Επιπλέον, οι μαθητές αντιμετωπίζουν δυσκολίες με τη σύνταξη των δομών ελέγχου ή την κατανόηση πολύπλοκων εκφράσεων μέσα στη συνθήκη, όταν για παράδειγμα περιέχουν πολλούς λογικούς τελεστές (Κόμης, 2005).

9. Διδακτικό συμβόλαιο –Διδακτική μετατόπιση –Θεωρητικά θέματα –Διδακτικός θόρυβος

Εφόσον πριν την εκτέλεση του σεναρίου υπάρχουν οι ελάχιστες τεχνικές προδιαγραφές (σύνδεση στο διαδίκτυο και εγκατεστημένη java) και ο εκπαιδευτικός έχει φροντίσει να έχει τοποθετήσει τοπικά στους ΗΥ των μαθητών ή σε ένα δημόσιο αποθετήριο τις διάφορες εκδόσεις των αλγορίθμων - προγραμμάτων αναμένεται ότι δεν θα υπάρξουν ιδιαίτερα

προβλήματα κατά την εκτέλεση του σεναρίου. Επομένως δεν θεωρούμε ότι δεν θα εμφανιστεί διδακτικός θόρυβος. Επίσης λόγω του γεγονότος ότι τα φύλλα εργασίας έχουν δημιουργηθεί με τέτοιο τρόπο ώστε να είναι απλά, ρεαλιστικά και να οδηγούν το μαθητή βήμα - βήμα στην ομαλή εξοικείωση του με την εφαρμογή θεωρούμε ότι το διδακτικό συμβόλαιο δεν θα ανατραπεί.

10. Χρήση εξωτερικών πηγών

Πηγές εκμάθησης

- [Online εκτέλεση του AppInventor](#)
- [Οδηγοί \(tutorials\) για το AppInventor](#)
- [Δωρεάν βιβλίο για το AppInventor](#)

Ιστολόγια και αναφορές στο Διαδίκτυο

Abelson, H. (2009). App Inventor for Android. Retrieved 10 January 2013 from <http://googleresearch.blogspot.gr/2009/07/app-inventor-for-android.html>

App Inventor Learning Portal (2012). Retrieved 10 January 2013 from <http://appinventor.mit.edu/explore/content/what-app-inventor.html>

Bloomberg Businessweek (2012). Can Android Market Catch Up to the App Store? Retrieved 10 January 2013 from <http://www.businessweek.com/stories/2009-10-26/can-android-market-catch-up-to-the-app-store>

King, R. (2012). Turning Girls into Tech Entrepreneurs with a Single App. Bloomberg Business Week 2012. Retrieved 28 February 2013 from <http://www.businessweek.com/technology/turning-girls-into-tech-entrepreneurs-with-a-single-app-02062012.html>

Kirkpatrick, M. (2010). 5 Big Questions About Google's New App Inventor. Retrieved 10 January 2013 from http://www.readwriteweb.com/archives/5_big_questions_about_googles_app_inventor.php

Lohr, S. (2010). Google's Do-It-Yourself App Creation Software. Retrieved 10 January 2013 from http://www.nytimes.com/2010/07/12/technology/12google.html?_r=2&partner=rss&emc=rss

MIT Center for Mobile Learning, (2011). Retrieved 10 December 2012 from <http://mitmobilelearning.org/welcome>

StarLogo TNG (2012). MIT Scheller Teacher Education Program. Retrieved 28 February 2013 from <http://education.mit.edu/projects/starlogo-tng>.

Open Blocks (2012). MIT Scheller Teacher Education Program. Retrieved 28 February 2013 from

Page, L. (2012). Update from the CEO. Retrieved 10 December 2012 from <http://investor.google.com/corporate/2012/ceo-letter.html>

Αρθρογραφία για το AppInventor και τον προγραμματισμό

Brennan, K. (2009). Scratch-Ed: an online community for scratch educators. In A. Dimitracopoulou, C. O'Malley, D. Suthers & P. Reimann (Eds.). *Proceedings of the 9th International Conference on*

Computer supported collaborative learning (CSCL'09), (Vol. 2) International Society of the Learning Sciences, (pp. 76-78).

Dehnadi, S., Bornat, R., & Adams, R. (2009). Meta-analysis of the effect of consistency on success in early learning of programming. 21st Annual Workshop of the Psychology of Programming Interest Group (p. 10pp).

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.

Morreli, R., de Lanerolle, T., Lake, P., Limardo, N., Tamotsu, E., & Uche, C. (2011). Can Android App Inventor bring Computational Thinking to K-12? *Proceedings of the 34th SIGCSE Technical Symposium on Computer science education*, March 9-12, 2011, USA: Dallas-Texas.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: programming for all. *Commun. ACM* 52, 11 November 2009, 60-67.

Μάργαρης, Α., & Παπαστεργίου, Μ. (2008). Εισάγοντας αρχάριους στον προγραμματισμό με τα περιβάλλοντα Kara: Μια προσέγγιση βασισμένη στη θεωρία υπολογισμού. Στο Β. Κόμης (επιμ) *Πρακτικά 4^{ου} Πανελληνίου Συνεδρίου με Διεθνή Συμμετοχή «Διδακτική της Πληροφορικής»*, (σσ. 81-90), Πάτρα, Μάρτιος 2008 Αθήνα: Εκδόσεις Νέων Τεχνολογιών.

Ουσαίτσα, Ε., & Σάμψων, Δ. (2012). Αξιοποίηση Εργαλείων για τη δημιουργία Ηλεκτρονικών Μαθημάτων κατάλληλων για Κινητές Συσκευές. Στο Χ. Καραγιαννίδης, Π. Πολίτης & Η. Καρασαββίδης (επιμ.) *Πρακτικά του 8ου Πανελλήνιο Συνέδριο με Διεθνή Συμμετοχή «Τεχνολογίες Πληροφορίας & Επικοινωνίας στην Εκπαίδευση»*, Πανεπιστήμιο Θεσσαλίας, Βόλος, 28-30 Σεπτεμβρίου 2012.

Ελληνική ακαδημαϊκή βιβλιογραφία

Στην Ελλάδα δεν υπάρχει σχεδόν καμία ακαδημαϊκή αναφορά για το AppInventor και την εκπαιδευτική αξιοποίησή του, εκτός της εργασίας των Παπαδάκη Σταμάτιου, Καλογιαννάκη Μιχαήλ και Ζαράνη Νικόλαου η οποία παρουσιάστηκε στο 7ο Πανελλήνιο Συνέδριο των Καθηγητών Πληροφορικής στην Θεσσαλονίκη τον Απρίλιο του 2013. Τίτλος εργασίας 'Δημιουργώντας εφαρμογές για έξυπνες φορητές συσκευές με το AppInventor'

