



ΕΚΠΑΙΔΕΥΤΙΚΟ ΕΓΧΕΙΡΙΔΙΟ

ΒΙΒΛΙΟ ΚΑΘΗΓΗΤΗ

Ανάδοχος Έργου



Κασταμονής 99α & Μακρυγιάννη
142 35 Ν. Ιωνία
τηλ. 210-2719100 fax 210-2718133
url : www.sdc.gr

Το παρόν εκπονήθηκε στο πλαίσιο
του Υποέργου 13 «Προσαρμογή Λογισμικού-Φάση ΙΙΙ»
της Πράξης «Επαγγελματικό λογισμικό στην ΤΕΕ: επιμόρφωση και εφαρμογή»
(Γ' ΚΠΣ, ΕΠΕΑΕΚ, Μέτρο 2.3, Ενέργεια 2.3.2)

που συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση/Ευρωπαϊκό Κοινωνικό Ταμείο

Φορέας Υλοποίησης και Τελικός Δικαιούχος



Υπουργείο Εθνικής Παιδείας και Θρησκευμάτων
Ειδική Υπηρεσία Εφαρμογής Προγραμμάτων ΚΠΣ

Φορέας Λειτουργίας



Υπουργείο Εθνικής Παιδείας και Θρησκευμάτων
Διεύθυνση Σπουδών Δευτεροβάθμιας Εκπαίδευσης-Τμήμα Β'

Επιστημονικός Τεχνικός Σύμβουλος



Ερευνητικό Ακαδημαϊκό Ινστιτούτο Τεχνολογίας Υπολογιστών

Υπεύθυνος Πράξης

2003-2007 Προϊστάμενος Μονάδας Α1-Ειδική Υπηρεσία Εφαρμογής Προγραμμάτων ΚΠΣ-ΥΠΕΠΘ.

2007- Προϊστάμενος Μονάδας Α1β-Ειδική Υπηρεσία Εφαρμογής Προγραμμάτων ΚΠΣ-ΥΠΕΠΘ.



ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΕΑΕΚ



ΕΥΡΩΠΑΪΚΗ ΕΝΩΣΗ
ΣΥΓΧΡΗΜΑΤΟΔΟΤΗΣΗ
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Η ΠΑΙΔΕΙΑ ΣΤΗΝ ΚΟΡΥΦΗ
Επιχειρησιακό Πρόγραμμα
Εκπαίδευσης και Αρχικής
Επαγγελματικής Κατάρτισης

ΠΕΡΙΕΧΟΜΕΝΑ

1. Γνωριμία με το περιβάλλον εργασίας της Visual Basic	4
Δραστηριότητα 1 : Δημιουργία εφαρμογής κονσόλας	6
Δραστηριότητα 2 : Εφαρμογή φορμών.....	7
2. Τα χειριστήρια της Visual Basic – Προγραμματισμός με συμβάντα	10
Δραστηριότητα 1 : Δημιουργία εφαρμογής φορμών	12
Δραστηριότητα 2 : Παίζοντας με τις ιδιότητες.....	16
3. Η έννοια των σταθερών και των μεταβλητών στη Visual Basic	18
Δραστηριότητα 1: σταθερές στην Visual Basic.....	19
Δραστηριότητα 2: Η εντολή καταχώρησης στην Visual Basic	20
Δραστηριότητα 3 : Ορισμός μεταβλητών και τύποι μεταβλητών στη Visual Basic.....	20
Δραστηριότητα 4: οι αριθμητικές μεταβλητές αναλυτικότερα.....	22
Δραστηριότητα 5: πράξεις με αριθμητικές μεταβλητές.....	24
Δραστηριότητα 6 : Άλφαριθμητικές μεταβλητές.....	25
4. Οι συναρτήσεις και υπορουτίνες στη Visual Basic	26
Δραστηριότητα 1: δημιουργία μιας υπορουτίνας	27
Δραστηριότητα 2: Υπορουτίνα με παραμέτρους.....	28
Δραστηριότητα 3: Υπορουτίνες που επιστρέφουν τιμές (συναρτήσεις).....	29
Δραστηριότητα 4: Υπορουτίνες με περισσότερες παραμέτρους.....	29
Δραστηριότητα 5: Κλήση υπορουτίνας από υπορουτίνα.....	30
5. Οι συναρτήσεις και υπορουτίνες στη Visual Basic	32
Δραστηριότητα 1- Το αντικείμενο console.....	33
Δραστηριότητα 2: Το αντικείμενο Math.....	35
Δραστηριότητα 3: Το String σαν αντικείμενο	36
Δραστηριότητα 4: Η κλάση Date.....	37
6. Οι συναρτήσεις και υπορουτίνες στη Visual Basic	39
Δραστηριότητα 1 : Η εντολή If .. then ... else.....	40
Δραστηριότητα 2 : Nested if.....	42
Δραστηριότητα 3 : Οι εντολές AND , OR, NOT.....	42
Δραστηριότητα 4 : Το σύνθετο σχήμα If...Then...Else If.....	43
Δραστηριότητα 5 : Δομή Select Case	44
7. Οι συναρτήσεις και υπορουτίνες στη Visual Basic	46
Δραστηριότητα 1 : Η δομή repeat.....	48
Δραστηριότητα 2 : Η δομή while.....	49
Δραστηριότητα 3 : Η δομή for .. next.....	50
Δραστηριότητα 4 : Οι πίνακες στη Visual Basic	50
8. Οι συναρτήσεις και υπορουτίνες στη Visual Basic	53
Δραστηριότητα 1 : Συντακτικά λάθη.....	54
Δραστηριότητα 2 : Που επηρεάζουν τα συντακτικά λάθη. (Option Explicit).....	55
Δραστηριότητα 3 : Επιλογές που επηρεάζουν τα συντακτικά λάθη. (Option Strict).....	55
Δραστηριότητα 4 : Οι εντολές try ... catch	56
Δραστηριότητα 5 : Λογικά λάθη 1.....	57
Δραστηριότητα 6 : Λογικά λάθη 2.....	57

Σενάριο Εκπαιδευτικών Δραστηριοτήτων -

1. Γνωριμία με το περιβάλλον εργασίας της Visual Basic

Σύντομη περιγραφή

Στο συγκεκριμένο σενάριο οι μαθητές χωρίζονται σε ομάδες των 2 ατόμων. Οι μαθητές με τη δραστηριότητα 1 θα δημιουργήσουν μια απλή εφαρμογή κονσόλας, ενώ με τη δραστηριότητα 2 θα ανοίξουν μια πιο σύνθετη εφαρμογή φορμών, με τη βοήθεια της οποίας θα αποκτήσουν μεγαλύτερη εξοικείωση με το περιβάλλον. Ακολουθώντας τις οδηγίες του φύλου εργασίας, θα ανοίξουν το περιβάλλον εργασίας της Visual Basic. Ένα μέλος της ομάδας ακολουθώντας τις οδηγίες του Φύλου εργασίας, θα χρησιμοποιήσει το περιβάλλον για να δημιουργήσει βήμα βήμα την ζητούμενη εφαρμογή. Το δεύτερο μέλος παρακολουθεί, ελέγχει και συμβουλεύει. Προτείνεται τα δυο μέλη να εναλλάσσουν ρόλους σε κάθε στάδιο της δραστηριότητας.

Γνωστικό αντικείμενο και σύνδεση με το πρόγραμμα σπουδών

Τάξη:

Τομέας:

ΠΛΗΡΟΦΟΡΙΚΗΣ

Γνωστικό Αντικείμενο:

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Διδακτική ενότητα:

1η

Διδακτικές ώρες:

4

Αριθμός μαθητών:

Στο συγκεκριμένο σενάριο οι μαθητές χωρίζονται σε ομάδες των 2 ατόμων. Η επιλογή των ατόμων της κάθε ομάδας θα γίνεται σύμφωνα με τη κρίση του καθηγητή με βασικό κριτήριο την καλύτερη συνεργασία των μαθητών μεταξύ τους.

Διδακτικοί Στόχοι για το Σενάριο

Μετά το πέρας της εκπαιδευτικής δραστηριότητας οι μαθητές θα είναι σε θέση:

- Να μπορούν να δημιουργήσουν ένα νέο έργο (project) με το περιβάλλον εργασίας της Visual Basic
- Να γνωρίσουν τα βασικά παράθυρα που αποτελούν το περιβάλλον εργασίας της Visual Basic
- Να μπορούν να μεταγλωττίσουν (compile) ένα έργο σε Visual Basic, ώστε να δημιουργήσουν μια εκτελέσιμη εφαρμογή.
- Να κατανοήσουν την διαφοροποίηση μεταξύ «εφαρμογών κονσόλας» και «εφαρμογών φορμών» της Visual Basic

Διδακτική προσέγγιση

Για την επίτευξη των στόχων του συγκεκριμένου σεναρίου έχει επιλεγεί η ομαδοσυνεργατική διδακτική προσέγγιση. Οι μαθητές χωρίζονται σε μικρές ομάδες (2 – 3 ατόμων) και με τη βοήθεια του εκπαιδευτικού καλούνται να ολοκληρώσουν τις προτεινόμενες στο φύλλο εργασίας δραστηριότητες. Ο ρόλος του εκπαιδευτικού είναι υποστηρικτικός, καθοδηγώντας τους μαθητές στα σημεία που είναι απαραίτητο και βοηθώντας τους να ανακαλύψουν τη γνώση, συμμετέχοντας ενεργά στην διδακτική διαδικασία.

Εργαλεία ΤΠΕ που αξιοποιούνται στο πλαίσιο του σεναρίου

- Microsoft Visual basic 2008 Express Edition
- Το έργο «drastiriotita1_2»

Βήματα υλοποίησης του σεναρίου

Ο καθηγητής κάνει μια εισαγωγή σχετικά με τα εργαλεία προγραμματισμού και τις ιδιαιτερότητες της Visual Basic .net. Θα πρέπει να αναφερθεί το Microsoft .Net Framework και το Net Framework class library.

Θα πρέπει επίσης να διευκρινιστεί ο ρόλος του προγραμματιστή και ο ρόλος του χρήστη μιας εφαρμογής. Το ζητούμενο προϊόν από τον προγραμματιστή είναι ένα αρχείο .exe που θα το δουλέψει ο χρήστης της εφαρμογής. Το περιβάλλον εργασίας της Visual Basic είναι ένα εργαλείο για τον προγραμματιστή, ενώ το περιβάλλον εργασίας του χρήστη θα το κατασκευάσει ο προγραμματιστής.

Δραστηριότητα 1 : Δημιουργία εφαρμογής κονσόλας

Στάδιο 1

Οι μαθητές καλούνται να δημιουργήσουν ένα νέο έργο (project). Για μια πρώτη επαφή με το περιβάλλον επιλέχτηκε να δημιουργήσουν μια εφαρμογή κονσόλας.

Στάδιο 2

Οι μαθητές καλούνται να εξερευνήσουν τα παράθυρα έργου, ιδιοτήτων και κώδικα.

Στάδιο 3

Οι μαθητές καλούνται να αποθηκεύσουν το έργο που δημιούργησαν. Καλούνται επίσης να εξερευνήσουν τα αρχεία που δημιουργήθηκαν από την Visual Basic. Στόχος είναι να κατανοήσουν πώς ένα έργο αποτελείται από ένα σύνολο αρχείων. Είναι σημαντικό να εξοικειωθούν με την αποθήκευση και ανάκτηση ενός «έργου».

Στάδιο 4

Οι μαθητές θα συμπληρώσουν τον παρακάτω κώδικα

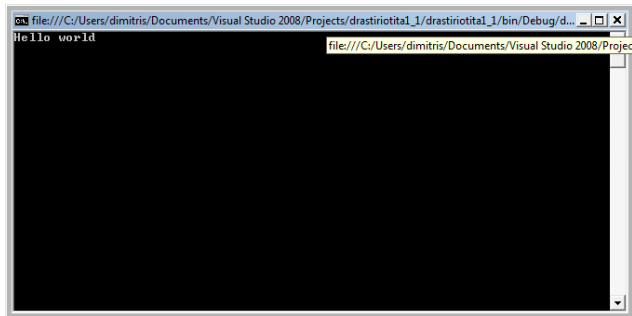
```
System.Console.WriteLine("Hello world")
```

```
System.Console.ReadLine()
```

για να εξοικειωθούν με το παράθυρο κώδικα της Visual Basic.

Εξηγούνται στο φύλλο εργασίας οι δυο προαναφερόμενες εντολές.

Στη συνέχεια, θα μεταγλωττίσουν το έργο ώστε να δημιουργήσουν ένα εκτελέσιμο αρχείο και να τρέξουν την εφαρμογή τους.

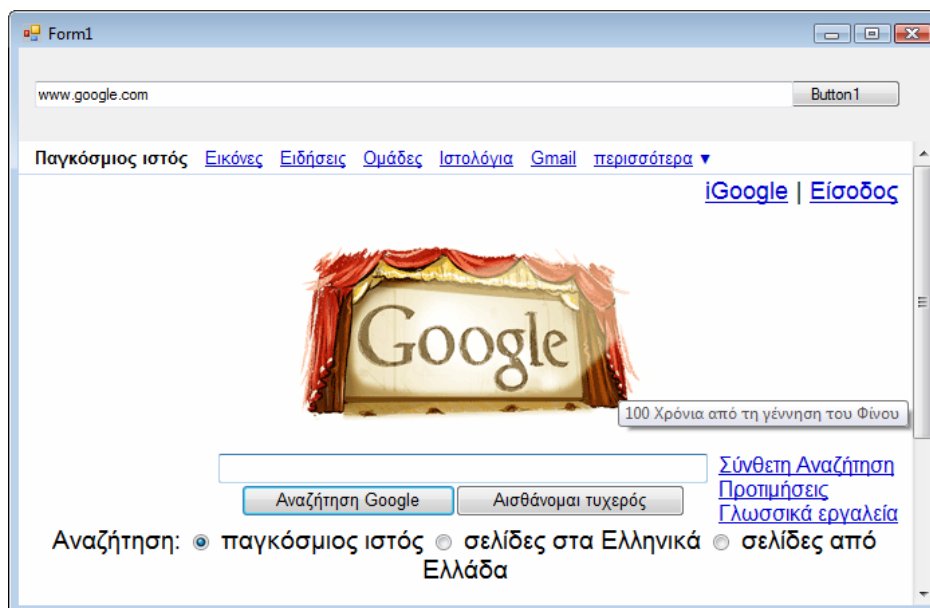


Η εφαρμογή εμφανίζει το μήνυμα «Hello world» στην κονσόλα. Για να τερματιστεί η εφαρμογή ο μαθητής πρέπει να πατήσει «Enter».

