Τ.Ε.Ι. ΛΑΜΙΑΣ ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΗΣ

 $\begin{array}{l} \Pi P O \Gamma P A M M A T I \Sigma M O \Sigma \\ V isual Basic Net 2005 \end{array}$

ΣΗΜΕΙΩΣΕΙΣ

ΘΕΩΡΙΑΣ - ΕΡΓΑΣΤΗΡΙΟΥ

Γρηγόρης Τζιάλλας

AAMIA 2007

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1	1 ΕΙΣΑΓΩΓΗ ΣΤΗΝ VI	SUAL BASIC NET 2005 4
	1.1 ТО ПЕРІВАЛЛОН ТН	Σ Visual Basic 2005 Express Edition
	Εκκίνηση της Visual Bas	ic
	Δημιουργία νέας εφαρμο	γής4
	Το πρώτο απλό πρόγραμ	иα6
	Αποθήκευση και εκτέλεσ	η εφαρμογών
	Άλλες επιλογές του Μενο	ύ File
	Συνέχιση εντολής σε επό	μενη γραμμή10
	Συνέχιση εντολής στην ίδ	δια γραμμή
	Σχόλια	
	1.2 ΛΕΙΤΟΥΡΓΙΚΕΣ ΜΟΝ	$IA\Delta E\Sigma (MODULES)10$
	1.3 H KAA Σ H ANTIKEIN	ENΩN CONSOLE
	Οι μέθοδοι Write και Wi	iteLine
	Οι μεθοδος ReadLine	
	1.4 ΑΣΚΗΣΕΙΣ	
2	2 ΜΕΤΑΒΛΗΤΕΣ ΚΑΙ Σ	ΣΤΑΘΕΡΕΣ16
	2.1 Δηλώσεις Μετάβλ	ΗΤΩΝ
	2.2 ΤΥΠΟΙ ΔΕΔΟΜΕΝΩΝ	1
	Ανάθεση τιμής	
	Είσοδος δεδομένων από	το πληκτρολόνιο
	2.3 ΜΕΤΑΤΡΟΠΗ ΔΕΔΟΝ	19
	Συναρτήσεις μετατροπής	μεταβλητών
	Η συνάρτηση Ctype	21
	Η μέθοδος Parse	
	Η μέθοδος ToString	
	2.4 ΣΤΑΘΕΡΕΣ	
	2.5 Ασκήσεις κεφαλά	loy 2
3	3 ΤΕΛΕΣΤΕΣ – ΜΑΘΗ	ΜΑΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ
3	3 ΤΕΛΕΣΤΕΣ – ΜΑΘΗ	ΜΑΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ
3	3 ΤΕΛΕΣΤΕΣ – ΜΑΘΗ 3.1 ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕ	ΜΑΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ
3	 ΤΕΛΕΣΤΕΣ – ΜΑΘΗ 3.1 ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕ 3.2 ΑΛΦΑΡΙΘΜΗΤΙΚΟΙ Τ 2.2 ΤΑ ΕΝΤΕΡ ΕΝΕΡΙΟΝΗΤΙΚΟΙ Τ 	ΜΑΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ
3	 ΤΕΛΕΣΤΕΣ – ΜΑΘΗ 3.1 ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕ 3.2 ΑΛΦΑΡΙΘΜΗΤΙΚΟΙ Τ 3.3 ΤΕΛΕΣΤΕΣ ΣΥΓΚΡΙΣΙ 2.4 ΑΟΕΨΙΟΙ ΤΕΛΕΡΙΣΙΟΝ 	MATIKEΣ ΣΥΝΑΡΤΗΣΕΙΣ
3	 ΤΕΛΕΣΤΕΣ – ΜΑΘΗ 3.1 ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕ 3.2 ΑΛΦΑΡΙΘΜΗΤΙΚΟΙ Τ 3.3 ΤΕΛΕΣΤΕΣ ΣΥΓΚΡΙΣΙ 3.4 ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ 3.5 ΤΕΛΕΣΤΕΣ ΤΑΛΑΘΕΥ 	MATIKEΣ ΣΥΝΑΡΤΗΣΕΙΣ 24 ΣΤΕΣ 24 ΕΛΕΣΤΕΣ 25 ΗΣ 26
3	 3 ΤΕΛΕΣΤΕΣ – ΜΑΘΗ 3.1 Αριφμητικοί τελε 3.2 Αλφαριφμητικοί τ 3.3 Τελεστες σύγκριση 3.4 Λογικοί τελεστες 3.5 Τελεστες αναφεσή 2.6 Χρισμή μαρελίος 	MATIKEΣ ΣΥΝΑΡΤΗΣΕΙΣ