11. Υποκείμενη θεωρία μάθησης

Το περιβάλλον App Inventor όπως το SCRATCH, το BYOB κ.α. έχει χαρακτήρα παιχνιδιάρη. Επίσης αναφέρεται σε ένα τομέα της καθημερινής ζωής (κινητή τηλεφωνία, έξυπνες φορητές συσκευές) ο οποίος σύμφωνα με έρευνες έχει κυριαρχήσει στη ζωή των μαθητών ανεξαρτήτως ηλικίας. Όπως χαρακτηριστικά αναφέρουν οι Johnson et al., (2010, 2011) οι φορητές συσκευές με τη μορφή των έξυπνων κινητών τηλεφώνων και των ταμπλετών αποκτούν όλο και μεγαλύτερη δημοτικότητα καθώς λόγω της ισχυρής επεξεργαστικής ισχύος τους σε συνδυασμό με τη δυνατότητα συνδεσιμότητά τους με το διαδίκτυο και τη διαθεσιμότητα διαφόρων τύπων,

πολυπληθών και εύκολων στη χρήση εφαρμογών για φορητές συσκευές ("mobile apps") έχουν διεισδύσει στις περισσότερες καθημερινές ανθρώπινες δραστηριότητες. Τα έξυπνα κινητά τηλέφωνα και οι ταμπλέτες θεωρούνται ως μία από τις έξι νέες τεχνολογίες που μπορούν να έχουν σημαντικό αντίκτυπο στη διδασκαλία, τη μάθηση και την έρευνα στην εκπαίδευση (Johnson et al., 2011).

Με βάση όσα προαναφέραμε αναμένεται λοιπόν να υπάρξει ζωνρή συμμετοχή των μαθητών σε όλες τις δραστηριότητες. Επιπλέον, είναι ενδεχόμενο, πολύ γρήγορα οι μαθητές να θελήσουν να δοκιμάσουν τροποποιήσεις του περιβάλλοντος και των λοιπών στοιχείων των (μικρών αρχικά) προγραμμάτων, επιθυμώντας να τροποποιήσουν τα χαρακτηριστικά του σύμφωνα με τις επιθυμίες τους. Στο παρόν διδακτικό σενάριο λοιπόν, θα πραγματοποιηθεί πειραματισμός με τα στοιχεία του προγραμματιστικού περιβάλλοντος με ενεργητική συμμετοχή. Άλλωστε όπως αναφέρει η Γλέζου (2002) *"οι ΤΠΕ και ιδιαίτερα τα Logo-like περιβάλλοντα μπορούν να αξιοποιηθούν για τη σχεδίαση και την ανάπτυξη υπολογιστικών εργαλείων, τα οποία προσφέρουν στους μαθητές τη δυνατότητα έκφρασης κι αξιοποίησης των σκέψεων, ιδεών και διαισθησέων τους και υποστηρίζουν τη διαδικασία οικοδόμησης της γνώσης διαμορφώνοντας πλούσια σε ευκαιρίες προβληματισμού και πειραματισμού περιβάλλοντα μάθησης"*.

Το στοιχείο αυτό, σε συνδυασμό με την οργάνωση της τάξης σε μικρές ομάδες, προσφέρει ένα πολύ καλό περιβάλλον για την ανάπτυξη μιας ισχυρής αλληλεπίδρασης που μπορεί να ευνοήσει τη μάθηση. Ευνοείται λοιπόν ιδιαίτερα η δημιουργία ενός τυπικού περιβάλλοντος κοινωνιο-κονστрукτιβιστικού καθώς θεωρούμε ότι μέσω των κοινωνικών αλληλεπιδράσεων επηρεάζεται η διαδικασία με την οποία οικοδομείται η γνώση. Το σενάριο είναι θεμελιωμένο στην θεωρία μάθησης του εποικοδομητισμού διότι ο μαθητής χτίζει την γνώση του ανιχνεύοντας, διερευνώντας και αλληλεπιδρώντας οπτικά με τις βασικές εντολές της γλώσσας προγραμματισμού του AppInventor δημιουργώντας απλά προγράμματα.

Όπως χαρακτηριστικά αναφέρουν οι Kafai & Resnick (1996), *"Ο Κονστрукτιβισμός προτείνει ότι οι μαθητευόμενοι κατασκευάζουν νέες ιδέες όταν ενεργά ασχολούνται με τη δημιουργία εξωτερικής κατασκευής -μπορεί ένα ρομπότ, ένα ποίημα, ένα κάστρο στην άμμο, ένα πρόγραμμα στον υπολογιστή- πάνω στις οποίες αναστοχάζονται και μοιράζονται με άλλους. Έτσι ο κονστрукτιβισμός εμπλέκει δυο διαπλεκόμενους τύπους κατασκευής: την οικοδόμηση της γνώσης στο πλαίσιο οικοδόμησης κατασκευών με προσωπικό νόημα"*.

12. Επισήμανση μικρομεταβολών

Μοναδικό πρόβλημα που ενδεχόμενα μπορεί να παρουσιαστεί είναι εφόσον τα προγράμματα που θα κατασκευαστούν είναι αρκετά μεγάλα σε μέγεθος η εκτέλεση τους στον προσομοιωτή είναι αρκετά αργή (εφόσον χρησιμοποιηθεί προσομοιωτής και όχι οι κινητές συσκευές των μαθητών). Ωστόσο επειδή οι δραστηριότητες που θα εκτελέσουν οι μαθητές θα είναι μικρές σε

μέγεθος και κατάλληλα στοχευμένες δε αναμένεται να υπάρξουν ιδιαίτερα προβλήματα. Εξάλλου το παιγνιώδες ύφος της εφαρμογής και η δυνατότητα προγραμματισμού με πλακίδια (blocks), μειώνει το νοητικό φορτίο που απαιτείται για την συγγραφή κώδικα και ταυτόχρονα βοηθάει ιδίως τους αρχάριους στον προγραμματισμό χρήστες να επικεντρωθούν στην επίλυση ενός προβλήματος δίχως να είναι αναγκασμένοι να δώσουν βαρύτητα στο αυστηρό τυπικό και συντακτικό μιας κανονικής γλώσσας προγραμματισμού.

13. Οργάνωση της τάξης – Εφικτότητα σχεδίασης

Η διδασκαλία θα λάβει χώρα εντός του σχολικού εργαστήριου πληροφορικής. Οι μαθητές θα εργαστούν ατομικά είτε σε ομάδες των δύο ή τριών ατόμων ανά ηλεκτρονικό υπολογιστή. Ενδεχόμενα ο εκπαιδευτικός μπορεί να επιτρέψει σε όσους μαθητές το επιθυμούν και διαθέτουν, να φέρουν και να συνδέσουν με τον υπολογιστή τους το κινητό τους τηλέφωνο με λειτουργικό Android, για μεγαλύτερη ταχύτητα και αληθοφάνεια στην εκτέλεση των προγραμμάτων σε σχέση με τη χρήση του προσομοιωτή.

14. Επεκτάσεις / διασυνδέσεις των εννοιών ή των δραστηριοτήτων

Μετά την ολοκλήρωση του παρόντος σεναρίου οι μαθητές θα είναι προετοιμασμένοι για να εργαστούν με επόμενα σενάρια τα οποία θα περιλαμβάνουν τη χρήση πιο σύνθετων αλγοριθμικών δομών όπως επανάληψη αλλά και χειρισμό στοιχείων αντικειμενοστραφούς προγραμματισμού προκειμένου οι εφαρμογές που θα δημιουργήσουν οι μαθητές να είναι πιο 'απαιτητικές' αλλά και πιο διαδραστικές.