Ο μαθητής καλείται να εντοπίσει τα νέα αρχεία που δημιούργησε η μεταγλώττιση (compilation) του προγράμματος. Σημαντικό είναι να εντοπίσει το εκτελέσιμο αρχείο drastiriotita1_1.exe και να εκτελέσει την εφαρμογή έξω από το περιβάλλον της Visual Basic.

Δραστηριότητα 2 : Εφαρμογή φορμών

Στάδιο 1



Ο μαθητής ανοίγει το έργο drastiriotita1_2 από τον φάκελο βοηθητικού υλικού «drastiriotita1_2». Η εφαρμογή είναι αρκετά απλή και εντυπωσιακή. Είναι ένας web browser. Ο μαθητής θα πρέπει πρώτα να δουλέψει την εφαρμογή πηγαίνοντας

τουλάχιστον σε μια σελίδα πχ. www.google.com και μετά να παρατηρήσει στο περιβάλλον της Visual Basic τις διαφοροποιήσεις που υπάρχουν ανάμεσα σε αυτή την εφαρμογή φορμών και την προηγούμενη εφαρμογή κονσόλας. Πρέπει να εντοπίσει τον κώδικα της εφαρμογής με δεξί κλικ πάνω σε οποιοδήποτε σημείο της φόρμας και επιλέγοντας «**View Code**».

Επίσης, πρέπει να εντοπίσει το παράθυρο της εργαλειοθήκης (Toolbox).

Στο φύλλο εργασίας του ζητείται να απαντήσει πώς η εργαλειοθήκη ενεργοποιείται, μόνο όταν είναι επιλεγμένο το TAB σχεδιασμού.

Στη συνέχεια ζητείται από τον μαθητή να περιηγηθεί στο παράθυρο σχεδιασμού της φόρμας. Επιλέγοντας διαδοχικά τα διάφορα αντικείμενα, θα παρατηρήσει το παράθυρο ιδιοτήτων. Θα πρέπει να απαντήσει ότι οι ιδιότητες που αναφέρονται στο παράθυρο ιδιοτήτων αλλάζουν ανάλογα με το αντικείμενο που είναι επιλεγμένο.

Προαιρετικά μπορεί να ζητηθεί από τους μαθητές να εντοπίσουν κοινές ιδιότητες στα τρία αντικείμενα που βρίσκονται πάνω στην φόρμα, όπως οι ιδιότητες «Anchor» και «Dock».

Στάδιο 2

Εδώ ζητείται από τον μαθητή να βάλει breakpoints στην εφαρμογή και να τρέξει τις εντολές που εκτελούνται μετά το πάτημα του κουμπιού «Button 1» μια προς μια. Ταυτόχρονα, πρέπει να έχουν ανοιχτό το παράθυρο τοπικών μεταβλητών και να παρατηρούν την αλλαγή των τιμών στη μεταβλητή *str*.

Στην ερώτηση «τι άλλαξε στο παράθυρο Τοπικών μεταβλητών μετά την εκτέλεση της εντολής *str = Me.TextBox1.Text*» θα πρέπει να απαντήσουν πως η μεταβλητή *str* πήρε σαν τιμή αυτά που πληκτρολόγησε στο πεδίο «TextBox1».

Στη συνέχεια ο μαθητής θα πειραματιστεί με την εντολή *Debug.Print* και το παράθυρο άμεσης εκτέλεσης.

Στην ερώτηση «Τι βλέπετε στο παράθυρο άμεσης εκτέλεσης μετά την εκτέλεση της εντολής *Debug.Print("navigating to: " + str)*» θα πρέπει να απαντήσει πως η παράμετρος της μεθόδου *Debug.Print* εκτυπώθηκε στο παράθυρο άμεσης εκτέλεσης.

Τέλος, οι μαθητές θα πληκτρολογήσουν την εντολή *Debug.Print* απευθείας στο παράθυρο άμεσης εκτέλεσης, ενώ δεν εκτελείται η εφαρμογή. Ζητείται από τον

μαθητή να πειραματιστεί με εντολές όπως *Debug.Print*(«Hello world») και *Debug.Print*(1+1). Τέλος του ζητείται να γράψει μια εντολή που να τυπώνει στο παράθυρο εκτέλεσης την τετραγωνική ρίζα του 2. Η εντολή αυτή είναι *Debug.Print*(*Math.Sqrt*(2)).

Σενάριο Εκπαιδευτικών Δραστηριοτήτων –

2. Τα χειριστήρια της Visual Basic – Προγραμματισμός με συμβάντα

Σύντομη περιγραφή

Στο συγκεκριμένο σενάριο οι μαθητές εργάζονται σε ομάδες των 2 ατόμων. Οι μαθητές με τη δραστηριότητα 1 δημιουργούν μια εφαρμογή φορμών και τοποθετούν αντικείμενα (controls) πάνω στη φόρμα. Σαν παράδειγμα για να εξοικειωθούν με τα αντικείμενα αυτά χρησιμοποιούν το αντικείμενο κουμπί (Button) και το αντικείμενο (Label). Ακολουθώντας τις οδηγίες του φύλλου εργασίας δημιουργούν δυο κουμπιά και τις υπορουτίνες διαχείρισης συμβάντων του πρώτου κουμπιού. Στη συνέχεια, δημιουργούν μόνοι τους τις υπορουτίνες διαχείρισης συμβάντων του δεύτερου κουμπιού. Ακολούθως, δημιουργούν ένα κουμπί του όταν πατηθεί τερματίζει την εφαρμογή.

Η δραστηριότητα 2 είναι προαιρετική και ο εκπαιδευτικός θα κρίνει ανάλογα με το επίπεδο της τάξης αν χρειάζεται να προχωρήσει στην υλοποίησή της. Σε αυτήν την δραστηριότητα οι μαθητές εμβαθύνουν στη διαχείριση συμβάντων χρησιμοποιώντας μια υπορουτίνα διαχείρισης για να χειριστούν δυο συμβάντα από διαφορετικά αντικείμενα.

Ένα μέλος της ομάδας ακολουθώντας τις οδηγίες του Φύλλου εργασίας θα χρησιμοποιήσει το περιβάλλον για να δημιουργήσει βήμα-βήμα τις ζητούμενες εφαρμογές. Το δεύτερο μέλος παρακολουθεί, ελέγχει και συμβουλεύει. Συστήνεται τα δυο μέλη να εναλλάσσουν ρόλους σε κάθε στάδιο της κάθε δραστηριότητας.

Γνωστικό αντικείμενο και σύνδεση με το πρόγραμμα σπουδών

Τάξη:

Τομέας:

ΠΛΗΡΟΦΟΡΙΚΗΣ-

Γνωστικό Αντικείμενο:

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΑ ΕΡΓΑΛΕΙΑ

Διδακτική ενότητα:

2η

Διδακτικές ώρες:

3

Αριθμός μαθητών:

Στο συγκεκριμένο σενάριο οι μαθητές χωρίζονται σε ομάδες των 2 ατόμων. Η επίλογη των ατόμων της

κάθε ομάδα θα γίνεται σύμφωνα με τη κρίση του καθηγητή με βασικό κριτήριο την καλύτερη συνεργασία των μαθητών μεταξύ τους.

Διδακτικοί Στόχοι για το Σενάριο

Μετά το πέρας της εκπαιδευτικής δραστηριότητας οι μαθητές θα είναι σε θέση:

- να δημιουργούν εφαρμογές φορμών
- να σχεδιάζουν αντικείμενα πάνω στις φόρμες
- να χειρίζονται συμβάντα των αντικειμένων με κώδικα
- να τερματίζουν την εφαρμογή χρησιμοποιώντας κώδικα.

Διδακτική προσέγγιση

Για την επίτευξη των στόχων του συγκεκριμένου σεναρίου έχει επιλεγεί η ομαδοσυνεργατική-ανακαλυπτική διδακτική προσέγγιση. Οι μαθητές χωρίζονται σε μικρές ομάδες (2 ατόμων) και με τη βοήθεια του εκπαιδευτικού καλούνται να ολοκληρώσουν τις προτεινόμενες στο φύλλο εργασίας δραστηριότητες. Ο ρόλος του εκπαιδευτικού είναι υποστηρικτικός, καθοδηγώντας τους μαθητές στα σημεία που είναι απαραίτητο και βοηθώντας τους να ανακαλύψουν τη γνώση, συμμετέχοντας ενεργά στην διδακτική διαδικασία.

Εργαλεία ΤΠΕ που αξιοποιούνται στο πλαίσιο του σεναρίου

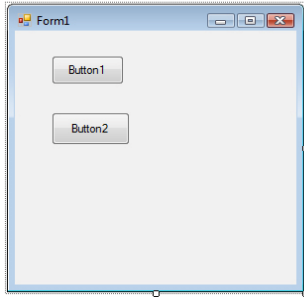
- Microsoft Visual basic 2008 Express Edition
- Το έργο «drastiriotita2_2»

Βήματα υλοποίησης του σεναρίου

Ο καθηγητής θυμίζει στους μαθητές τι είναι μια εφαρμογή φορμών. Καλό είναι να τους δώσει ένα παράδειγμα με μια έτοιμη εφαρμογή φορμών, όπως το calculator και εξηγήσει τα διαφορετικά αντικείμενα που υπάρχουν στη φόρμα του calculator.

Δραστηριότητα 1 : Δημιουργία εφαρμογής φορμών

Στάδιο 1. Δημιουργία μιας φόρμας με αντικείμενα.



Οι μαθητές καλούνται να δημιουργήσουν ένα νέο έργο (project) που θα δημιουργεί μια εφαρμογή φορμών. Στο στάδιο αυτό θα δημιουργήσουν μια απλή φόρμα με δυο κουμπιά.

Όταν τρέξουν την εφαρμογή καλούνται να πατήσουν τα κουμπιά με το ποντίκι και μετά να ανακαλύψουν τρόπους που μπορούν να κάνουν το ίδιο χρησιμοποιώντας το

πληκτρολόγιο. Αυτό γίνεται χρησιμοποιώντας τα κουμπιά «TAB» και «SPACE» του πληκτρολογίου.

Στάδιο 2. Δίνοντας τίτλους στα αντικείμενα

Οι μαθητές καλούνται χρησιμοποιώντας το παράθυρο ιδιοτήτων να αλλάξουν τις τιμές στην ιδιότητα Text των κουμπιών και μετά της φόρμας.

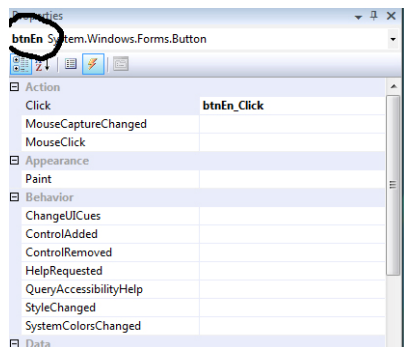
Στην πρώτη άσκηση οι μαθητές πρέπει να παρατηρήσουν πως αλλάζοντας την τιμή της ιδιότητας Text αλλάζει το κείμενο που εμφανίζεται πάνω στο αντικείμενο που βρίσκεται στη φόρμα.

Στη δεύτερη άσκηση ο μαθητής καλείται να εντοπίσει την αλλαγή που γίνεται στον τίτλο της φόρμας όταν αλλάζει η τιμή της ιδιότητας Text στην φόρμα.

Ο καθηγητής θα πρέπει να βοηθήσει τους μαθητές να κατανοήσουν ότι η ιδιότητα Text που είναι κοινή σε πολλά αντικείμενα επηρεάζει τον τρόπο που αυτά εμφανίζονται στο χρήστη της εφαρμογής.

Στάδιο 3. Αλλάζοντας τα ονόματα των αντικειμένων.

Οι μαθητές καλούνται χρησιμοποιώντας το παράθυρο ιδιοτήτων να αλλάξουν τις τιμές στην ιδιότητα όνομα (Name) των αντικειμένων.



Στην πρώτη άσκηση σε αυτό το στάδιο θα πρέπει να παρατηρήσουν πως δεν αλλάζει η εμφάνιση του αντικειμένου αλλάζοντας τιμή σε αυτή την ιδιότητα, αλλά αλλάζει το όνομα του αντικειμένου στο παράθυρο ιδιοτήτων.

Από τον καθηγητή θα πρέπει να τονιστούν οι

συμβάσεις ονοματοδοσίας της Visual Basic που εξασφαλίζουν την αναγνωσιμότητα του κώδικα. Π.χ.

Αντικείμενο	Πρόθεμα	Παράδειγμα
CheckBox	chk	chkPrint
Button	btn	btnOk
Label	lbl	lblMessage
ListBox	lst	lstResults
Panel	pnl	pnlSettings
TextBox	txt	txtAddress

Στάδιο 4. Ελέγχοντας τα συμβάντα (events)

Οι μαθητές καλούνται χρησιμοποιώντας τις δυνατότητες του περιβάλλοντος της Visual Basic και γράφοντας μια γραμμή κώδικα να δημιουργήσουν την υπορουτίνα διαχείρισης του συμβάντος Click του κουμπιού btnEn.

Στην πρώτη άσκηση θα πρέπει να παρατηρήσουν πως πατώντας διπλό κλικ πάνω στο κουμπί η Visual basic έγραψε δυο γραμμές κώδικα που ορίζουν μια υπορουτίνα.

Στη συνέχεια τους εξηγείται η δήλωση «Handles» που δηλώνει αυτή την υπορουτίνα ως υπορουτίνα διαχείρισης συμβάντος.

Τέλος, υποδεικνύεται ο κώδικας που θα εμφανίζει το μήνυμα «Hello world» στη φόρμα όταν πατηθεί το κουμπί btnEn.

Στάδιο 5. Γράφοντας κώδικα για ένα συμβάν

Εδώ οι μαθητές καλούνται να επαναλάβουν μόνοι τους όσα έκαναν στο προηγούμενο στάδιο εφαρμόζοντάς τα στο κουμπί btnE1 της φόρμας.