3	 3 ΤΕΛΕΣΤΕΣ – ΜΑΘΗ 3.1 ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕ 3.2 ΑΛΦΑΡΙΘΜΗΤΙΚΟΙ Τ 3.3 ΤΕΛΕΣΤΕΣ ΣΥΓΚΡΙΣΙ 3.4 ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ 3.5 ΤΕΛΕΣΤΕΣ ΑΝΑΘΕΣΙ 3.6 ΧΡΗΣΗ ΠΑΡΕΝΘΕΣΕ 2.7 ΜΑΟΠΑΔΤΙΚΕΣ ΣΥΣ 	MATIKEΣ ΣΥΝΑΡΤΗΣΕΙΣ
3	 3 ΤΕΛΕΣΤΕΣ – ΜΑΘΗ 3.1 ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕ 3.2 ΑΛΦΑΡΙΘΜΗΤΙΚΟΙ Τ 3.3 ΤΕΛΕΣΤΕΣ ΣΥΓΚΡΙΣΙ 3.4 ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ 3.5 ΤΕΛΕΣΤΕΣ ΑΝΑΘΕΣΙ 3.6 ΧΡΗΣΗ ΠΑΡΕΝΘΕΣΕ 3.7 ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΥ 	MATIKEΣ ΣΥΝΑΡΤΗΣΕΙΣ
3	 ΤΕΛΕΣΤΕΣ – ΜΑΘΗ 3.1 ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕ 3.2 ΑΛΦΑΡΙΘΜΗΤΙΚΟΙ Τ 3.3 ΤΕΛΕΣΤΕΣ ΣΥΓΚΡΙΣΙ 3.4 ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ 3.5 ΤΕΛΕΣΤΕΣ ΑΝΑΘΕΣΙ 3.6 ΧΡΗΣΗ ΠΑΡΕΝΘΕΣΕ 3.7 ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΥ 3.8 ΑΣΚΗΣΕΙΣ ΚΕΦΑΛΑ 	ΜΑΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ 24 ΣΤΕΣ 24 ΕΛΕΣΤΕΣ 25 ΙΣ 26 Ν 28 ΗΣ ΤΙΜΗΣ 28 ΩΝ - ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΤΕΛΕΣΤΩΝ 29 ΙΔΑΡΤΗΣΕΙΣ 29 ΙΟΥ 3 31
3	 ΤΕΛΕΣΤΕΣ – ΜΑΘΗ 3.1 ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕ 3.2 ΑΛΦΑΡΙΘΜΗΤΙΚΟΙ Τ 3.3 ΤΕΛΕΣΤΕΣ ΣΥΓΚΡΙΣΙ 3.4 ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ 3.5 ΤΕΛΕΣΤΕΣ ΑΝΑΘΕΣΙ 3.6 ΧΡΗΣΗ ΠΑΡΕΝΘΕΣΕ 3.7 ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΝ 3.8 ΑΣΚΗΣΕΙΣ ΚΕΦΑΛΑ 	MATIKEΣ ΣΥΝΑΡΤΗΣΕΙΣ 24 ΣΤΕΣ 24 ΕΛΕΣΤΕΣ 25 ΗΣ 26 ΗΣ 26 ΗΣ 28 ΗΣ ΤΙΜΗΣ 28 ΩΝ - ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΤΕΛΕΣΤΩΝ 29 ΝΑΛΗΨΗ 32
3	 ΤΕΛΕΣΤΕΣ – ΜΑΘΗ 3.1 ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕ 3.2 ΑΛΦΑΡΙΘΜΗΤΙΚΟΙ Τ 3.3 ΤΕΛΕΣΤΕΣ ΣΥΓΚΡΙΣΙ 3.4 ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ 3.5 ΤΕΛΕΣΤΕΣ ΑΝΑΘΕΣΙ 3.6 ΧΡΗΣΗ ΠΑΡΕΝΘΕΣΕ 3.7 ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΥ 3.8 ΑΣΚΗΣΕΙΣ ΚΕΦΑΛΑ ΕΠΙΛΟΓΗ ΚΑΙ ΕΠΑΝ 4.1 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓ 	MATIKEΣ ΣΥΝΑΡΤΗΣΕΙΣ 24 ΣΤΕΣ 24 ΕΛΕΣΤΕΣ 25 ΗΣ 26 ΗΣ ΤΙΜΗΣ 28 ΟΝ - ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΤΕΛΕΣΤΩΝ 29 ΝΑΛΤΗΨΗ 29 ΗΣ ΙΓ 32