15. Περιγραφή και ανάλυση φύλλων εργασίας

Τα φύλλα εργασίας αφορούν την εκμάθηση την εκμάθηση της δομής επιλογής μέσα από το περιβάλλον του AppInventor..

16. Αξιολόγηση

Ο εκπαιδευτικός μπορεί να χρησιμοποιήσει τα συνηθισμένα και αξιόπιστα open source λογισμικά όπως π.χ. το κλασικό hot potatoes, το ελληνικό και δωρεάν hot pepper ή το ISPRING προκειμένου να δημιουργήσει τεστ αυτοαξιολόγησης (κλειστού τύπου), σταυρόλεξα, κρυπτόλεξα κλπ. αλλά και κουίζ, flashcards κ.α.

17. Το επιμορφωτικό σενάριο

Θα πραγματοποιηθεί γνωριμία των μαθητών με την δομή επιλογής στο περιβάλλον του AppInventor και θα γίνει προσπάθεια δημιουργίας εφαρμογών για έξυπνες φορητές συσκευές

με λειτουργικό σύστημα Android από τους μαθητές. Θα αντιμετωπιστούν ενδεχόμενα προβλήματα που οφείλονται σε λανθασμένες αντιλήψεις και προϋπάρχουσες γνώσεις των μαθητών όπως παραλληλισμός, ανθρωπομορφισμός κ.α. Τα φύλλα εργασίας που θα δοθούν στους μαθητές παρουσιάζονται στις επόμενες ενότητες.

18. Φύλλο εργασίας 1 –

Δραστηριότητα 1

Στόχος της δραστηριότητας είναι η εξοικείωση των μαθητών με την έννοια της δομής επιλογής σε συνδυασμό με την χρήση λογικών συνθηκών.

Άσκηση

Γνωρίζετε ήδη ότι αρκετές φορές στην καθημερινή σας ζωή προκειμένου για να λύσετε ένα πρόβλημα πρέπει να ελέγξετε αν κάτι (μια συνθήκη) ισχύει ή όχι. Επίσης έχετε ήδη μάθει ότι αυτό που ελέγχουμε ονομάζεται συνθήκη και είναι δυνατόν να έχει μια από δύο δυνατές τιμές: αλήθεια ή ψέμα. Αλήθεια είναι αν αυτό που ελέγχουμε ισχύει, ψέμα αν δεν ισχύει.

Με βάση την τιμή της συνθήκης που ελέγχουμε εμείς καλούμαστε να πάρουμε κάποιες αποφάσεις:

- Αν βρέχει θα πάρω ομπρέλα
- Αν είναι Κυριακή θα πάω εκκλησία
- Αν το φανάρι είναι κόκκινο, σταματάω και περιμένω.

Στα παραπάνω παραδείγματα διαπιστώνουμε ότι οι ενέργειες που περιγράψαμε εκτελούνται μόνο εφόσον η συνθήκη είναι αληθής, δηλαδή ισχύει. Όταν η συνθήκη είναι ψευδής απλά δεν γίνεται κάτι.

Συμπληρώστε και εσείς 3 παραδείγματα από την καθημερινή σας ζωή, στα οποία καλείστε να πάρετε μια απόφαση.

1.
2.
3.

Πολλές φορές, και στον προγραμματισμό υπολογιστών, ανάλογα με τα δεδομένα του προβλήματος καλούμαστε να πάρουμε αποφάσεις μέσα στον αλγόριθμο και να εκτελέσουμε κάποια συγκεκριμένα κομμάτια του αλγορίθμου και να αγνοήσουμε κάποια άλλα. Προκύπτει, λοιπόν, η έννοια της δομής επιλογής.

Η δομή επιλογής είναι από τις βασικότερες δομές για την ανάπτυξη ενός αλγορίθμου. Χρησιμοποιείται για τη λήψη απόφασης μεταξύ δύο διαφορετικών καταστάσεων, εκ των οποίων η μία είναι αληθής και η άλλη ψευδής. Στα περισσότερα προβλήματα περιλαμβάνονται

κάποιοι έλεγχοι δεδομένων. Ανάλογα με τα αποτελέσματά τους επιλέγονται οι ενέργειες (επεξεργασίες) που θα ακολουθήσουν.

Η δομή επιλογής συναντάται σε διάφορες μορφές, η απλούστερη εκ των οποίων είναι η δομή απλή επιλογής, στην οποία απλά ελέγχουμε την τιμή μιας συνθήκης και αν αυτή είναι αληθής απλά εκτελούμε τις εντολές που ορίζει (περικλείει) ειδάλλως δεν κάνουμε κάτι.

Για παράδειγμα αν θέλαμε το πρόγραμμα μας, να διαβάσει την βαθμολογία ενός μαθητή και αν είναι μεγαλύτερη του 18 να εμφανίζει το μήνυμα 'Άριστος' σε γενική μορφή θα γράφαμε μία εντολή της μορφής:

Αν (συνθήκη) τότε εμφάνισε μήνυμα

Σε πιο συγκεκριμένη μορφή θα γράφαμε:

Αν (βαθμολογία > 18) τότε γράψε 'Άριστος'

Η (βαθμολογία > 18) είναι η λογική συνθήκη και η εντολή γράψε 'Άριστος' θα εκτελεστεί **μόνο όταν** η συνθήκη είναι αληθής.

Δηλαδή όπως και στα παραδείγματα από την καθημερινή ζωή, παρατηρούμε ότι και στον υπολογιστή εκτελείται η εντολή ή η ομάδα εντολών μόνο όταν η συνθήκη είναι αληθής. Σε περίπτωση που η συνθήκη είναι ψευδής απλά δεν εκτελείται καμία εντολή. Είναι πολύ σημαντικό να θυμόμαστε ότι ανάλογα με την τιμή της συνθήκης είναι πιθανόν η εντολή να μην εκτελεστεί ποτέ.

Η μορφή αυτή της δομής επιλογής που μόλις περιγράψαμε είναι η πιο απλή και ονομάζεται δομή απλής επιλογής.

Μπορείτε να σκεφτείτε 3 παραδείγματα και να γράψετε τις εντολές όπως θα τις γράφατε στον υπολογιστή; Ας θυμηθούμε ότι οι τελεστές ανισότητας που χρησιμοποιούμε είναι οι >, <, >=, <=, =, <> ενώ οι λογικοί τελεστές είναι οι: ΚΑΙ, Η, ΟΧΙ.

Σημειώστε στα παραδείγματα σας, ποια είναι η συνθήκη που εφόσον είναι αληθής θα εκτελεστεί η εντολή ή η ομάδα εντολών και ποια/ποιες είναι αυτές.

Παράδειγμα 1:

Συνθήκη:.....

Εντολή /Ομάδα εντολών:

Παράδειγμα 2:

Συνθήκη:.....

Εντολή /Ομάδα εντολών:

Παράδειγμα 3:

Συνθήκη:.....