Ο κώδικας που θα πρέπει να γράψουν έχει ως εξής:

```
Private Sub btnE1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnE1.Click
    Me.lblMessage.Text = "Γειά σου Κόσμε"
End Sub
```

Στάδιο 6. Τερματισμός της εφαρμογής.

Εδώ μαθαίνουν την απλή εντολή Application.Exit() που τερματίζει την εφαρμογή. Με τις οδηγίες του φύλλου εργασίας δημιουργούν πάλι ένα νέο κουμπί το btnExit και έναν διαχειριστή συμβάντων του κουμπιού αυτού που θα τερματίζει την εφαρμογή μόλις πατηθεί το btnExit.

Παρατίθεται ο κώδικας του διαχειριστή συμβάντος του κουμπιού αυτού:

```
Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnExit.Click
    Application.Exit()
End Sub
```

Στάδιο 7. Διαχείριση και άλλων συμβάντων.

Οι μαθητές με τις οδηγίες του φύλλου εργασίας μαθαίνουν να διαχειρίζονται και άλλα συμβάντα, τα συμβάντα MouseEnter και MouseLeave του κουμπιού btnEn. Το φύλλο εργασίας τους καθοδηγεί να δημιουργήσουν την υπορουτίνα διαχείρισης του πρώτου συμβάντος (MouseEnter) χρησιμοποιώντας τις δυνατότητες του περιβάλλοντος της Visual Basic και γράφοντας μια γραμμή κώδικα. Για την δημιουργία της υπορουτίνας διαχείρισης του δεύτερου συμβάντος (MouseLeave) το φύλλο εργασία καθοδηγεί τους μαθητές έτσι ώστε να τη δημιουργήσουν απευθείας στο παράθυρο κώδικα. Η τελική μορφή της υπορουτίνας είναι η εξής:

```
Private Sub btnEn_MouseLeave(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnEn.MouseLeave
    Me.lblMessageHelp.Text = ""
End Sub
```

Στάδιο 8. Γράφοντας ξανά κώδικα για συμβάντα.

Οι μαθητές καλούνται να δημιουργήσουν τις υπορουτίνες διαχείρισης των συμβάντων MouseEnter και MouseLeave του κουμπιού btnEl. Μπορούν να τις δημιουργήσουν με οποιονδήποτε από τους δύο τρόπους που έμαθαν παραπάνω. Η τελική μορφή του κώδικα που καλούνται να γράψουν έχει ως εξής:

```
Private Sub btnEl_MouseEnter(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnEl.MouseEnter
    Me.lblMessageHelp.Text = "Πατηστε για να δειτε ενα μηνυμα στα Ελληνικά"
End Sub
```

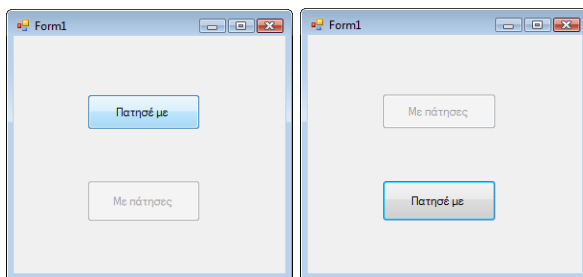
```
Private Sub btnEl_MouseLeave(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles btnEl.MouseLeave  
    Me.lblMessageHelp.Text = ""  
End Sub
```

Δραστηριότητα 2 : Παίζοντας με τις ιδιότητες

Στάδιο 1. Δημιουργία εφαρμογής με δυο διαχειριστές συμβάντων

Σε αυτό το στάδιο οι μαθητές επαναλαμβάνοντας αυτά που έχουν μάθει μέχρι τώρα δημιουργούν μια νέα εφαρμογή φορμών με δυο κουμπιά. Το φύλλο εργασίας τους καθοδηγεί να αλλάξουν με κώδικα την ιδιότητα *enabled* των κουμπιών μέσα από την υπορουτίνα διαχείρισης συμβάντος κάθε κουμπιού. Με την ιδιότητα *enabled* έρχονται σε μια πρώτη επαφή με τις λογικές μεταβλητές της Visual Basic.

Στην εφαρμογή θα πρέπει να εναλλάσσονται οι δυο παρακάτω καταστάσεις.



Επίσης, μαθαίνουν την παράμετρο *sender* που χρησιμοποιούν οι υπορουτίνες διαχείρισης συμβάντων.

Ο μαθητής πρέπει να απαντήσει στην άσκηση ότι δεν υπάρχει καμία διαφορά στην εκτέλεση του προγράμματος, είτε χρησιμοποιήσουμε την παράμετρο *sender*, είτε το όνομα του αντικείμενου που προκάλεσε το συμβάν.

Αυτό το στάδιο είναι προετοιμασία για το επόμενο.

Στάδιο 2. Διαχείριση πολλών συμβάντων με την ίδια υπορουτίνα

Σε αυτό το στάδιο οι μαθητές καλούνται να ανοίξουν ένα έτοιμο έργο το *drastiriotita2_2*. Αυτό το έργο έχει ακριβώς την ίδια συμπεριφορά με το έργο που δημιουργήθηκε στο προηγούμενο στάδιο. Υπάρχει όμως μια διαφοροποίηση στον κώδικα. Αντί για δυο υπορουτίνες διαχείρισης συμβάντων υπάρχει μόνο μια που διαχειρίζεται και τα δυο συμβάντα χρησιμοποιώντας το αντικείμενο *sender* για να επηρεάσει την εμφάνιση του κουμπιού που πατήθηκε και προκάλεσε το συμβάν.

Οι μαθητές καλούνται να ξαναγράψουν τη εφαρμογή που έγραψαν στο προηγούμενο στάδιο με τον νέο αυτό τρόπο.

Εναλλακτικά μπορείτε να συζητήσετε τα πλεονεκτήματα και τα μειονεκτήματα της διαχείρισης πολλών συμβάντων με την ίδια υπορουτίνα. Για παράδειγμα, αναφερθείτε την εφαρμογή του calculator. Στην εφαρμογή αυτή υπάρχουν δέκα κουμπιά με τα νούμερα μηδέν έως εννέα. Τα κουμπιά αυτά κάνουν σχεδόν την ίδια δουλειά. Θα ήταν προτιμότερο να φτιαχτεί μια υπορουτίνα για κάθε κουμπί (δηλαδή δέκα υπορουτίνες συνολικά) ή μια υπορουτίνα που να διαχειρίζεται το συμβάν Click και για τα δέκα κουμπιά;

Σενάριο Εκπαιδευτικών Δραστηριοτήτων –

3. Η έννοια των σταθερών και των μεταβλητών στη Visual Basic

Σύντομη περιγραφή

Στο συγκεκριμένο σενάριο οι μαθητές χωρίζονται σε ομάδες των 2-3 ατόμων. Οι μαθητές εκτελούν τις δραστηριότητες του φύλλου εργασίας, έτσι ώστε να κατανοήσουν την έννοια των μεταβλητών και την έννοια της καταχώρησης στη Visual Basic.

Γνωστικό αντικείμενο και σύνδεση με το πρόγραμμα σπουδών

Τάξη:

Τομέας: ΠΛΗΡΟΦΟΡΙΚΗΣ-

Γνωστικό Αντικείμενο: ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΑ ΕΡΓΑΛΕΙΑ

Διδακτική ενότητα: 3η

Διδακτικές ώρες: 3

Αριθμός μαθητών: Στο συγκεκριμένο σενάριο οι μαθητές χωρίζονται σε ομάδες των 2-3 ατόμων. Η επιλογή των ατόμων της κάθε ομάδας θα γίνεται σύμφωνα με τη κρίση του καθηγητή με βασικό κριτήριο την καλύτερη συνεργασία των μαθητών μεταξύ τους.

Διδακτικοί Στόχοι για το Σενάριο

Μετά το πέρας της εκπαιδευτικής δραστηριότητας οι μαθητές θα:

- γνωρίζουν τους τύπους των μεταβλητών και των σταθερών στη Visual Basic
- μπορούν να δηλώνουν μεταβλητές σε ένα πρόγραμμα σε Visual Basic
- είναι ικανοί να γράφουν εντολές καταχώρησης σε Visual Basic
- είναι ικανοί να επιλέγουν το κατάλληλο τύπο μεταβλητής για να αποθηκεύει τις τιμές που χρειάζεται το πρόγραμμα.
- είναι ικανοί να βάζουν σχόλια σε ένα πρόγραμμα Visual Basic

Διδακτική προσέγγιση

Για την επίτευξη των στόχων του συγκεκριμένου σεναρίου έχει επιλεγεί η ομαδοσυνεργατική διδακτική προσέγγιση. Οι μαθητές χωρίζονται σε μικρές ομάδες (2-3 ατόμων) και με τη βοήθεια του εκπαιδευτικού καλούνται να ολοκληρώσουν τις προτεινόμενες στο φύλλο εργασίας δραστηριότητες. Ο ρόλος του εκπαιδευτικού είναι υποστηρικτικός, καθοδηγώντας τους μαθητές στα σημεία που είναι απαραίτητο και βοηθώντας τους να ανακαλύψουν τη γνώση, συμμετέχοντας ενεργά στην διδακτική διαδικασία.

Εργαλεία ΤΠΕ που αξιοποιούνται στο πλαίσιο του σεναρίου

- Microsoft Visual basic 2008 Express Edition

Βήματα υλοποίησης του σεναρίου

Ο καθηγητής θυμίζει στους μαθητές το ρόλο των μεταβλητών και των σταθερών στη Visual Basic. Επίσης, καλό είναι να τους θυμίσει πώς γράφεται ένα πρόγραμμα σε ψευδοκώδικα. Οι δραστηριότητες του σεναρίου υλοποιούνται στο φύλλο εργασίας. Προαιρετικά ο καθηγητής μπορεί να ζητήσει από τους μαθητές να δημιουργήσουν μια εφαρμογή κονσόλας και, αφού λύσουν τις ασκήσεις που έχουν κώδικα στο φύλλο εργασίας, να πληκτρολογήσουν τις εντολές στην εφαρμογή. Μετά βάζοντας breakpoints θα μπορούν να παρακολουθήσουν τις τιμές των μεταβλητών στο παράθυρο τοπικών μεταβλητών.

Δραστηριότητα 1: σταθερές στην Visual Basic

Στο φύλλο εργασίας εξηγούνται στους μαθητές οι τρεις τύποι σταθερών (αριθμητικές, λογικές, αλφαριθμητικές)

Ακολούθως του ζητείται να κατηγοριοποιήσουν τις σταθερές στον επόμενο πίνακα σε αριθμητικές, αλφαριθμητικές και λογικές.

18	αριθμητική
“2104896520”	αλφαριθμητική

“A9”	αλφαριθμητική
9.30	αριθμητική
true	λογική
10/2	αριθμητική

Δραστηριότητα 2: Η εντολή καταχώρησης στην Visual Basic

Οι μαθητές μαθαίνουν με παραδείγματα την εντολή καταχώρισης. Μετά θα πρέπει να απαντήσουν στην παρακάτω άσκηση.

Παρακολουθήστε τις παρακάτω εντολές και στο τέλος γράψτε τι τιμή θα έχει η μεταβλητή c:

a = 3

b = 4

c = b ' εδώ παίρνει την τιμή 4

c = a ' εδώ παίρνει την τιμή 3

c = c ' εδώ παίρνει την τιμή 3

Η μεταβλητή c θα έχει την τιμή 3.

Ο εκπαιδευτικός θα πρέπει να ενθαρρύνει τους μαθητές να διατυπώσουν σε ψευδοκώδικα τις εντολές καταχώρησης, έτσι ώστε να κατανοήσουν καλύτερα την λειτουργία των εντολών.

Δραστηριότητα 3 : Ορισμός μεταβλητών και τύποι μεταβλητών στη Visual Basic

Στο φύλλο εργασίας εξηγείται στους μαθητές πώς γίνεται ο ορισμός των μεταβλητών στη Visual Basic και δίνεται το παρακάτω παράδειγμα (σε παρένθεση παρατίθεται ο τρόπος που διαβάζεται η εντολή δήλωσης)

Dim num As Integer (δημιούργησε μια μεταβλητή, στην οποία θα αποθηκεύονται ακέραιες τιμές)

Dim str As String (δημιούργησε μια μεταβλητή, στην οποία θα αποθηκεύονται αλφαριθμητικές τιμές)

Dim bool As Boolean (δημιούργησε μια μεταβλητή, στην οποία θα αποθηκεύονται λογικές τιμές)

Στην παρακάτω άσκηση οι μαθητές καλούνται να αναγνωρίσουν αν οι καταχωρίσεις σταθερών στις μεταβλητές είναι σωστές (σύμφωνα με τον τύπο της μεταβλητής).

Εντοπίστε αν οι παρακάτω καταχωρήσεις καταχωρούν τιμές σύμφωνα με τον τύπο της μεταβλητής και γιατί (θεωρήστε τις τρεις δηλώσει μεταβλητών που δώσαμε σαν παράδειγμα πιο πάνω).

num = 100	Σωστή. Το 100 είναι ακέραιος (integer)
num = 12.10	Λάθος. Το 12.10 είναι δεκαδικός
str = 100	Λάθος. Το 100 είναι ακέραιος ενώ το str αλφαριθμητικό
str = “είμαι μαθητής”	Σωστή. Καταχώρηση αλφαριθμητικής σταθεράς σε αλφαριθμητική μεταβλητή
str = “18.30”	Σωστή. Το 18.30 είναι δεκαδικός
bool = 217	Λάθος. Καταχώρηση ακέραιου σε λογική μεταβλητή
bool = false	Σωστή. Καταχώρηση λογικής σταθεράς σε λογική μεταβλητή

Προαιρετικά οι μαθητές μπορούν να πληκτρολογήσουν τον κώδικα σε μια εφαρμογή κονσόλας της Visual Basic. Παρατίθεται ο συνολικός κώδικας:

Option Strict On

Module Module1

Sub Main()

Dim num **As Integer**

Dim str **As String**

Dim bool **As Boolean**

num = 100

num = 12.1

str = 100

str = "είμαι μαθητής"

str = "18.30"

bool = 217

bool = False

End Sub

End Module

Καλέστε του μαθητές να πειραματιστούν με την επιλογή «Option Strict On».