3	 ΤΕΛΕΣΤΕΣ – ΜΑΘΗ 3.1 ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕ 3.2 ΑΛΦΑΡΙΘΜΗΤΙΚΟΙ Τ 3.3 ΤΕΛΕΣΤΕΣ ΣΥΓΚΡΙΣΙ 3.4 ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ 3.5 ΤΕΛΕΣΤΕΣ ΑΝΑΘΕΣΙ 3.6 ΧΡΗΣΗ ΠΑΡΕΝΘΕΣΕ 3.7 ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΥ 3.8 ΑΣΚΗΣΕΙΣ ΚΕΦΑΛΑ ΕΠΙΛΟΓΗ ΚΑΙ ΕΠΑΝ 4.1 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓ <i>Η μορφή If Then End</i> 	MATIKEΣ ΣΥΝΑΡΤΗΣΕΙΣ 24 ΣΤΕΣ 24 ΕΛΕΣΤΕΣ 25 ΙΣ 26 ΙΣ 26 ΙΣ 28 ΙΣ ΤΙΜΗΣ 28 ΩΝ - ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΤΕΛΕΣΤΩΝ 29 ΙΔΑΤΗΨΗ 29 ΙΥ 31 ΝΑΛΗΨΗ 32 ΗΣ ΙΓ 32
3	 ΤΕΛΕΣΤΕΣ – ΜΑΘΗ 3.1 ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕ 3.2 ΑΛΦΑΡΙΘΜΗΤΙΚΟΙ Τ 3.3 ΤΕΛΕΣΤΕΣ ΣΥΓΚΡΙΣΙ 3.4 ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ 3.5 ΤΕΛΕΣΤΕΣ ΑΝΑΘΕΣΙ 3.6 ΧΡΗΣΗ ΠΑΡΕΝΘΕΣΕ 3.7 ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΥ 3.8 ΑΣΚΗΣΕΙΣ ΚΕΦΑΛΑ 4 ΕΠΙΛΟΓΗ ΚΑΙ ΕΠΑΝ 4.1 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓ <i>Η μορφή If Then Encli</i> <i>Η μορφή If Then Els</i> 	MATIKEΣ ΣΥΝΑΡΤΗΣΕΙΣ 24 ΣΤΕΣ 24 ΕΛΕΣΤΕΣ 25 ΙΣ 26 ΙΣ 26 ΙΣ 28 ΙΣ ΤΙΜΗΣ 28 ΙΣ ΤΙΜΗΣ 28 ΙΩΝ - ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΤΕΛΕΣΤΩΝ 29 ΙΔΑΥΗΨΗ 29 ΙΟΥ 3 31 ΝΑΛΗΨΗ 32 μΣ ΙΓ 32 μ 32 μ 32 μ 32
4	 ΤΕΛΕΣΤΕΣ – ΜΑΘΗ 3.1 ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕ 3.2 ΑΛΦΑΡΙΘΜΗΤΙΚΟΙ Τ 3.3 ΤΕΛΕΣΤΕΣ ΣΥΓΚΡΙΣΙ 3.4 ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ 3.5 ΤΕΛΕΣΤΕΣ ΑΝΑΘΕΣΙ 3.6 ΧΡΗΣΗ ΠΑΡΕΝΘΕΣΕ 3.7 ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΥ 3.8 ΑΣΚΗΣΕΙΣ ΚΕΦΑΛΑΙ ΕΠΙΛΟΓΗ ΚΑΙ ΕΠΑΝ 4.1 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓ Η μορφή If Then End Η μορφή If Then Els Ένθεση εντολών If The 	MATIKEΣ ΣΥΝΑΡΤΗΣΕΙΣ 24 ΣΤΕΣ 24 ΕΛΕΣΤΕΣ 25 ΙΣ 26 Ν 28 ΗΣ ΤΙΜΗΣ 28 ΩΝ - ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΤΕΛΕΣΤΩΝ 29 ΙΔΑΤΗΨΗ 29 ΙΝΑΛΗΨΗ 32 ΗΣ ΙΓ 32 μ 33