Εντολή /Ομάδα εντολών:

Δραστηριότητα 2

Οδηγίες υλοποίησης

Ας προσπαθήσουμε να υλοποιήσουμε ένα από τα παραπάνω παραδείγματα στο AppInventor. Μιας και είσατε μαθητές θα υλοποιήσουμε το παράδειγμα με την βαθμολογία: θα δημιουργήσετε μια εφαρμογή η οποία θα ζητάει από τον χρήστη την βαθμολογία του και αν αυτή είναι μεγαλύτερη του 18 θα εμφανίζει το μήνυμα 'Άριστος'.

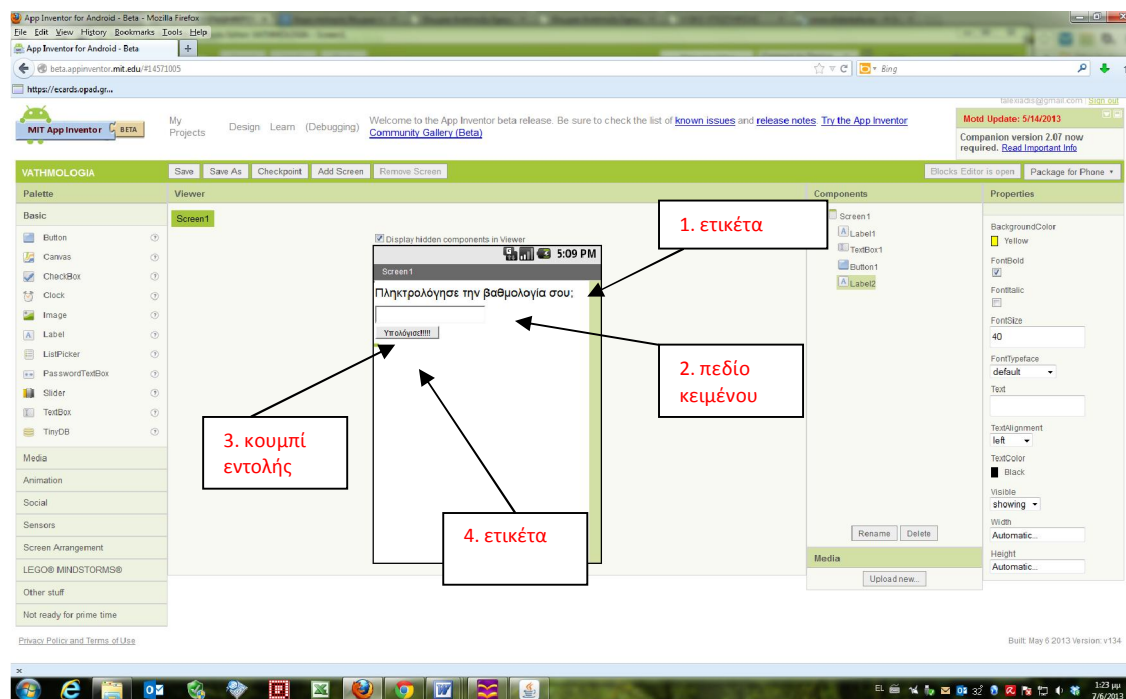
Η γενική δομή της απλής επιλογής που θα χρησιμοποιήσουμε είναι:

Αν (συνθήκη) τότε εμφάνισε μήνυμα

Αν (βαθμολογία > 18) τότε γράψε 'Άριστος'

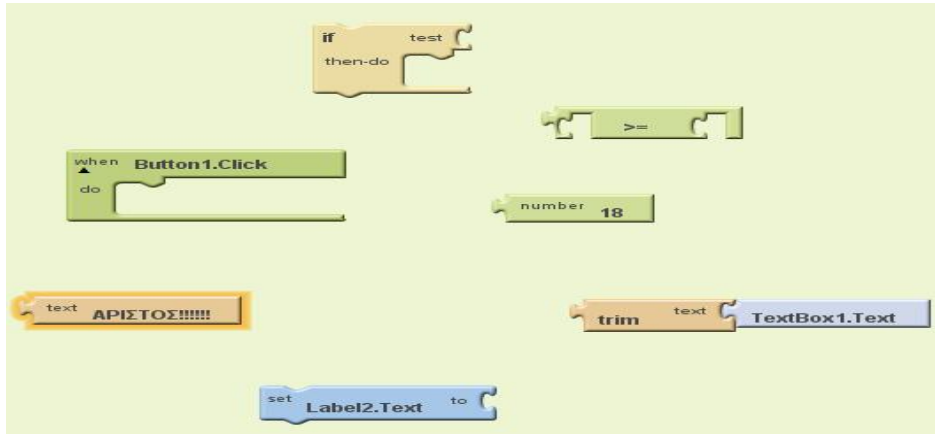
Η (βαθμολογία > 18) είναι η λογική συνθήκη και η εντολή γράψε 'Άριστος' θα εκτελεστεί μόνο όταν η συνθήκη είναι αληθής.

Το οπτικό κομμάτι της εφαρμογής σας θα είναι το ακόλουθο:

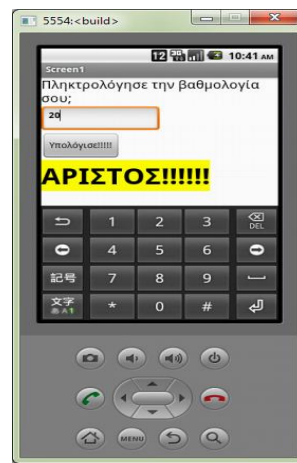


Αν σε περίπτωση που δυσκολεύεσαι να υλοποιήσεις το οπτικό κομμάτι της εφαρμογής, ζήτη από τον καθηγητή σου, να σου δώσει το project (έργο) προκειμένου να ασχοληθείς με την υλοποίηση της δομής επιλογής στον Block Editor.

Τα πλακίδια που θα χρειαστείς να συνδέσεις είναι τα :



Στις επόμενες 2 εικόνες, φαίνεται η εφαρμογή όταν θα τρέξει μέσα από τον προσομοιωτή και το αποτέλεσμα που θα προκύψει όταν πληκτρολογήσουμε το νούμερο 20.



Δοκιμάστε να πληκτρολογήσετε διάφορα νούμερα, μικρότερα του 18. Τι παρατηρείτε;

.....

.....

Για πιο λόγο νομίζετε ότι παρατηρήσατε κάτι παράξενο:

.....

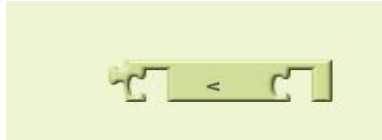
.....

Δραστηριότητα 3

Οδηγίες υλοποίησης

Χρησιμοποιώντας ως βάση την προηγούμενη εφαρμογή, δοκιμάστε να τροποποιήσετε τον κώδικα της ώστε να εμφανίζει το μήνυμα 'Μετεξεταστέος' για όποιον μαθητή έχει βαθμολογία μικρότερη του 9.

Ο τελεστής ανισότητας που θα χρησιμοποιήσετε είναι ο :



Αντιμετωπίσατε κάποιο πρόβλημα στην εκτέλεση της εφαρμογής;

.....