Δραστηριότητα 4: οι αριθμητικές μεταβλητές αναλυτικότερα

Εδώ εξηγείται στους μαθητές το εύρος τιμών των αριθμητικών μεταβλητών. Παρακάτω παρατίθεται ο πίνακας με όλους τους τύπους αριθμητικών μεταβλητών της Visual Basic.

Οι μαθητές καλούνται να απαντήσουν στην παρακάτω άσκηση:

Byte	0 through 255 (unsigned)	ακέραιες	Όχι
Decimal	0 through +/- 79,228,162,514,264,337,593,543,950,335 (+/- 7.9...E+28) [†] with no decimal point; 0 through +/- 7.9228162514264337593543950335 with 28 places to the right of the decimal; smallest nonzero number is +/- 0.000000000000000000000000000001 (+/-1E-28) [†]	δεκαδικές	ναι
Double (double- precision floating- point)	-1.79769313486231570E+308 through - 4.94065645841246544E-324 [†] for negative values; 4.94065645841246544E-324 through 1.79769313486231570E+308 [†] for positive values	δεκαδικές	ναι

Integer	-2,147,483,648 through 2,147,483,647 (signed)	ακέραιες	ναι
Long (long integer)	-9,223,372,036,854,775,808 through 9,223,372,036,854,775,807 (9.2...E+18 †) (signed)	ακέραιες	ναι
SByte	-128 through 127 (signed)	ακέραιες	ναι
Short (short integer)	-32,768 through 32,767 (signed)	ακέραιες	ναι
Single (single-precision floating-point)	-3.4028235E+38 through -1.401298E-45 † for negative values; 1.401298E-45 through 3.4028235E+38 † for positive values	δεκαδικές	ναι
UInteger	0 through 4,294,967,295 (unsigned)	ακέραιες	όχι
ULong	0 through 18,446,744,073,709,551,615 (1.8...E+19 †) (unsigned)	ακέραιες	όχι
UShort	0 through 65,535 (unsigned)	ακέραιες	όχι

Σημειώστε στην τρίτη στήλη αν οι τιμές που μπορεί να πάρει κάθε τύπος είναι ακέραιες ή δεκαδικές.

Μετά σημειώστε με ένα ναι ή ένα όχι στην τελευταία στήλη αν ο τύπος μπορεί να πάρει αρνητικές τιμές (π.χ -2)

Προτείνεται οι μαθητές να ανοίξουν μια εφαρμογή κονσόλας της Visual Basic και, αφού δηλώσουν «[Option Strict On](#)», να φτιάξουν παραδείγματα με τους παραπάνω τύπους μεταβλητών.

Π.χ.

[Option Strict On](#)

Module Module1

Sub Main()

Dim myByte As Byte

myByte = 300

myByte = -10

myByte = 200

End Sub

End Module

Δραστηριότητα 5: πράξεις με αριθμητικές μεταβλητές

Στη δραστηριότητα αυτή οι μαθητές μαθαίνουν να δημιουργούν αριθμητικές παραστάσεις με την Visual Basic και να καταχωρούν το αποτέλεσμα τους σε μεταβλητές.

Δίνονται παραδείγματα και μετά ζητείται να λύσουν την παρακάτω άσκηση.

Στις παρακάτω γραμμές κώδικα συμπληρώστε στα σχόλια τις τιμές που παίρνουν οι μεταβλητές που ζητούνται μετά την εκτέλεση της εντολής.

Dim num As Integer

Dim int As Integer

num = 10 ' num : ..10..

num = num + 1 ' num : ..11..

int = 9 ' num : ..11.. , int : ..9...

num = int \ 3 ' num : ...3.. , int : ..9...

int = num + int ' num : ...3.. , int : ..12..

num = int Mod 5 ' num : ...2.. , int : ..12..

Μετά τη λύση της άσκησης καλό θα είναι οι μαθητές να πληκτρολογήσουν τις εντολές σε μια εφαρμογή κονσόλας της Visual Basic και να ελέγξουν την ορθότητα των απαντήσεών τους, βάζοντας breakpoints και παρακολουθώντας τις τιμές των μεταβλητών από το παράθυρο τοπικών μεταβλητών.

Δραστηριότητα 6 : Αλφαριθμητικές μεταβλητές

Στη δραστηριότητα αυτή εξηγείται στους μαθητές η δήλωση των αλφαριθμητικών μεταβλητών και δίνονται παραδείγματα χρήσης τους.

Επίσης, εξηγείται η πράξη της σύνθεσης (concatenation) αλφαριθμητικών τιμών. Οι μαθητές μπορούν να τα πληκτρολογήσουν σε μια εφαρμογή κονσόλας της Visual Basic.

Τέλος, ζητείται από του μαθητές να λύσουν την παρακάτω άσκηση:

Συμπληρώστε παρακάτω στα σχόλια τις τιμες που θα πάρει η μεταβλητή str μετα την εκτέλεση της κάθε εντολής

Dim str As String

str = "Hello " + "World" ' _____ "Hello World" _____

str = str & "." ' _____ "Hello World." _____

str = "Hello+" & "world" ' _____ "Hello+world" _____

str = "Hello & World" ' _____ "Hello & World" _____

str = "Hello & " + "World" ' _____ "Hello & World" _____

Σενάριο Εκπαιδευτικών Δραστηριοτήτων –

4. Οι συναρτήσεις και υπορουτίνες στη Visual Basic

Σύντομη περιγραφή

Στο συγκεκριμένο σενάριο οι μαθητές χωρίζονται σε ομάδες των 2 ατόμων. Ακολουθώντας τις οδηγίες του φύλλου εργασίας, θα πρέπει να απαντήσουν στην ερώτηση που βρίσκεται στο τέλος κάθε δραστηριότητας. Ωστόσο, για να κατανοήσουν την ερώτηση θα πρέπει πρώτα να έχουν γράψει τον κώδικα που δίνεται στο φύλλο εργασίας σε μια εφαρμογή κονσόλας της Visual Basic. Ένα μέλος της ομάδας θα χρησιμοποιήσει το περιβάλλον για να πληκτρολογήσει τον κώδικα. Το δεύτερο μέλος παρακολουθεί, ελέγχει και συμβουλεύει. Προτείνεται τα δυο μέλη να εναλλάσσουν ρόλους σε κάθε στάδιο της δραστηριότητας.

Γνωστικό αντικείμενο και σύνδεση με το πρόγραμμα σπουδών

Τάξη:

Τομέας: ΠΛΗΡΟΦΟΡΙΚΗΣ-

Γνωστικό Αντικείμενο: ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΑ ΕΡΓΑΛΕΙΑ

Διδακτική ενότητα:

Διδακτικές ώρες: 3

Αριθμός μαθητών: Στο συγκεκριμένο σενάριο οι μαθητές χωρίζονται σε ομάδες των 2 ατόμων. Η επιλογή των ατόμων της κάθε ομάδας θα γίνεται σύμφωνα με τη κρίση του καθηγητή με βασικό κριτήριο την καλύτερη συνεργασία των μαθητών μεταξύ τους.

Διδακτικοί Στόχοι για το Σενάριο

Μετά το πέρας της εκπαιδευτικής δραστηριότητας οι μαθητές θα είναι ικανοί να:

- αναγνωρίζουν τους τύπους των μεταβλητών και των σταθερών στη Visual Basic
- δηλώνουν μεταβλητές σε ένα πρόγραμμα σε Visual Basic
- γράφουν εντολές καταχώρησης σε Visual Basic

- επιλέγουν το κατάλληλο τύπο μεταβλητής για να αποθηκεύουν τις τιμές που χρειάζεται το πρόγραμμα.
- προσθέτουν σχόλια σε ένα πρόγραμμα Visual Basic

Διδακτική προσέγγιση

Για την επίτευξη των στόχων του συγκεκριμένου σεναρίου έχει επιλεγεί η ομαδοσυνεργατική-ανακαλυπτική διδακτική προσέγγιση. Οι μαθητές χωρίζονται σε μικρές ομάδες (2 ατόμων) και με τη βοήθεια του εκπαιδευτικού καλούνται να ολοκληρώσουν τις προτεινόμενες στο φύλλο εργασίας δραστηριότητες. Ο ρόλος του εκπαιδευτικού είναι υποστηρικτικός, καθοδηγώντας τους μαθητές στα σημεία που είναι απαραίτητο και βοηθώντας τους να ανακαλύψουν τη γνώση, συμμετέχοντας ενεργά στην διδακτική διαδικασία.

Εργαλεία ΤΠΕ που αξιοποιούνται στο πλαίσιο του σεναρίου

- Microsoft Visual basic 2008 Express Edition

Βήματα υλοποίησης του σεναρίου

Ο καθηγητής θυμίζει στους μαθητές το ρόλο των υπορουτίνων και των συναρτήσεων στον προγραμματισμό. Επίσης, καλό είναι να γίνει μια εισαγωγή σχετικά με τις τοπικές και καθολικές μεταβλητές (local variables, global variables). Οι δραστηριότητες του σεναρίου υλοποιούνται στο φύλλο εργασίας, αλλά χρησιμοποιείται το περιβάλλον εργασίας της Visual Basic για να κατανοήσουν την λειτουργία υπορουτίνων και συναρτήσεων.

Δραστηριότητα 1: δημιουργία μιας υπορουτίνας

Στη δραστηριότητα αυτή δίνεται αρχικά ένα παράδειγμα κώδικα με τη μορφή εφαρμογής κονσόλας. Ο κώδικας αυτός καταχωρεί σε μια μεταβλητή, την *num*, τον περιοδικό αριθμό 3.3... Μετά καταχωρεί στην ίδια μεταβλητή τον ίδιο αριθμό στρογγυλοποιημένο στα δυο δεκαδικά ψηφία. Οι μαθητές καλούνται να πληκτρολογήσουν τον κώδικα και να παρατηρήσουν το αποτέλεσμα.

```

C:\> file:///C:/Users/dimitris/Desktop/vb lessons...
3,3333333333333333
3,33
    
```

Ακολουθώντας, τους δίνεται ένας κώδικας που έχει το ίδιο αποτέλεσμα αλλά χρησιμοποιεί μια υπορουτίνα και μια καθολική μεταβλητή (*num*).

Στη δραστηριότητα που δίνεται, οι μαθητές καλούνται να μεταφέρουν τη δήλωση της μεταβλητής μέσα στην κεντρική υπορουτίνα του προγράμματος μετατρέποντάς την σε τοπική μεταβλητή. Καλούνται να παρατηρήσουν πως το περιβάλλον εργασίας της Visual Basic επισημαίνει το λάθος βγάζοντας το μήνυμα «Name 'num' is not declared», όταν χρησιμοποιείται η μεταβλητή *num* μέσα στην υπορουτίνα.

Σε αυτό το σημείο είναι καλό να αναλυθεί ξανά από τον εκπαιδευτικό η έννοια των καθολικών και τοπικών μεταβλητών. Επίσης, μπορεί να αναλυθεί η δομή μιας εφαρμογής κονσόλας. Οι εφαρμογές κονσόλας έχουν μια κεντρική υπορουτίνα την «Sub Main(»). Από αυτή ξεκινάει η εκτέλεση του προγράμματος. Οι μεταβλητές που δηλώνονται μέσα στην «Sub Main(» είναι τοπικές. Οι καθολικές μεταβλητές δηλώνονται εκτός υπορουτινών.

Δραστηριότητα 2: Υπορουτίνα με παραμέτρους.

Στη δραστηριότητα αυτή τροποποιείται ο παραπάνω κώδικας έτσι ώστε η υπορουτίνα να κληθεί με παραμέτρους. Οι μαθητές καλούνται να προσέξουν ότι η αλλαγή της δήλωσης της μεταβλητής «*num*» από καθολική σε τοπική και αντίστροφα δεν δημιουργεί πρόβλημα. Ωστόσο, όταν εκτελεστεί το πρόγραμμα, το αποτέλεσμα δεν είναι το επιθυμητό.

```

C:\> file:///C:/Users/dimitris/Desktop...
3,3333333333333333
3,3333333333333333
    
```

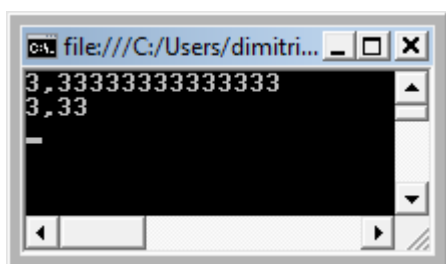
Οι μαθητές καλούνται να ανακαλύψουν τον λόγο. Για την υλοποίηση αυτής της δραστηριότητας θα ήταν χρήσιμο οι μαθητές να βάλουν breakpoints στον κώδικα και

από το παράθυρο τοπικών μεταβλητών να παρατηρήσουν την εξέλιξη των τιμών. Στην ερώτηση του φύλλου εργασίας θα πρέπει να απαντήσουν πως, ενώ η τιμή της μεταβλητής «theNum» μέσα στην υπορουτίνα μεταβάλλεται, η τιμή της μεταβλητής «num» που περάστηκε σαν παράμετρος δεν μεταβλήθηκε.

Δραστηριότητα 3: Υπορουτίνες που επιστρέφουν τιμές (συναρτήσεις).

Μετά το πρόβλημα που τέθηκε από την προηγούμενη δραστηριότητα, δηλαδή πως μια υπορουτίνα μπορεί να επιστρέψει μια τιμή, παρουσιάζονται στους μαθητές οι συναρτήσεις της Visual Basic. Θα πρέπει να τονιστεί στους μαθητές ότι οι συναρτήσεις έχουν και τύπο (π.χ. Double).

Δίνεται τροποποιημένος ο κώδικας που δόθηκε πριν. Αντί για υπορουτίνα, αυτή τη φορά χρησιμοποιείται συνάρτηση. Η κλήση της γίνεται με την εντολή καταχώρησης «num1 = calculate2DecimalPlaces(num1)». Αν τρέξουμε το πρόγραμμα το αποτέλεσμα είναι πλέον το επιθυμητό.



Ο καθηγητής θα πρέπει να καθοδηγεί τους μαθητές ώστε να κατανοήσουν πως η εντολή «return» δεν τερματίζει απλώς την εκτέλεση της συνάρτησης, αλλά επιστρέφει και την τιμή της συνάρτησης. Άρα, η τιμή που θα επιστρέψει η εντολή «return» θα πρέπει να είναι ίδιου τύπου με τον τύπο της συνάρτησης.