Προσπαθήστε να πληκτρολογήσετε νούμερα μεγαλύτερα του 9. Τι μήνυμα εμφανίζει η εφαρμογή σας και για πιο λόγο πιστεύετε ότι γίνεται αυτό;

.....

.....

19. Φύλλο εργασίας 2 –

Δραστηριότητα 1

Πολύ συχνά στη καθημερινή μας ζωή χρειάζεται να παίρνουμε αποφάσεις σύμφωνα με τις οποίες υποχρεωτικά πρέπει να κάνουμε κάτι ανάλογα με αν αυτό που εξετάζουμε είναι αληθής ή όχι.

Για παράδειγμα ένας μαθητής ενδεχόμενα κάνει τους εξής συλλογισμούς:

- Αν ο ουρανός είναι βροχερός θα πάρω αδιάβροχο αλλιώς θα πάρω ζακέτα.
- Αν αύριο έχω διαγώνισμα θα διαβάσω μέχρι αργά αλλιώς θα δω τηλεόραση.
- Αν βρέχει θα πάω στο σχολείο με το λεωφορείο αλλιώς θα περπατήσω.
- Αν το φανάρι είναι κόκκινο θα περιμένω αλλιώς θα περάσω απέναντι.

Παρατηρούμε στα παραπάνω παραδείγματα ότι εκτελούμε διαφορετικές ενέργειες ανάλογα με το αν αυτό που εξετάζουμε ισχύει ή όχι.

Συμπληρώστε και εσείς 3 παραδείγματα από την καθημερινή σας ζωή, στα οποία καλείστε να πάρετε 2 διαφορετικές αποφάσεις ανάλογα με την τιμή μιας συνθήκης.

1.
2.
3.

Πολύ συχνά στον προγραμματισμό χρειάζεται να εκτελέσουμε υποχρεωτικά την μια από δυο διαφορετικές ομάδες εντολών. Στην περίπτωση αυτή χρησιμοποιούμε μια ειδική μορφή της δομής επιλογής η οποία ονομάζεται δομή σύνθετης επιλογής. Η δομή σύνθετης επιλογής χρησιμοποιείται στην περίπτωση που επιθυμούμε να εκτελέσουμε εναλλακτικά δυο ομάδες εντολών. Κριτήριο για το ποιο σεν εντολών θα εκτελεστεί αποτελεί κάποια συνθήκη. Στην περίπτωση που η συνθήκη ισχύει θα εκτελεστεί η πρώτη ομάδα εντολών διαφορετικά θα εκτελεστεί η δεύτερη

Για παράδειγμα αν θέλαμε το πρόγραμμα μας, να διαβάζει την βαθμολογία ενός μαθητή και αν είναι μεγαλύτερη του 9 να εμφανίζει το μήνυμα 'Πέρασες την τάξη', ενώ αν είναι μικρότερη ή ίση του 9 να εμφανίζει το μήνυμα 'Μετεξεταστέος', σε γενική μορφή θα γράφαμε μία εντολή της μορφής:

```
Αν (συνθήκη) τότε
    εμφάνισε μήνυμα1
αλλιώς
    εμφάνισε μήνυμα2
τέλος_αν
```

Σε πιο συγκεκριμένη μορφή θα γράφαμε:

Αν (βαθμολογία >9) τότε

γράψε 'Πέρασες την τάξη'

αλλιώς

γράψε 'Μετεξεταστέος'

Τελος_αν

Η (βαθμολογία > 9) είναι η λογική συνθήκη. Η εντολή **γράψε 'Πέρασες την τάξη'** θα εκτελεστεί **μόνο όταν** η συνθήκη είναι αληθής ενώ η εντολή **γράψε 'Μετεξεταστέος'** θα εκτελεστεί **μόνο όταν** η συνθήκη είναι ψευδής. Είναι πολύ σημαντικό να θυμόμαστε ότι μόνο μια από τις 2 εντολές θα εκτελεστεί και ποτέ και οι 2 διαφορετικές εντολές.

Μπορείτε να σκεφτείτε 3 παραδείγματα σύνθετης επιλογής και να γράψετε τις εντολές όπως θα τις γράφατε στον υπολογιστή; Ας θυμηθούμε ότι οι τελεστές ανισότητας που χρησιμοποιούμε είναι οι >, <, >=, <=, =, <> ενώ οι λογικοί τελεστές είναι οι: ΚΑΙ, Η, ΟΧΙ.

Σημειώστε στα παραδείγματα σας, ποια είναι η συνθήκη, και ποιές είναι οι 2 διαφορετικές εντολές ή ομάδα εντολών.

Παράδειγμα 1:

Συνθήκη:.....

Εντολή1 /Ομάδα εντολών1:

Εντολή2 /Ομάδα εντολών2:

Παράδειγμα 2:

Συνθήκη:.....

Εντολή1 /Ομάδα εντολών1:

Εντολή2 /Ομάδα εντολών2:

Παράδειγμα 3:

Συνθήκη:.....

Εντολή1 /Ομάδα εντολών1:

Εντολή2 /Ομάδα εντολών2:

Δραστηριότητα 2

Οδηγίες υλοποίησης

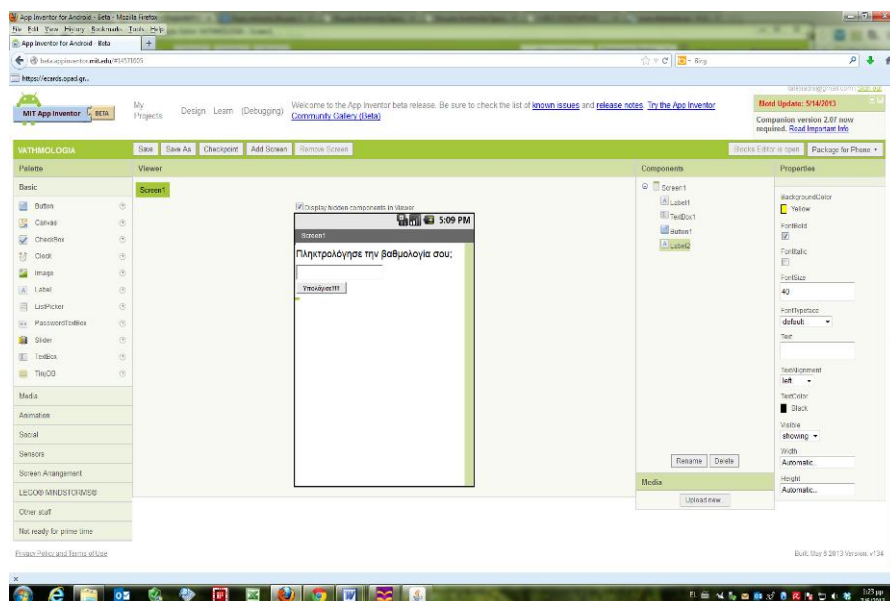
Ας προσπαθήσουμε να υλοποιήσουμε ένα από τα παραπάνω παραδείγματα στο AppInventor. Μιας και είσαστε μαθητές θα υλοποιήσουμε το παράδειγμα με την βαθμολογία: θα δημιουργήσετε μια εφαρμογή η οποία θα ζητάει από τον χρήστη την βαθμολογία του και αν αυτή είναι μεγαλύτερη του 9 θα εμφανίζει το μήνυμα 'Πέρασες την τάξη' αλλιώς θα εμφανίζει το μήνυμα 'Μετεξεταστέος'.