Τέλος, ζητείται από τους μαθητές να γράψουν την παρακάτω συνάρτηση.

```
Function piValue() As Double
    Return 3.1415926535
End Function
```

Δραστηριότητα 4: Υπορουτίνες με περισσότερες παραμέτρους.

Στη δραστηριότητα αυτή δίνεται τροποποιημένη η συνάρτηση της προηγούμενης δραστηριότητας. Αυτή τη φορά έχει δυο παραμέτρους. Η πρώτη είναι πάλι ο αριθμός που θέλουμε να στρογγυλοποιήσουμε και η δεύτερη είναι ο αριθμός των δεκαδικών ψηφίων. Σαν παράδειγμα δίνεται η κλήση της συνάρτησης με τις εξής παραμέτρους: num1 = calculateNDecimalPlaces(num1, 0).

Ο εκπαιδευτικός θα πρέπει να ενθαρρύνει τους μαθητές να καλέσουν την συνάρτηση με διαφορετικές παραμέτρους, π.χ

```
num1 = calculateNDecimalPlaces(num1, 3)
```

```
num1 = calculateNDecimalPlaces(num1, -1).
```

```
num1 = calculateNDecimalPlaces(10.125478 , 0).
```

Στο τέλος ζητείται από του μαθητές να γράψουν την παρακάτω συνάρτηση ή το ισοδύναμό της.

```
Function embadonOrt(ByVal basi As Double, ByVal ypsos As Double) As Double
    Dim embadon As Double
    embadon = basi * ypsos
    Return embadon
End Function
```

Καλό θα είναι να δημιουργήσουν οι μαθητές μια εφαρμογή κονσόλας και να χρησιμοποιήσουν αυτή την συνάρτηση.

Δραστηριότητα 5: Κλήση υπορουτίνας από υπορουτίνα.

Σε αυτή την δραστηριότητα ζητείται από τους μαθητές να δημιουργήσουν μια συνάρτηση που να υπολογίζει το εμβαδόν του κύκλου, όπως η παρακάτω:

```
Function embadonKyklou(ByVal aktina As Double) As Double
    Dim embadon As Double
    embadon = piValue() * (aktina ^2)
    Return embadon
End Function
```

Στην συνάρτηση αυτή οι μαθητές χρησιμοποιούν τη συνάρτηση *piValue* που δημιούργησαν στη δραστηριότητα 3.

Συστήνεται να φτιάξουν μια εφαρμογή κονσόλας που να χρησιμοποιεί την συνάρτηση αυτή. Παράδειγμα εφαρμογής δίνεται παρακάτω:

```
Module Module1
    Function piValue() As Double
        Return 3.14159
    End Function
    Function embadonKyklou(ByVal aktina As Double) As Double
        Dim embadon As Double
        embadon = piValue() * (aktina ^ 2)
```

```
Return embadon  
End Function
```

```
Sub Main()  
    Dim emb As Double  
    emb = embadonKyklou(2)  
    Console.WriteLine(emb)  
    Console.ReadLine()  
End Sub
```

```
End Module
```

Σενάριο Εκπαιδευτικών Δραστηριοτήτων

5. Οι συναρτήσεις και υπορουτίνες στη Visual Basic

Σύντομη περιγραφή

Στο συγκεκριμένο σενάριο οι μαθητές χωρίζονται σε ομάδες των 2 ατόμων. Ακολουθώντας τις οδηγίες του φύλλου εργασίας θα πρέπει σε κάθε δραστηριότητα να απαντήσουν στην ερώτηση που βρίσκεται στο τέλος. Ωστόσο, για να κατανοήσουν την ερώτηση θα πρέπει πρώτα να έχουν γράψει τον κώδικα που δίνεται στο φύλλο εργασίας σε μια εφαρμογή κονσόλας της Visual Basic. Ένα μέλος της ομάδας θα χρησιμοποιήσει το περιβάλλον για να πληκτρολογήσει τον κώδικα. Το δεύτερο μέλος παρακολουθεί, ελέγχει και συμβουλεύει. Προτείνεται τα δυο μέλη να εναλλάσσουν ρόλους σε κάθε στάδιο της δραστηριότητας.

Γνωστικό αντικείμενο και σύνδεση με το πρόγραμμα σπουδών

Τάξη:

Τομέας:

ΠΛΗΡΟΦΟΡΙΚΗΣ-

Γνωστικό Αντικείμενο:

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΑ ΕΡΓΑΛΕΙΑ

Διδακτική ενότητα:

3η

Διδακτικές ώρες:

3

Αριθμός μαθητών:

Στο συγκεκριμένο σενάριο οι μαθητές χωρίζονται σε ομάδες των 2 ατόμων. Η επιλογή των ατόμων της κάθε ομάδας θα γίνεται σύμφωνα με τη κρίση του καθηγητή με βασικό κριτήριο την καλύτερη συνεργασία των μαθητών μεταξύ τους.

Διδακτικοί Στόχοι για το Σενάριο

Μετά το πέρας της εκπαιδευτικής δραστηριότητας οι μαθητές θα:

- γνωρίζουν τους τύπους των μεταβλητών και των σταθερών στη Visual Basic
- μπορούν να μεταχειριστούν το αντικείμενο Console της Visual Basic για να δημιουργούν input/output σε εφαρμογές κονσόλας

- μπορούν να χρησιμοποιούν τις βασικές συναρτήσεις του αντικείμενου Math
- μπορούν να χειρίζονται τις μεταβλητές τύπου String σαν αντικείμενα
- μπορούν να δηλώσουν μεταβλητές τύπου Date και να τους δώσουν τιμές

Διδακτική προσέγγιση

Για την επίτευξη των στόχων του συγκεκριμένου σεναρίου έχει επιλεγεί η ομαδοσυνεργατική διδακτική προσέγγιση. Οι μαθητές χωρίζονται σε μικρές ομάδες (2 ατόμων) και με τη βοήθεια του εκπαιδευτικού καλούνται να ολοκληρώσουν τις προτεινόμενες στο φύλλο εργασίας δραστηριότητες. Ο ρόλος του εκπαιδευτικού είναι υποστηρικτικός, καθοδηγώντας τους μαθητές στα σημεία που είναι απαραίτητο και βοηθώντας τους να ανακαλύψουν τη γνώση, συμμετέχοντας ενεργά στην διδακτική διαδικασία.

Εργαλεία ΤΠΕ που αξιοποιούνται στο πλαίσιο του σεναρίου

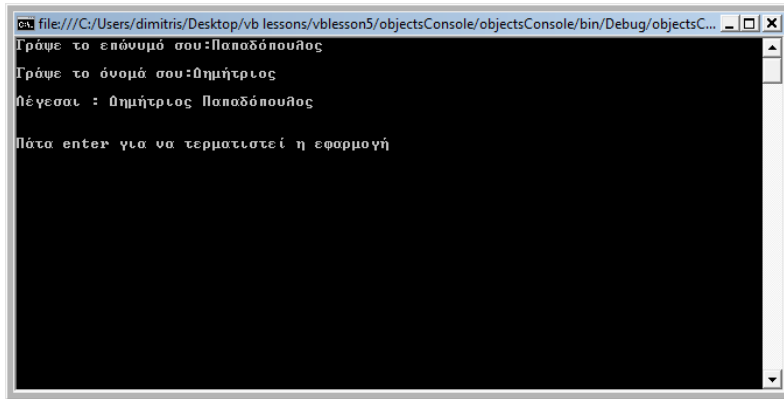
- Microsoft Visual basic 2008 Express Edition

Βήματα υλοποίησης του σεναρίου

Σε αυτό το φύλλο εργασίας γίνεται μία εισαγωγή στον αντικειμενοστραφή προγραμματισμό. Ο εκπαιδευτικός καλό θα είναι να θυμίσει στους μαθητές τη χρησιμότητα των αντικειμένων στον προγραμματισμό. Επίσης, είναι σημαντικό να διευκρινίσει τη διαφορά μεταξύ κλάσης και στιγμιότυπου (instance).

Δραστηριότητα 1- Το αντικείμενο console

Δίνεται στους μαθητές έτοιμος ο κώδικας για μια εφαρμογή κονσόλας που ζητάει όνομα και επώνυμο από το χρήστη και τυπώνει μια φράση που τα χρησιμοποιεί. Ο κώδικας αυτός θα χρησιμεύσει σαν παράδειγμα για τον κώδικα που ζητείται από τους μαθητές να γράψουν.



Ο κώδικας που πρέπει να γράψουν οι μαθητές έχει τις παρακάτω προδιαγραφές:

Στόχος της εφαρμογής: ο υπολογισμός του εμβαδού ενός τριγώνου.

Περιγραφή.

Εμφανίζεται το μήνυμα «Δώσε τιμή για βάση».

Ο χρήστης πληκτρολογεί μια τιμή

Εμφανίζεται το μήνυμα «Δώσε τιμή για ύψος»

Ο χρήστης πληκτρολογεί μια τιμή

Εμφανίζεται το μήνυμα «Το εμβαδόν του τριγώνου είναι: » και η τιμή του εμβαδού (που υπολογίζεται από τον τύπο $E = \beta * \upsilon / 2$)

Ο κώδικας που πρέπει να δημιουργήσουν οι μαθητές δίνεται παρακάτω:

Module Module1

Sub Main()

Console.WriteLine(Math.Floor(6.8))

Console.Write("Δώσε τιμή για βάση:")

Dim basi **As Integer** = Console.ReadLine()

Console.WriteLine()

Console.Write("Δώσε τιμή για ύψος:")

Dim ypsos **As Integer** = Console.ReadLine()

Console.WriteLine()

Console.Write("Το εμβαδόν του τριγώνου είναι: ")

Console.Write(basi * ypsos / 2)

Console.WriteLine()

Console.WriteLine()

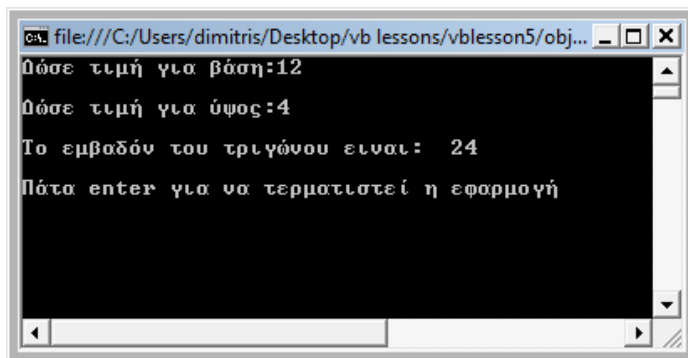
Console.Write("Πάτα enter για να τερματιστεί η εφαρμογή")

```
Console.ReadLine()
```

```
End Sub
```

```
End Module
```

Παρακάτω δίνεται το output της εφαρμογής



Δραστηριότητα 2: Το αντικείμενο Math

Στη δραστηριότητα αυτή οι μαθητές μαθαίνουν να χρησιμοποιούν το αντικείμενο Math. Τους ζητείται να δημιουργήσουν ένα απλό πρόγραμμα που υπολογίζει εμβαδόν και περίμετρο του κύκλου με βάση την ακτίνα που δίνει ο χρήστης. Οι μαθητές θα πρέπει να χρησιμοποιήσουν την σταθερά Math.PI για να υλοποιήσουν το πρόγραμμα. Παρακάτω παρατίθεται ο κώδικας

```
Module Module1
```

```
Sub Main()
```

```
Console.Write("Δώσε τιμή για ακτίνα:")
```

```
Dim r As Integer = Console.ReadLine()
```

```
Console.WriteLine()
```

```
Console.Write("Το εμβαδόν του κύκλου είναι: ")
```

```

Console.Write((r ^ 2) * Math.PI)
Console.WriteLine()
Console.Write("Η περίμετρος του κύκλου είναι: ")
Console.Write(2 * Math.PI * r)
Console.WriteLine()
Console.WriteLine()
Console.Write("Πάτα enter για να τερματιστεί η εφαρμογή")
Console.ReadLine()

```

End Sub

End Module

Συζητήστε στην τάξη το πλεονέκτημα της χρήσης της σταθεράς Math.PI που δίνει η Visual Basic έναντι της χρήσης μιας σταθεράς που θα πληκτρολογήσει ο προγραμματιστής (π.χ 3.14)

Δραστηριότητα 3: Το String σαν αντικείμενο

Σε αυτή τη δραστηριότητα οι μαθητές θα πρέπει να εμβαθύνουν στη χρήση των αντικειμένων. Γίνεται για πρώτη φορά η διαφοροποίηση μεταξύ κλάσης και απεικόνισης (instance). Σαν παράδειγμα χρησιμοποιείται η κλάση String. Δίνονται τμήματα κώδικα έτσι ώστε οι μαθητές να εξοικειωθούν με την κλάση String. Ο εκπαιδευτικός μπορεί να ζητήσει από τους μαθητές να δημιουργήσουν μια εφαρμογή κονσόλας και να δοκιμάσουν τις ιδιότητες και μεθόδους της κλάσης String που παρουσιάζονται.

Στο τέλος ζητείται από του μαθητές να δημιουργήσουν μια εφαρμογή με τις παρακάτω προδιαγραφές.

Στόχος της εφαρμογής: βρίσκει και τυπώνει την πρώτη λέξη και το μήκος της από μια φράση που πληκτρολογεί ο χρήστης.

Περιγραφή.

Εμφανίζεται το μήνυμα «Πληκτρολόγησε μια φράση ».

Ο χρήστης πληκτρολογεί μια φράση

Εμφανίζεται το μήνυμα «Η πρώτη λέξη της φράσης σου είναι: » και η πρώτη λέξη

Εμφανίζεται το μήνυμα «Η λέξη έχει x χαρακτήρα/ες» όπου x ο αριθμός χαρακτήρων της λέξης.

Οι μαθητές θα πρέπει να αναγνωρίσουν την πρώτη λέξη της φράσης από το πρώτο κενό που θα συναντήσουν. Συζητήστε στην τάξη τα προβλήματα που μπορούν να προκύψουν (π.χ στη φράση «επομένως, το συμπέρασμα που βγαίνει είναι ...»)