Η μορφή της σύνθετης δομής επιλογής είναι η ακόλουθη:

Αν (συνθήκη) τότε εμφάνισε μήνυμα1 αλλιώς εμφάνισε μήνυμα1

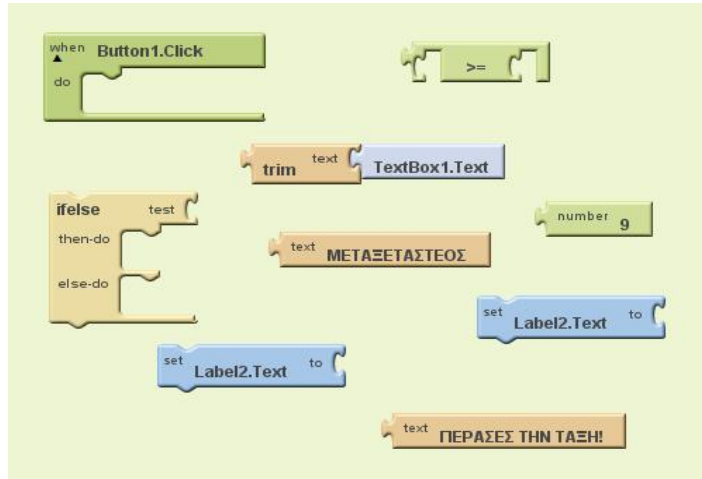
Αν (βαθμολογία > 9) τότε γράψε 'Πέρασες την τάξη' αλλιώς γράψε 'Μετεξεταστέος'.

Η παράσταση (βαθμολογία > 9) είναι η λογική συνθήκη και η εντολή **γράψε 'Άριστος'** θα εκτελεστεί **μόνο όταν** η συνθήκη είναι αληθής. Αντίστοιχα η εντολή **γράψε 'Μετεξεταστέος'** θα εκτελεστεί **μόνο όταν** η συνθήκη είναι ψευδής. Σε καμία περίπτωση δεν θα εκτελεστούν και οι 2 εντολές.



Το οπτικό κομμάτι της εφαρμογής σας θα είναι το ίδιο με την προηγούμενη δραστηριότητα. Αν δε το έχετε ήδη υλοποιήσει ζητήστε από τον καθηγητή σας, να σας δώσει το project (έργο) προκειμένου να ασχοληθείτε με την υλοποίηση της δομής επιλογής στον Block Editor.

Τα πλακίδια που θα χρειαστείτε να συνδέσετε είναι τα :



Στις επόμενες 3 εικόνες, φαίνεται η εφαρμογή όταν θα τρέξει μέσα από τον προσομοιωτή και το αποτέλεσμα που θα προκύψει όταν πληκτρολογήσουμε το νούμερο 20 ή το νούμερο 8.



Δοκιμάστε να πληκτρολογήσετε διάφορα νούμερα, μικρότερα του 1 ή μεγαλύτερα του 20. Τι παρατηρείτε;

.....

.....

Για πιο λόγο νομίζετε ότι συμβαίνει αυτό:

.....

.....

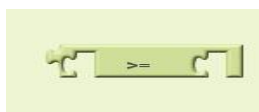
Δραστηριότητα 3

Οδηγίες υλοποίησης

Προσπαθήστε να δημιουργήσετε μια εφαρμογή η οποία θα δέχεται έναν αριθμό και θα εμφανίζει το μήνυμα θετικός ή αρνητικός ανάλογα την τιμή του. Ο αριθμός 0 θεωρείται ότι ανήκει στους θετικούς αριθμούς.

Αν δυσκολεύεστε στην υλοποίηση του οπτικού τμήματος της εφαρμογής σας ζητήστε από τον καθηγητή σας, να σας δώσει το project (έργο) προκειμένου να ασχοληθείτε με την υλοποίηση της δομής επιλογής στον Block Editor.

Ο τελεστής ανισότητας που θα χρησιμοποιήσετε είναι ο :



Αντιμετωπίσατε κάποιο πρόβλημα στην εκτέλεση της εφαρμογής;

.....

.....

20. Φύλλο εργασίας 3 –

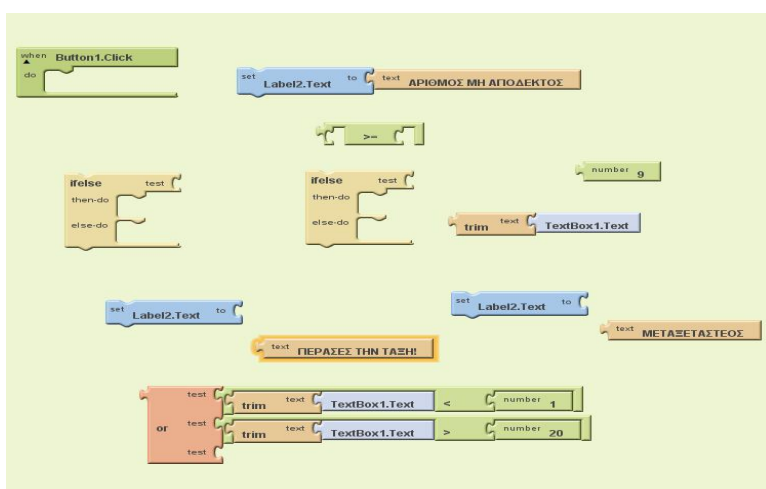
Δραστηριότητα 1

Ας θυμηθούμε λίγο την δραστηριότητα την οποία υλοποιήσατε και σύμφωνα με την οποία αν η βαθμολογία ενός μαθητή είναι μεγαλύτερη του 9 θα εμφανίζει το μήνυμα 'Πέρασες την τάξη' αλλιώς θα εμφανίζει το μήνυμα 'Μετεξεταστέος'. Επίσης γνωρίζετε ότι η κλίμακα βαθμολογίας ενός μαθητή είναι από το 1 έως το 20. Ωστόσο αν εσείς πληκτρολογήσατε τιμές όπως -1, 300 κλπ η εφαρμογή θα σας εμφάνιζε μηνύματα τύπου 'Πέρασες την τάξη' ή 'Μετεξεταστέος', παρότι οι τιμές που δίδεται είναι λάθος διότι δεν μπορεί να υπάρχει τέτοια βαθμολογία του μαθητή.

Πολλές φορές στον προγραμματισμό παρουσιάζονται προβλήματα αυτού του είδους, στα οποία θέλουμε να περιορίσουμε το εύρος κάποιων τιμών προτού ελέγξουμε τις τιμές αυτές. Στην περίπτωση αυτή μπορούμε να χρησιμοποιήσουμε τις λεγόμενες εμφωλευμένες δομές επιλογής δηλαδή να χρησιμοποιήσουμε μια δομή επιλογής εντός μιας άλλης δομής επιλογής κ.ο.κ Το δυνατό βάθος των εμφωλευμένων δομών περιορίζεται μόνο από την φαντασία μας κατά τα στάδια δημιουργίας του αλγορίθμου.

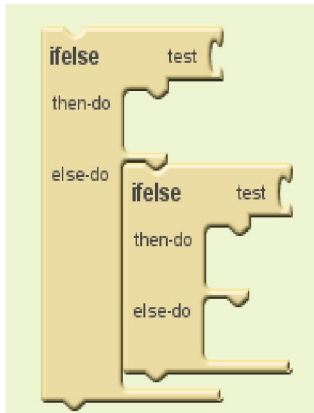
Προκειμένου να βελτιώσετε την εφαρμογή σας, ώστε να μην λαμβάνει υπόψη της βαθμολογία του μαθητή εκτός των αποδεκτών τιμών [1..20], πρέπει προτού ξεκινήσετε να ελέγχετε την βαθμολογία του μαθητή, να διασφαλίσετε ότι η βαθμολογία του είναι εντός των επιτρεπτών ορίων.

Τα πλακίδια που θα χρειαστείς είναι τα ακόλουθα:



Υποδείξεις - οδηγίες:

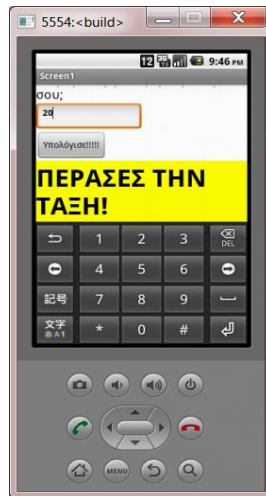
1. Προκειμένου να ελέγξετε ότι οι τιμές βρίσκονται εντός του αποδεκτού εύρους τιμών 1 έως 20 χρειάζεται να χρησιμοποιήσετε το παρακάτω πλακίδιο:



2. η σύνταξη των εμφωλευμένων Αν θα είναι της μορφής:

3. για κάθε AN να εισάγετε πρώτα την συνθήκη και μετά τις αντίστοιχες εντολές.

Αν δημιουργήσετε σωστά την εφαρμογή και την εκτελέσετε διαδοχικά με τις τιμές 80, 20, 3 ο προσομοιωτής θα φαίνεται ως ακολούθως:



Δραστηριότητα 2

Αν υποθέσουμε ότι προκειμένου να καταχωρήσουμε το φύλο ενός ανθρώπου στους υπολογιστές καταχωρούμε την τιμή 1 ή 2 ανάλογα με το αν είναι άρρεν (άνδρας) ή γυναίκα. Δημιουργήστε μια εφαρμογή η οποία θα δέχεται το φύλο ενός ανθρώπου και θα εμφανίζει στην οθόνη το χαρακτηρισμό του (άνδρας ή γυναίκα) ανάλογα με την τιμή που του δόθηκε. Σε περίπτωση καταχώρησης λανθασμένης τιμής δηλαδή διαφορετικής του 1 ή του 2 θα εμφανίζει κατάλληλο μήνυμα λάθους.

Αν δυσκολεύεστε στην υλοποίηση του οπτικού τμήματος της εφαρμογής σας ζητήστε από τον καθηγητή σας, να σας δώσει το project (έργο) προκειμένου να ασχοληθείτε με την υλοποίηση της δομής επιλογής στον Block Editor.

Υπόδειξη: Θα χρησιμοποιήσετε 2 εμφωλευμένες σύνθετες δομές επιλογές. Η εξωτερική συνθήκη θα ελέγχει αν η τιμή είναι έγκυρη (δηλαδή ο χρήστης καταχώρησε την τιμή 1 ή 2) ενώ η εσωτερική δομή θα εκτελείται εφόσον η τιμή είναι έγκυρη. Σε περίπτωση λανθασμένης καταχώρισης, η εφαρμογή σας θα πληροφορεί τον χρήστη με κατάλληλο μήνυμα.

21. Φύλλο εργασίας 4 –

Δραστηριότητα 1

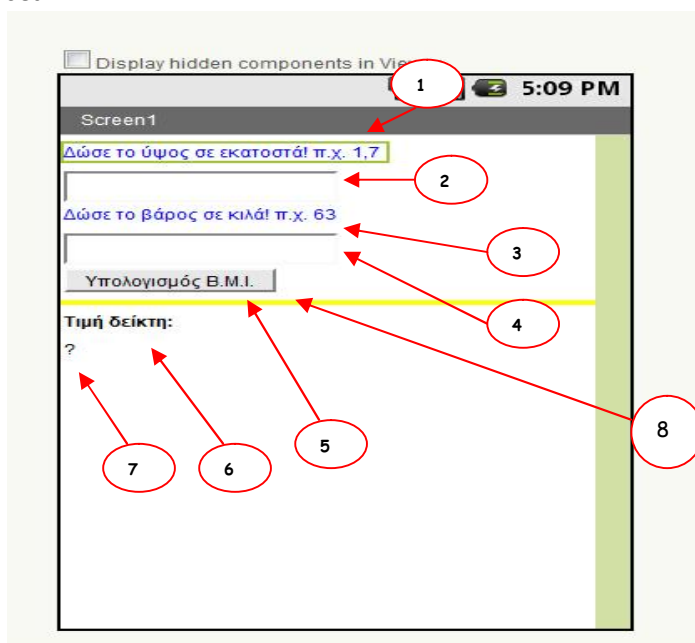
Θα δημιουργήσετε τώρα μια εφαρμογή για τον υπολογισμό του δείκτη μάζας σώματος (BMI). Ο δείκτης μάζας σώματος (BMI, Body Mass Index) είναι ένας εύκολος τρόπος να δούμε αν είμαστε παχύσαρκοι ή όχι. Χρησιμοποιήθηκε για πρώτη φορά περίπου το 1850 και από τότε αποτελεί το πιο ευρέως χρησιμοποιούμενο εργαλείο για τον εντοπισμό προβλημάτων βάρους. Για να υπολογίσουμε το BMI διαιρούμε το σωματικό βάρος με το τετράγωνο του ύψους (kg/m^2) και η αξιολόγηση γίνεται ως εξής :

- BMI μέχρι 18,5 ->Λιποβαρής
- BMI από 18,5 έως 25 ->Κανονικό Βάρος
- BMI από 25,1 - 30 ->Υπέρβαρος
- BMI μεγαλύτερο από 30 ->Παχύσαρκος

Π.χ. για κάποιον που έχει ύψος 1,80 m και βάρος 80 κιλά το BMI του είναι $80/(1,8)^2 = 24,69 \text{ kg}/\text{m}^2$ και είναι στο ιδανικό του βάρος. Ο δείκτης δεν εφαρμόζεται σε παιδιά και εφήβους.

Ας προχωρήσουμε στην υλοποίηση της εφαρμογής.