Ο ενδεικτικός κώδικας για την υλοποίηση της εφαρμογής παρατίθεται πιο κάτω:

Module Module1

Sub Main()

```

Console.Write("Πληκτρολόγησε μια φράση ")
Dim frasi As String = Console.ReadLine()
Dim i As Integer = frasi.IndexOf(" ")
Dim lexi As String = frasi.Substring(0, i)
Console.Write("Η πρώτη λέξη της φράσης σου είναι:")
Console.WriteLine(lexi)
Console.Write("Η λέξη έχει ")
Console.Write(lexi.Length)
Console.WriteLine(" χαρακτήρα/ες")
Console.ReadLine()

```

End Sub

End Module

Δραστηριότητα 4: Η κλάση Date

Σε αυτή τη δραστηριότητα οι μαθητές έρχονται σε επαφή με τις κλάσεις Date και TimeSpan. Δίνονται παραδείγματα κώδικα για τη χρήση των μεταβλητών τύπου Date.

Ακολούθως εξηγείται η πράξη αφαίρεσης Date από Date που το αποτέλεσμά της είναι τύπου TimeSpan.

Οι μαθητές συνδυάζοντας τις γνώσεις που έχουν αποκτήσει μέχρι τώρα μπορούν να υλοποιήσουν την εφαρμογή που τους ζητείται.

Στόχος της εφαρμογής: βρίσκει την ηλικία του χρήστη αφού ο χρήστης έχει πληκτρολογήσει την ημερομηνία γέννησης

Περιγραφή.

Εμφανίζεται το μήνυμα «Πληκτρολόγησε την ημερομηνία γέννησης με τη μορφή μέρα/μήνας/έτος ».

Ο χρήστης πληκτρολογεί μια ημερομηνία

Εμφανίζεται το μήνυμα «Με βάση τα στοιχεία που έδωσες σήμερα είσαι x ετών» όπου x η ηλικία του χρήστη σε έτη.

Ενδεικτικός κώδικας για τη υλοποίηση της εφαρμογής παρατίθεται παρακάτω.

Module Module1

Sub Main()

```
Console.WriteLine("Πληκτρολόγησε την ημερομηνία γεννησης με τη μορφή  
μερα/μηνας/ετος ")
```

```
Dim dateOfBirthStr As String = Console.ReadLine()
```

```
Dim dateOfBirth As Date = Date.Parse(dateOfBirthStr)
```

```
Dim dateOfToday As Date = Date.Now
```

```
Dim dtDiff As TimeSpan = dateOfToday - dateOfBirth
```

```
Dim userYears As Integer = Math.Round(dtDiff.Days / 365)
```

```
Console.WriteLine("Με βάση τα στοιχεία που έδωσες σήμερα είσαι ")
```

```
Console.WriteLine(userYears)
```

```
Console.WriteLine(" ετών")
```

```
Console.ReadLine()
```

End Sub

End Module

Σενάριο Εκπαιδευτικών Δραστηριοτήτων

6. Οι συναρτήσεις και υπορουτίνες στη Visual Basic

Σύντομη περιγραφή

Στο συγκεκριμένο σενάριο οι μαθητές χωρίζονται σε ομάδες των 2 ατόμων. Σε κάθε δραστηριότητα δίνεται ένα παράδειγμα κώδικα και μετά ζητείται από τους μαθητές να δημιουργήσουν μια ανάλογη εφαρμογή. Σε μερικές από τις δραστηριότητες δίνεται στους μαθητές ψευδοκώδικας που θα πρέπει να υλοποιήσουν σε Visual Basic. Ένα μέλος της ομάδας θα πρέπει να χειρίζεται τον υπολογιστή πληκτρολογώντας τον κώδικα, αφού έχει συζητήσει μαζί με το άλλο μέλος σχετικά με την υλοποίησή του. Το δεύτερο μέλος της ομάδας θα λάβει το ρόλο του χρήστη και θα ρεστάρει τις εφαρμογές που θα δημιουργήσουν σε αυτές τις δραστηριότητες. Προτείνεται τα δυο μέλη να εναλλάσσουν ρόλους σε κάθε δραστηριότητα.

Γνωστικό αντικείμενο και σύνδεση με το πρόγραμμα σπουδών

Τάξη:

Τομέας:

ΠΛΗΡΟΦΟΡΙΚΗΣ-

Γνωστικό Αντικείμενο:

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΑ ΕΡΓΑΛΕΙΑ

Διδακτική ενότητα:

6η

Διδακτικές ώρες:

3

Αριθμός μαθητών:

Στο συγκεκριμένο σενάριο οι μαθητές χωρίζονται σε ομάδες των 2 ατόμων. Η επιλογή των ατόμων της κάθε ομάδας θα γίνεται σύμφωνα με τη κρίση του καθηγητή με βασικό κριτήριο την καλύτερη συνεργασία των μαθητών μεταξύ τους.

Διδακτικοί Στόχοι για το Σενάριο

Μετά το πέρας των εκπαιδευτικών δραστηριοτήτων οι μαθητές θα:

- γνωρίζουν τις εντολές διακλάδωσης *If .. then ... else*

- γνωρίζουν τις λογικές πράξεις AND και OR και πώς αυτές συνδυάζονται με τις εντολές διακλάδωσης
- γνωρίζουν τις πιο σύνθετες εντολές διακλάδωσης *If .. then ... elseIf* και *select .. case*

Διδακτική προσέγγιση

Για την επίτευξη των στόχων του συγκεκριμένου σεναρίου έχει επιλεγεί η ομαδοσυνεργατική-ανακαλυπτική διδακτική προσέγγιση. Οι μαθητές χωρίζονται σε μικρές ομάδες (2 ατόμων) και με τη βοήθεια του εκπαιδευτικού καλούνται να ολοκληρώσουν τις προτεινόμενες στο φύλλο εργασίας δραστηριότητες. Ο ρόλος του εκπαιδευτικού είναι υποστηρικτικός, καθοδηγώντας τους μαθητές στα σημεία που είναι απαραίτητο και βοηθώντας τους να ανακαλύψουν τη γνώση, συμμετέχοντας ενεργά στην διδακτική διαδικασία.

Εργαλεία ΤΠΕ που αξιοποιούνται στο πλαίσιο του σεναρίου

- Microsoft Visual basic 2008 Express Edition

Βήματα υλοποίησης του σεναρίου

Σε αυτό το φύλλο εργασίας οι μαθητές εξασκούνται στις εντολές διακλάδωσης *if .. then, if .. then .. elseIf, select .. case* και στις λογικές πράξεις. Σε κάθε δραστηριότητα δίνεται ένα παράδειγμα κώδικα που καλό θα είναι οι μαθητές να το υλοποιήσουν έτσι ώστε να το κατανοήσουν καλύτερα. Στη συνέχεια, τους ζητείται να φτιάξουν ένα αντίστοιχο. Ο εκπαιδευτικός θα πρέπει να εξηγήσει στους μαθητές τη χρησιμότητα αυτών των εντολών με τη χρήση λογικού διαγράμματος ή ψευδοκώδικα.

Δραστηριότητα 1 : Η εντολή *If .. then ... else*

Στην δραστηριότητα αυτή παρουσιάζονται στους μαθητές παραδείγματα χρήσης των εντολών *If .. then .. else*. Πρώτα τους παρουσιάζεται μια συνάρτηση που υπολογίζει την απόλυτη τιμή ενός αριθμού, χρησιμοποιώντας μόνο *if .. then*. Μετά του

παρουσιάζεται μια εφαρμογή κονσόλας που ανάλογα με την ώρα της ημέρας τυπώνει «καλημέρα» ή «καλησπέρα», χρησιμοποιώντας *if.. then .. else*.

Στη συνέχεια ζητείται από τους μαθητές να δημιουργήσουν μια εφαρμογή κονσόλας με τις παρακάτω προδιαγραφές:

Στόχος της εφαρμογής: Η εφαρμογή τυπώνει αν κυκλοφορούν σήμερα τα μονά ή τα ζυγά αυτοκίνητα (δεν λαμβάνει υπ όψιν της το σαββατοκύριακο)

Περιγραφή:

Η εφαρμογή παίρνει τη σημερινή ημερομηνία

Εξετάζει την ημέρα του μήνα (από την ιδιότητα *Day*) και αν αυτή διαιρείται ακριβώς με το 2 εμφανίζει το μήνυμα «Σήμερα κυκλοφορούν τα ζυγά» αλλιώς εμφανίζει το μήνυμα «Σήμερα κυκλοφορούν τα μονά.»

Χρησιμοποιείτε τον τελεστή *mod* για να υπολογίσετε το υπόλοιπο της διαίρεσης της ημερομηνίας με το 2

Για την υλοποίησή της οι μαθητές θα πρέπει να χρησιμοποιήσουν δεξιότητες που έχουν αποκτήσει στα προηγούμενα φύλλα εργασίας (π.χ το χειρισμό του αντικειμένου *console*).

Παρακάτω παρατίθεται μια ενδεικτική υλοποίηση.

```
Module Module1

    Sub Main()
        Dim dt As Date = Date.Now
        Dim day As Integer = dt.Day()
        Dim ypol As Integer = day Mod 2
        If ypol = 1 Then
            Console.WriteLine("Σήμερα κυκλοφορούν τα μονά")
        Else
            Console.WriteLine("Σήμερα κυκλοφορούν τα ζυγά")
        End If
        Console.ReadLine()
    End Sub

End Module
```

Δραστηριότητα 2 : Nested if

Στη δραστηριότητα αυτή οι μαθητές θα τοποθετήσουν τις εντολές *if .. then* μέσα σε εντολές *if .. then* για να κατανοήσουν την έννοια του nested if.

Ζητείται από τους μαθητές να επεκτείνουν την εφαρμογή που έγραψαν στην προηγούμενη δραστηριότητα λαμβάνοντας υπ όψιν και τον παράγοντα σαββατοκύριακο. Η υλοποίηση της εφαρμογής μπορεί να γίνει με πολλούς τρόπους. Γι αυτό δίνεται στους μαθητές ψευδικώδικας που πρέπει να ακολουθήσουν.

Η υλοποίηση της εφαρμογής με βάση τον ψευδοκώδικα βρίσκεται παρακάτω:

```
Module Module1

    Sub Main()
        Console.WriteLine("δωσε τον λήγοντα από την πινακίδα του αυτοκινήτου σου")
        Dim pinakida As Integer = Console.ReadLine()
        Dim dt As Date = Date.Now
        Dim mera As Integer = dt.DayOfWeek
        If mera <> 7 Then
            If mera <> 6 Then
                Dim pinakidaYpoloipo As Integer = pinakida Mod 2
                Dim meraYpoloipo As Integer = dt.Day Mod 2
                If pinakidaYpoloipo <> meraYpoloipo Then
                    Console.WriteLine("Σήμερα αν κατέβεις στο κέντρο, θα σε γράψουν")
                End If
            End If
        End If

        Console.ReadLine()
    End Sub

End Module
```

Δραστηριότητα 3 : Οι εντολές AND , OR, NOT

Στη δραστηριότητα αυτή, παρουσιάζεται στους μαθητές η χρήση των λογικών τελεστών. Σαν παράδειγμα τους δίνεται η υλοποίηση της εφαρμογής που ζητήθηκε στην προηγούμενη δραστηριότητα με τη χρήση του τελεστή and. Επίσης, τους παρουσιάζεται η μέθοδος της αποθήκευσης των αποτελεσμάτων λογικών πράξεων σε λογικές μεταβλητές. Για την κατανόηση του τελεστή or παρουσιάζεται μια νέα εφαρμογή που ελέγχει την ημέρα της εβδομάδας και τυπώνει ένα μήνυμα ανάλογα αν είναι σαββατοκύριακο ή αν είναι καθημερινή.

Τέλος, ζητείται από τους μαθητές να υλοποιήσουν μια εφαρμογή που να υπολογίζει αν ένα έτος είναι δίσεκτο ή όχι. Για την υλοποίηση της θα πρέπει να χρησιμοποιήσουν τους τελεστές *and* και *or*.

Παρακάτω δίνεται η υλοποίηση της εφαρμογή αυτής με τη χρήση λογικών τελεστών.

```
Module Module1

    Sub Main()
        Console.WriteLine("Δώσε έτος:")
        Dim theYear As Integer = Console.ReadLine()
        Dim div4 As Boolean = (theYear Mod 4) = 0
        Dim notDiv100 As Boolean = (theYear Mod 100) <> 0
        Dim div400 As Boolean = (theYear Mod 400) = 0
        If (div4 And notDiv100) Or div400 Then
            Console.WriteLine("Δισσεκτο")
        Else
            Console.WriteLine("δεν είναι Δισσεκτο")
        End If
        Console.ReadLine()

    End Sub

End Module
```

Για να κατανοήσουν οι μαθητές τη χρήση του *NOT* ο καθηγητής θα πρέπει να τους ζητήσει να αντικαταστήσουν την εντολή

```
Dim notDiv100 As Boolean = (theYear Mod 100) <> 0
με τις
    Dim div100 As Boolean = (theYear Mod 100) = 0
    Dim notDiv100 As Boolean = Not div100
```

Δραστηριότητα 4 : Το σύνθετο σχήμα If...Then...Else If

Στόχος της δραστηριότητας αυτής είναι οι μαθητές να κατανοήσουν τη δομή *if.. then .. elseif*. Πρώτα εξηγείται στους μαθητές πως δομείται ένα πρόγραμμα με την εν λόγω δομή. Μετά τους ζητείται να υλοποιήσουν ξανά την εφαρμογή που έφτιαξαν στην προηγούμενη δραστηριότητα με τη χρήση αυτής της δομής. Για να βοηθηθούν οι μαθητές δίνεται ψευδοκώδικας που θα πρέπει να ακολουθήσουν για την υλοποίηση.

Παρακάτω παρατίθεται η υλοποίηση της εφαρμογής.

```
Module Module1

    Sub Main()
        Console.WriteLine("Δώσε έτος:")
        Dim theYear As Integer = Console.ReadLine()
        Dim div4 As Boolean = (theYear Mod 4) = 0
        Dim div100 As Boolean = (theYear Mod 100) = 0
        Dim div400 As Boolean = (theYear Mod 400) = 0

        Dim isLeapYear As Boolean
        If div400 Then
            isLeapYear = True
```

```

ElseIf div100 Then
    isLeapYear = False
ElseIf div4 Then
    isLeapYear = True
Else
    isLeapYear = False
End If
If isLeapYear Then
    Console.WriteLine("Δισσεκτο")
Else
    Console.WriteLine("δεν είναι Δισσεκτο")
End If
Console.ReadLine()
End Sub

```

End Module

Οι μαθητές θα πρέπει να αποθηκεύσουν την εφαρμογή για να τη χρησιμοποιήσουν στο επόμενο φύλλο εργασίας.

Δραστηριότητα 5 : Δομή Select Case

Στους μαθητές παρουσιάζεται η δομή *select .. case*. Τους δίνεται μια εφαρμογή που εξετάζει την ημερομηνία και τυπώνει την αντίστοιχη μέρα της εβδομάδας. Οι μαθητές καλό θα είναι να γράψουν την εφαρμογή αυτή σε Visual Basic και να τη δοκιμάσουν. Ακολούθως τους ζητείται να την ξαναγράψουν χρησιμοποιώντας τη δομή *if .. then .. elseif*

Παρακάτω δίνεται η υλοποίηση της εφαρμογής.

```

Module Module1

    Sub Main()
        Dim dt As Date = Date.Now
        Dim dayOfWeek As Integer = dt.DayOfWeek
        If dayOfWeek = 1 Then
            Console.WriteLine("Σήμερα είναι Δευτέρα")
        ElseIf dayOfWeek = 2 Then
            Console.WriteLine("Σήμερα είναι Τρίτη")
        ElseIf dayOfWeek = 3 Then
            Console.WriteLine("Σήμερα είναι Τετάρτη")
        ElseIf dayOfWeek = 4 Then
            Console.WriteLine("Σήμερα είναι Πέμπτη")
        ElseIf dayOfWeek = 5 Then
            Console.WriteLine("Σήμερα είναι Παρασκευή")
        ElseIf dayOfWeek = 5 Or dayOfWeek = 6 Then
            Console.WriteLine("Σαββατοκύριακο, μπορείς να
ξεκουραστείς")
        End If

        Console.ReadLine()
    End Sub

```

End Module

Τέλος συζητήστε στην τάξη του διαφορετικούς τρόπους που είδατε για την υλοποίηση μια εφαρμογής. Ποιον τρόπο πρέπει να επιλέγουμε και με ποιο κριτήριο; (ταχύτητα υλοποίησης, αναγνωσιμότητα, ταχύτητα εκτέλεσης)

Σενάριο Εκπαιδευτικών Δραστηριοτήτων

7. Οι συναρτήσεις και υπορουτίνες στη Visual Basic

Σύντομη περιγραφή

Στο συγκεκριμένο σενάριο οι μαθητές χωρίζονται σε ομάδες των 2 ατόμων. Σε κάθε δραστηριότητα δίνεται ένα παράδειγμα κώδικα και μετά ζητείται από τους μαθητές να δημιουργήσουν μια ανάλογη εφαρμογή. Σε μερικές από τις δραστηριότητες δίνεται στους μαθητές ψευδοκώδικας που θα πρέπει να υλοποιήσουν σε Visual Basic. Ένα μέλος της ομάδας θα πρέπει να χειρίζεται τον υπολογιστή πληκτρολογώντας τον κώδικα, αφού έχει συζητήσει μαζί με το άλλο μέλος σχετικά με την υλοποίησή του. Το δεύτερο μέλος της ομάδας θα λάβει το ρόλο του χρήστη και θα τεστάρει τις εφαρμογές που θα δημιουργήσουν σε αυτές τις δραστηριότητες. Προτείνεται τα δυο μέλη να εναλλάσσουν ρόλους σε κάθε δραστηριότητα.

Γνωστικό αντικείμενο και σύνδεση με το πρόγραμμα σπουδών

Τάξη:

Τομέας:

ΠΛΗΡΟΦΟΡΙΚΗΣ-

Γνωστικό Αντικείμενο:

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΑ ΕΡΓΑΛΕΙΑ

Διδακτική ενότητα:

7η

Διδακτικές ώρες:

3

Αριθμός μαθητών:

Στο συγκεκριμένο σενάριο οι μαθητές χωρίζονται σε ομάδες των 2 ατόμων. Η επιλογή των ατόμων της κάθε ομάδας θα γίνεται σύμφωνα με τη κρίση του καθηγητή με βασικό κριτήριο την καλύτερη συνεργασία των μαθητών μεταξύ τους.

Διδακτικοί Στόχοι για το Σενάριο

Μετά το πέρας της εκπαιδευτικής δραστηριότητας οι μαθητές θα:

- Θα γνωρίζουν να χρησιμοποιούν τις δομές επανάληψης while, repeat, for.
- Θα γνωρίζουν τη διαφοροποίηση ανάμεσα στις δομές repeat και while

- Θα μπορούν να δημιουργούν πίνακες στη Visual Basic
- Θα μπορούν να προσπελάσουν όλα τα στοιχεία ενός πίνακα με τη χρήση της δομής for.

Διδακτική προσέγγιση

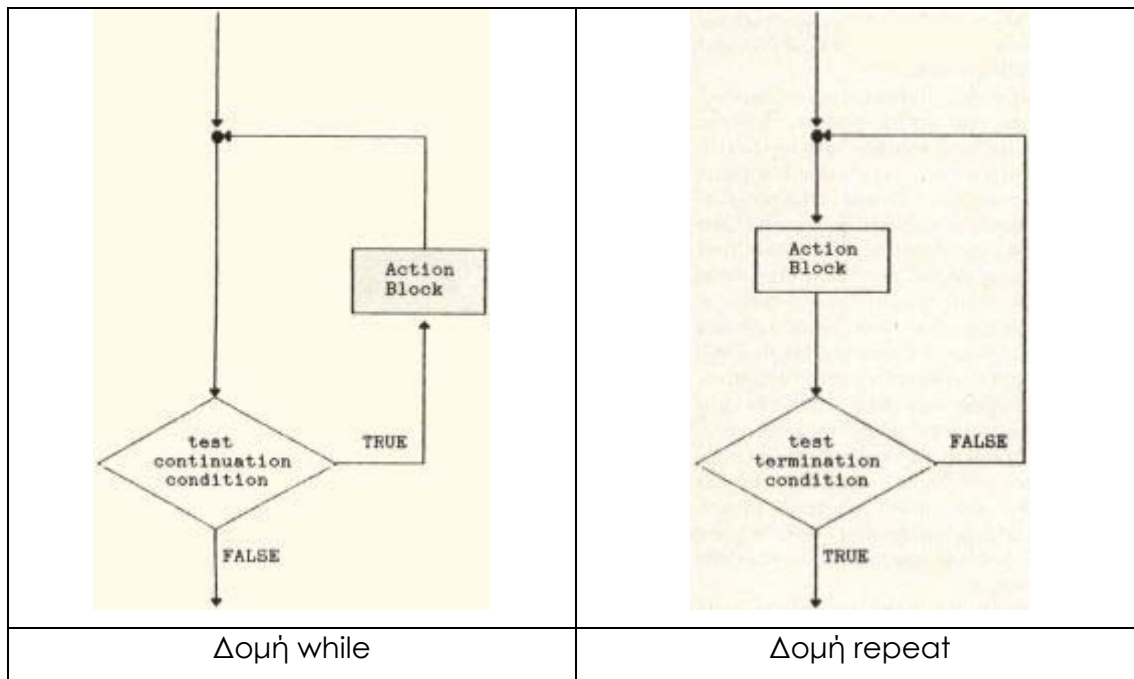
Για την επίτευξη των στόχων του συγκεκριμένου σεναρίου έχει επιλεγεί η ομαδοσυνεργατική-ανακαλυπτική διδακτική προσέγγιση. Οι μαθητές χωρίζονται σε μικρές ομάδες (2 ατόμων) και με τη βοήθεια του εκπαιδευτικού καλούνται να ολοκληρώσουν τις προτεινόμενες στο φύλλο εργασίας δραστηριότητες. Ο ρόλος του εκπαιδευτικού είναι υποστηρικτικός, καθοδηγώντας τους μαθητές στα σημεία που είναι απαραίτητο και βοηθώντας τους να ανακαλύψουν τη γνώση, συμμετέχοντας ενεργά στην διδακτική διαδικασία.

Εργαλεία ΤΠΕ που αξιοποιούνται στο πλαίσιο του σεναρίου

- Microsoft Visual basic 2008 Express Edition

Βήματα υλοποίησης του σεναρίου

Σε αυτό το φύλλο εργασίας οι μαθητές εξασκούνται στις δομές επανάληψης. Δίνονται παραδείγματα κώδικα που χρησιμοποιεί δομές επανάληψης και μετά ζητείται από τους μαθητές με βάση τα παραδείγματα του φύλλου εργασίας να συγγράψουν δικό τους κώδικα. Θα ήταν χρήσιμο οι μαθητές να δημιουργήσουν σε περιβάλλον Visual Basic τις εφαρμογές που δίνονται σαν παράδειγμα. Ο καθηγητής θα πρέπει να εξηγήσει στους μαθητές τη λειτουργία των δομών επανάληψης με τη χρήση λογικού διαγράμματος και ειδικότερα τη διαφοροποίηση της δομής «repeat» από τη δομή «while».



Η σύνταξη των δομών αυτών γίνεται με πολλούς τρόπους στη Visual Basic. Στο φύλλο εργασίας δίνεται μόνο ένας τρόπος σύνταξης για κάθε δομή. Ανάλογα το επίπεδο της τάξης ο καθηγητής μπορεί να εξηγήσει στους μαθητές και τους υπόλοιπους τρόπους σύνταξης που βελτιώνουν την αναγνωσιμότητα του κώδικα. Π.χ. *Do while not a < 0* μπορεί να συνταθεί ως *Do Until a < 0*.

Δραστηριότητα 1 : Η δομή repeat

Στη δραστηριότητα αυτή στους μαθητές δίνονται παραδείγματα επαναληπτικών διαδικασιών με τη χρήση της δομής «repeat». Τέλος ζητείται από τους μαθητές να δημιουργήσουν μια απλή εφαρμογή κονσόλας με την παρακάτω περιγραφή:

Στόχος της εφαρμογής: ζητάει από το χρήστη να πληκτρολογήσει μια λέξη μέχρι ο χρήστης να πληκτρολογήσει τη λέξη «bye», με αυτή τη λέξη τερματίζεται η εφαρμογή.

Καλό θα είναι ο καθηγητής να ζητήσει από τους μαθητές να υλοποιήσουν την εφαρμογή σε λογικό διάγραμμα πριν προχωρήσουν στην υλοποίησή της σε κώδικα.

Παρακάτω δίνεται ο κώδικας της εφαρμογής

```
Module Module1
```



```

Dim str As String
Sub Main()
    Do
        Console.WriteLine("Δώσε τη μαγική λέξη:")
        str = Console.ReadLine()
    Loop Until str = "bye"
End Sub

End Module

```

Συζητήστε στην τάξη γιατί η δομή «repeat» είναι η πιο κατάλληλη για την υλοποίηση αυτή της εφαρμογής.

Δραστηριότητα 2 : Η δομή while

Στη δραστηριότητα 2 και 3 δίνεται στους μαθητές ένας κώδικας που στην τελική του μορφή χρησιμοποιεί και τις τρεις δομές. Σε αυτή τη δραστηριότητα δίνεται ο κώδικας με τις δομές «repeat» και «while» ενώ η συνάρτηση που θα χρησιμοποιήσει την δομή «for» δίνεται ημιτελής. Η εφαρμογή υπολογίζει το παραγοντικό ενός αριθμού που πληκτρολογεί ο χρήστης. Οι μαθητές έρχονται για πρώτη φορά σε επαφή με την αξιολόγηση των δεδομένων εισόδου.

Οι μαθητές θα πρέπει να συγκρίνουν τη χρήση της δομής «repeat» που είδαν στην πρώτη δραστηριότητα με τη χρήση της δομής «while» που δίνεται σε αυτή της δραστηριότητα, στην κεντρική υπορουτίνα του προγράμματος. Επίσης θα πρέπει να δουν την χρήση της δομής «repeat» στην υπορουτίνα *getUserInput()*.

Τέλος ζητείται από τους μαθητές να τροποποιήσουν την υπορουτίνα *getUserInput()* έτσι ώστε να εμφανίζει ένα μήνυμα αν ο χρήστης δεν πληκτρολογήσει τα αναμενόμενα δεδομένα. Για την υλοποίηση της παραπάνω προδιαγραφή καταλληλότερη είναι η δομή «while». Παρακάτω παρατίθεται η υλοποίηση της υπορουτίνας.

```

Function getUserInput() As Integer
    Dim strTheNumber As String
    Dim theNumber As Integer
    Console.WriteLine("Δωσε έναν αριθμό")
    Console.WriteLine("0 για τερματισμό")
    strTheNumber = Console.ReadLine()
    Do While Not IsNumeric(strTheNumber)

```

```

        Console.WriteLine("Το '" + strTheNumber + "' δεν είναι
αριθμός")
        Console.WriteLine("Δωσε έναν αριθμό")
        Console.Write("0 για τερματισμό")
        strTheNumber = Console.ReadLine()
    Loop
    theNumber = strTheNumber
    Return theNumber
End Function

```

Δραστηριότητα 3 : Η δομή for .. next

Στη δραστηριότητα αυτή εξηγείται στους μαθητές η χρήση της δομής «for». Δίνονται παραδείγματα χρήσης της δομής. Τέλος ζητείται από τους μαθητές να συμπληρώσουν την υπορουτίνα που υπολογίζει το παραγοντικό (*calculateParagontiko*).