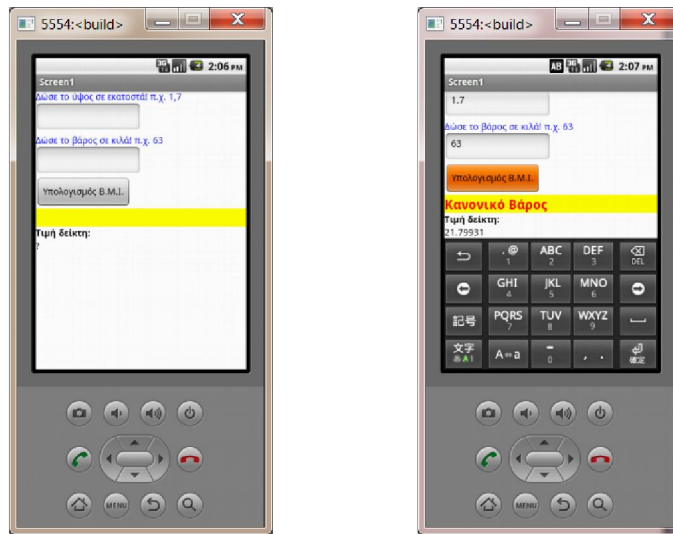
Το οπτικό κομμάτι της εφαρμογής θα πρέπει να μοιάζει σαν αυτό που βλέπετε στην εικόνα που ακολουθεί.



Οι τιμές 1, 3, 5, 6, 7 αναφέρονται σε ετικέτα (label), οι τιμές 2, 4 σε πεδία κειμένου (text box) ενώ η τιμή 8 σε κουμπί (button).

Σε περίπτωση που θεωρείται ότι δεν μπορείτε να ολοκληρώσετε το οπτικό κομμάτι της

εφαρμογής μπορείτε να ζητήσετε από τον καθηγητή σας, να σας δώσει το αρχείο της



εφαρμογής προκειμένου να προχωρήσετε στην σύνταξη των προγραμματιστικών εντολών. Προτού προχωρήσουμε στην σύνταξη των προγραμματιστικών εντολών ας θυμηθούμε ότι στις περισσότερες περιπτώσεις δεν υπάρχει ένας και μόνο αποδεκτός αλγόριθμος για την επίλυση ενός προβλήματος. Παρομοίως στο παράδειγμα μας, τα βήματα για την λύση που θα σας δοθούν είναι απλά ενδεικτικά και ο καθένας από εσάς θα μπορούσε να επιλύσει το πρόβλημα με όποια λύση αυτός/ή θεωρεί αποδεκτή.

Υπόδειξη:

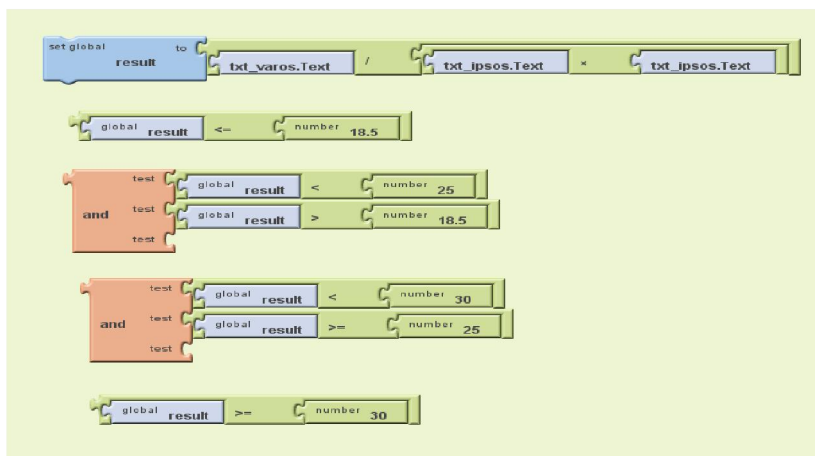
Ένας τρόπος να υλοποιήσουμε τον αλγόριθμο είναι αφού υπολογίσουμε τον δείκτη BMI, να χρησιμοποιήσουμε απλές δομές επιλογής (μία για κάθε κατηγοριοποίηση του BMI) προκειμένου να εμφανίσουμε το κατάλληλο μήνυμα.

Η σύνταξη των συνθηκών που θα χρησιμοποιήσετε και ο υπολογισμός του δείκτη BMI σας παρουσιάζετε παρακάτω:

Επίσης σας παρουσιάζονται διάφορα στιγμιότυπα από την λειτουργία του προσομοιωτή.

Ασκήσεις για περαιτέρω διερεύνηση & προβληματισμό:

Άσκηση 1



Προσπαθήστε να υπολογίσετε τον δείκτη, δίχως να δώσετε τιμές στον αριθμητή (ύψος) ή στον παρανομαστή (βάρος) ή και στα δύο. Τι παρατηρείτε; Γιατί νομίζετε ότι συμβαίνει αυτό; Πως θα μπορούσατε να το αντιμετωπίσετε;

.....
.....
.....

Άσκηση 2

Σκεφτείτε πως θα μπορούσατε να υλοποιήσετε την παραπάνω δραστηριότητα με την χρήση εμφωλευμένων σύνθετων επιλογών αντί για την χρήση μεμονωμένων δομών απλής επιλογής.

Άσκηση 3

Ένας πιο σωστός τρόπος υπολογισμού του δείκτη BMI, θα έπρεπε να λαμβάνει υπόψη του και το φύλο του ατόμου. Παρακάτω σας δίδεται ένας πιο ολοκληρωμένος τρόπος υπολογισμού του BMI. Μπορείτε να τον υλοποιήσετε;

	<i>Άνδρες</i>	<i>Γυναίκες</i>
Λιποσαρκία	< 19,5	< 18,5
Κανονικό βάρος	19,5 - 24,9	18,5 - 23,5
1ος βαθμός παχυσαρκίας	25 - 29,9	23,6 - 28,6
2ος βαθμός παχυσαρκίας	30 - 40	28,7 - 40
3ος βαθμός παχυσαρκίας	> 40	> 40

22. Φύλλο εργασίας 1 - Επανάληψη - Εμπέδωση

Ερωτήσεις κλειστού τύπου

1. Αν θέλουμε να πάρουμε μια απόφαση ανάλογα με την τιμή μιας συνθήκης θα πρέπει να χρησιμοποιήσουμε την δομή επιλογής

Σωστό

Λάθος

2. Η δομή επιλογής διακρίνεται στη δομή απλής, διπλής και τριπλή επιλογής.

Σωστό

Λάθος

3. Δεν μπορούμε να έχουμε μια δομή επιλογής μέσα σε μια άλλη δομή επιλογής

Σωστό

Λάθος

4. Στη δομή σύνθετης επιλογής εκτελούνται υποχρεωτικά οι εντολές που αντιστοιχούν στο τμήμα της συνθήκης που αληθεύει (ισχύει).

Σωστό

Λάθος

5. Μια δομή σύνθετης επιλογής δεν μπορεί να περιέχει μια δομή απλής επιλογής.

Σωστό

Λάθος

6. Μια δομή απλής επιλογής δεν μπορεί να περιέχει μια σύνθετη δομή επιλογής.

Σωστό

Λάθος

23. Προτάσεις για περαιτέρω δραστηριότητες - προτεινόμενες εργασίες

Ολοκληρώστε τη δημιουργία των φύλλων εργασίας που προορίζονται για τους μαθητές, παραλλάσσοντας στοιχεία των προτεινόμενων δραστηριοτήτων.