Ο κώδικας της υπορουτίνας παρατίθεται παρακάτω.

```

Function calculateParagontiko(ByVal number As Integer) As String
    Dim parag As Long = 1
    For i = 1 To number
        parag = parag * i
    Next
    Return parag.ToString()
End Function

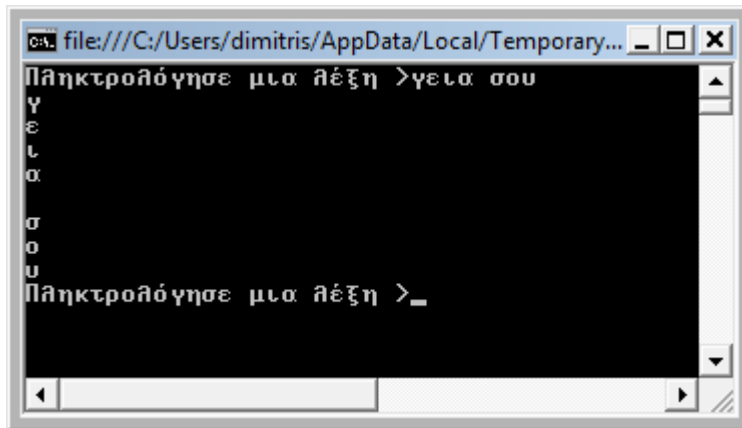
```

Στην υπορουτίνα αυτή είναι σημαντικός ο τύπος μεταβλητής που επιλέγεται για να αποθηκεύσει το παραγοντικό. Ωστόσο ο αριθμός που προκύπτει είναι υπερβολικά μεγάλος για να χειριστεί από μεταβλητές ακόμα και τύπου long. Συζητήστε το πρόβλημα αυτό στην τάξη.

Δραστηριότητα 4 : Οι πίνακες στη Visual Basic

Σε αυτή τη δραστηριότητα οι μαθητές έρχονται σε επαφή με την έννοια του πίνακα.

Αρχικά τους παρουσιάζεται η μεταβλητή τύπου string σαν πίνακας από char. Τους δίνεται σαν παράδειγμα μια εφαρμογή που παίρνει ένα string και τυπώνει τους χαρακτήρες στην οθόνη έναν ανά γραμμή.



Η εφαρμογή χρησιμοποιεί τη δομή «while» για να συνεχίσει τη λειτουργία της μέχρι ο χρήστης να πατήσει απλώς το <Enter>. Μπορείτε αν θέλετε να ζητήσετε από τους μαθητές να αντικαταστήσουν την εντολή *Do While str <> ""* με την *Do Until str = ""*.

Μετά τους δίνεται σαν παράδειγμα ένας πίνακας τύπου string όπου αποθηκεύονται οι μέρες της εβδομάδος. Με τη χρήση αυτού του πίνακα δίνεται στους μαθητές μια υλοποίηση της εφαρμογής που είδαν στο «φύλλο εργασίας 6», που τυπώνει την τρέχουσα μέρα της εβδομάδας.

Τέλος ζητείται από τους μαθητές να δημιουργήσουν μια εφαρμογή που θα ζητάει από το χρήστη 3 αριθμούς. Οι αριθμοί θα αποθηκεύονται σε έναν πίνακα και στο τέλος θα εμφανίζει το άθροισμά τους. Μπορείτε να ζητήσετε από τους μαθητές να ενσωματώσουν έλεγχο για τα δεδομένα που πληκτρολογεί ο χρήστης.

Παρακάτω παρατίθεται ο κώδικας της εφαρμογής:

Module Module1

```

Function getUserInput() As Integer
    Dim strTheNumber As String
    Dim theNumber As Integer
    Console.WriteLine("Δωσε έναν αριθμό")
    Console.Write("0 για τερματισμό")
    strTheNumber = Console.ReadLine()
    Do While Not IsNumeric(strTheNumber)
        Console.WriteLine("Το '" + strTheNumber + "' δεν είναι
αριθμός")
        Console.WriteLine("Δωσε έναν αριθμό")
        strTheNumber = Console.ReadLine()
    Loop
    theNumber = strTheNumber
    Return theNumber
End Function

```

```

Sub Main()
    Dim n As Double
    Dim numbers(2) As Double
    For i = 0 To 2
        n = getUserInput()
        numbers(i) = n
    Next

    Dim sum As Double = 0
    For i = 0 To 2
        sum = sum + numbers(i)
    Next
    Console.WriteLine("Το άθροισμα των αριθμών που έδωσες είναι: ")
    Console.WriteLine(sum)
    Console.ReadLine()
End Sub

End Module

```

Η εφαρμογή ξαναχρησιμοποιεί την υπορουτίνα *getUserInput* που χρησιμοποιήθηκε στην δραστηριότητα 2.

Σενάριο Εκπαιδευτικών Δραστηριοτήτων

8. Οι συναρτήσεις και υπορουτίνες στη Visual Basic

Σύντομη περιγραφή

Στο συγκεκριμένο σενάριο οι μαθητές χωρίζονται σε ομάδες των 2 ατόμων. Σε κάθε δραστηριότητα δίνεται ένα παράδειγμα κώδικα και μετά ζητείται από τους μαθητές να δημιουργήσουν μια ανάλογη εφαρμογή. Σε μερικές από τις δραστηριότητες δίνεται στους μαθητές ψευδοκώδικας που θα πρέπει να υλοποιήσουν σε Visual Basic. Ένα μέλος της ομάδας θα πρέπει να χειρίζεται τον υπολογιστή πληκτρολογώντας τον κώδικα, αφού έχει συζητήσει μαζί με το άλλο μέλος σχετικά με την υλοποίησή του. Το δεύτερο μέλος της ομάδας θα λάβει το ρόλο του χρήστη και θα τεστάρει τις εφαρμογές που θα δημιουργήσουν σε αυτές τις δραστηριότητες. Προτείνεται τα δυο μέλη να εναλλάσσουν ρόλους σε κάθε δραστηριότητα.

Γνωστικό αντικείμενο και σύνδεση με το πρόγραμμα σπουδών

Τάξη:

Τομέας:

ΠΛΗΡΟΦΟΡΙΚΗΣ-

Γνωστικό Αντικείμενο:

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΑ ΕΡΓΑΛΕΙΑ

Διδακτική ενότητα:

7η

Διδακτικές ώρες:

3

Αριθμός μαθητών:

Στο συγκεκριμένο σενάριο οι μαθητές χωρίζονται σε ομάδες των 2 ατόμων. Η επιλογή των ατόμων της κάθε ομάδας θα γίνεται σύμφωνα με τη κρίση του καθηγητή με βασικό κριτήριο την καλύτερη συνεργασία των μαθητών μεταξύ τους.

Διδακτικοί Στόχοι για το Σενάριο

Μετά το πέρας της εκπαιδευτικής δραστηριότητας οι μαθητές θα:

- Θα γνωρίζουν να χρησιμοποιούν τις δομές επανάληψης while, repeat, for.
- Θα γνωρίζουν τη διαφοροποίηση ανάμεσα στις δομές repeat και while

- Θα μπορούν να δημιουργούν πίνακες στη Visual Basic
- Θα μπορούν να προσπελάσουν όλα τα στοιχεία ενός πίνακα με τη χρήση της δομής for.

Διδακτική προσέγγιση

Για την επίτευξη των στόχων του συγκεκριμένου σεναρίου έχει επιλεγεί η ομαδοσυνεργατική-ανακαλυπτική διδακτική προσέγγιση. Οι μαθητές χωρίζονται σε μικρές ομάδες (2 ατόμων) και με τη βοήθεια του εκπαιδευτικού καλούνται να ολοκληρώσουν τις προτεινόμενες στο φύλλο εργασίας δραστηριότητες. Ο ρόλος του εκπαιδευτικού είναι υποστηρικτικός, καθοδηγώντας τους μαθητές στα σημεία που είναι απαραίτητο και βοηθώντας τους να ανακαλύψουν τη γνώση, συμμετέχοντας ενεργά στην διδακτική διαδικασία.

Εργαλεία ΤΠΕ που αξιοποιούνται στο πλαίσιο του σεναρίου

- Microsoft Visual basic 2008 Express Edition

Βήματα υλοποίησης του σεναρίου

Σε αυτό το φύλλο εργασίας οι μαθητές εξασκούνται στην διαδικασία εκσφαλμάτωσης. Πρώτα τους εξηγείται η διαδικασία αντιμετώπισης συντακτικών σφαλμάτων. Στη συνέχεια η αντιμετώπιση run time errors με τη δομή try .. catch. Τέλος εξασκούνται στην αντιμετώπιση λογικών λαθών μέσα από παραδείγματα.

Δραστηριότητα 1 : Συντακτικά λάθη.

Στη δραστηριότητα αυτή εξηγείται στους μαθητές πως υποδεικνυει ο compiler τα συντακτικά λάθη.

Στην άσκηση τους ζητείται να εντοπίσουν τα συντακτικά λάθη στις παρακάτω γραμμές κώδικα.

```
Dim str As String
str = "";
Dim num As Integer = 0
Console.WriteLine ("Ο αριθμός "num" είναι 0")
```

Τα λάθη είναι :

`str = "";` Το σύμβολο “;” δεν μπορεί να τοποθετηθεί στο τέλος μια εντολής.

`Console.WriteLine ("Ο αριθμός "num" είναι 0")` η παράσταση στην παρένθεση δεν είναι συντακτικά σωστή. Η σωστή είναι `Console.WriteLine ("Ο αριθμός "#" είναι 0")`

Δραστηριότητα 2 : Που επηρεάζουν τα συντακτικά λάθη. (Option Explicit)

Στη δραστηριότητα αυτή παρουσιάζονται στους μαθητές οι επιλογές που μπορεί να κάνει ο προγραμματιστής και επηρεάζουν τον τρόπο που ο compiler αντιμετωπίζει τα συντακτικά λάθη. Στην συγκεκριμένη δραστηριότητα απρουσιάζεται η επιλογή Option Explicit.

Στην άσκηση ζητείται από τους μαθητές να εντοπίσουν τη default τιμή της επιλογής Option Explicit. Η τιμή αυτή είναι On.

Δραστηριότητα 3 : Επιλογές που επηρεάζουν τα συντακτικά λάθη. (Option Strict)

Στη δραστηριότητα αυτή παρουσιάζεται η επιλογή Option Strict. Η default τιμή της είναι off. Από τους μαθητές ζητείται να βάλουν την επιλογή αυτή on και να δουν την αλλαγή στην συμπεριφορά του compiler.

Στην άσκηση τους ζητείται να εντοπίσουν τα λάθη που θα εντοπίσει ο compiler στον παρακάτω κώδικα.

```
Option Strict On
Module Module1

    Sub Main()
        Console.WriteLine("Ο αριθμός 10 είναι άρτιος")
        Console.WriteLine("Ο αριθμός " + 11 + " είναι περιττός")
        Dim str As String = 0
        Dim myName As String
        myName = "Δημήτρης"
        Dim myAge As Integer = 40
        Console.WriteLine(myName + ", " + myAge)
    End Sub

End Module
```

Τα λάθη είναι στις γραμμές :

```
Console.WriteLine("Ο αριθμός " + 11 + " είναι περιττός")
Dim str As String = 0
```

```
Console.WriteLine(myName + ", " + myAge)
```

Δραστηριότητα 4 : Οι εντολές try ... catch

Στη δραστηριότητα αυτή οι μαθητές μαθαίνουν να χειρίζονται τα run time errors με τη δομή try .. catch. Δίνονται παραδείγματα χρήσης της δομής και τέλος ζητείται από τους μαθητές να εφαρμόσουν τη δομή αυτή στην εφαρμογή υπολογισμού του παραγοντικού που δημιούργησαν στο Φ.Ε 7. Ο κώδικας που θα πρέπει να δημιουργήσουν θα πρέπει να έχει την εξής μορφή:

```
Module Module1
    Function calculateParagontiko(ByVal number As Integer) As String
        Dim parag As ULong = 1
        Try
            For i = 1 To number
                parag = parag * i
            Next
            Catch ex As Exception
                Return "πολύ μεγάλο για να υπολογιστεί από αυτό το
λογισμικό"
            End Try
            Return parag.ToString()
        End Function

    Function getUserInput() As Integer
        Dim strTheNumber As String
        Dim theNumber As Integer
        Do
            Console.Write("Δωσε έναν αριθμό")
            Console.Write("0 για τερματισμό")
            strTheNumber = Console.ReadLine()
        Loop Until IsNumeric(strTheNumber)
        theNumber = strTheNumber
        Return theNumber
    End Function

    Sub Main()
        Dim n As Integer
        n = getUserInput()
        Do While n <> 0
            Console.Write("Το παραγοντικό του " + n.ToString() + "
είναι ")
            Console.WriteLine(calculateParagontiko(n))
            n = getUserInput()
        Loop
    End Sub

End Module
```


Δραστηριότητα 5 : Λογικά λάθη 1.

Στη δραστηριότητα αυτή εξηγείται στους μαθητές η Τρίτη κατηγορία λαθών, τα λογικά λάθη. Σαν άσκηση τους παρουσιάζεται ένα πρόγραμμα που πρέπει να πληκτρολογήσουν. Το πρόγραμμα θα έπρεπε να εμφανίζει το σύνολο των τιμών που είναι αποθηκευμένες σε έναν πίνακα, αλλά εμφανίζει κάτι διαφορετικό. Ζητείται από τους μαθητές να εντοπίσουν το λάθος στον κώδικα.

```
Module Module1

    Sub Main()

        Dim numbers() As Double = {1, 1, 1, 1, 1, 1}
        Dim sum = 0
        For i = 1 To numbers.Length - 1
            sum = sum + numbers(i)
        Next
        Console.WriteLine(sum)
        Console.ReadLine()
    End Sub

End Module
```

End Module

Το λάθος βρίσκεται στη γραμμή

```
For i = 1 To numbers.Length - 1
Η σωστή διατυπωση είναι
For i = 0 To numbers.Length - 1
```

Δραστηριότητα 6 : Λογικά λάθη 2

Στην δραστηριότητα αυτή οι μαθητές καλούνται να αντιμετωπίσουν ένα σύνθετο πρόβλημα. Τους ζητείται να πληκτρολογήσουν μια εφαρμογή η οποία έχει ένα λογικό λάθος. Παράλληλα αυτή η εφαρμογή δεν θεωρείται σταθερή διότι είναι ευάλωτη σε run time errors.

Το λογικό λάθος της εφαρμογής βρίσκεται στην κλήση της συνάρτησης `GetKMPH(hours, kilometers)` . Η σωστή κλήση είναι `GetKMPH(kilometers, hours)` .



ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΕΑΕΚ

ΕΥΡΩΠΑΪΚΗ ΕΝΩΣΗ
ΣΥΓΧΡΗΜΑΤΟΔΟΤΗΣΗ
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Η ΠΑΙΔΕΙΑ ΣΤΗΝ ΚΟΡΥΦΗ
Επιχειρησιακό Πρόγραμμα
Εκπαίδευσης και Αρχικής
Επαγγελματικής Κατάρτισης