4	 ΤΕΛΕΣΤΕΣ – ΜΑΘΗ 3.1 ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕ 3.2 ΑΛΦΑΡΙΘΜΗΤΙΚΟΙ Τ 3.3 ΤΕΛΕΣΤΕΣ ΣΥΓΚΡΙΣΙ 3.4 ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ 3.5 ΤΕΛΕΣΤΕΣ ΑΝΑΘΕΣΙ 3.6 ΧΡΗΣΗ ΠΑΡΕΝΘΕΣΕ 3.7 ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΥ 3.8 ΑΣΚΗΣΕΙΣ ΚΕΦΑΛΑΑ ΕΠΙΛΟΓΗ ΚΑΙ ΕΠΑΝ 4.1 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓ <i>Η μορφή If Then End</i> <i>Η μορφή If Then Els</i> <i>Ένθεση εντολών If The</i> 	MATIKEΣ ΣΥΝΑΡΤΗΣΕΙΣ 24 ΣΤΕΣ 24 ΕΛΕΣΤΕΣ 25 ΙΣ 26 ΙΣ 28 ΗΣ ΤΙΜΗΣ 28 ΩΝ - ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΤΕΛΕΣΤΩΝ 29 ΝΑΛΗΨΗ 29 ΝΑΛΗΨΗ 32 ΗΣ ΙΓ ΤΗΕΝ 32 μ Γ
4	 ΤΕΛΕΣΤΕΣ – ΜΑΘΗ 3.1 ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕ 3.2 ΑΛΦΑΡΙΘΜΗΤΙΚΟΙ Τ 3.3 ΤΕΛΕΣΤΕΣ ΣΥΓΚΡΙΣΙ 3.4 ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ 3.5 ΤΕΛΕΣΤΕΣ ΑΝΑΘΕΣΙ 3.6 ΧΡΗΣΗ ΠΑΡΕΝΘΕΣΕ 3.7 ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΥ 3.8 ΑΣΚΗΣΕΙΣ ΚΕΦΑΛΑΑ ΕΠΙΛΟΓΗ ΚΑΙ ΕΠΑΝ 4.1 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓ <i>Η μορφή If</i> Then End <i>Η μορφή If</i> Then End <i>Η μορφή If</i> Then Els <i>Ένθεση εντολών If</i> The 4.2 Η ΕΝΤΟΛΗ ΠΟΛΛΑΓ 	MATIKEΣ ΣΥΝΑΡΤΗΣΕΙΣ 24 ΣΤΕΣ 24 ΕΛΕΣΤΕΣ 25 ΗΣ 26
4	 ΤΕΛΕΣΤΕΣ – ΜΑΘΗ 3.1 ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕ 3.2 ΑΛΦΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕ 3.3 ΤΕΛΕΣΤΕΣ ΣΥΓΚΡΙΣΙ 3.4 ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ 3.5 ΤΕΛΕΣΤΕΣ ΑΝΑΘΕΣΙ 3.6 ΧΡΗΣΗ ΠΑΡΕΝΘΕΣΕ 3.7 ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΥ 3.8 ΑΣΚΗΣΕΙΣ ΚΕΦΑΛΑ ΕΠΙΛΟΓΗ ΚΑΙ ΕΠΑΝ 4.1 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓ Η μορφή If Then Elst Ένθεση εντολών If The Ένθεση με τη μορφή If 4.2 Η ΕΝΤΟΛΗ ΠΙΛΛΑΙ 4.3 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 	MATIKEΣ ΣΥΝΑΡΤΗΣΕΙΣ 24 ΣΤΕΣ 24 ΕΛΕΣΤΕΣ 25 ΗΣ 26
4	 ΤΕΛΕΣΤΕΣ – ΜΑΘΗ 3.1 ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕ 3.2 ΑΛΦΑΡΙΘΜΗΤΙΚΟΙ Τ 3.3 ΤΕΛΕΣΤΕΣ ΣΥΓΚΡΙΣΙ 3.4 ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ 3.5 ΤΕΛΕΣΤΕΣ ΑΝΑΘΕΣΙ 3.6 ΧΡΗΣΗ ΠΑΡΕΝΘΕΣΕ 3.7 ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΥ 3.8 ΑΣΚΗΣΕΙΣ ΚΕΦΑΛΑ ΕΠΙΛΟΓΗ ΚΑΙ ΕΠΑΝ 4.1 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓ Η μορφή If Then Els Ενθεση με τη μορφή If 4.2 Η ΕΝΤΟΛΗ ΠΟΛΛΑΓ 4.3 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 4.4 Η ΕΝΤΟΛΗ ΠΟΛΛΑΓ 	MATIKEΣ ΣΥΝΑΡΤΗΣΕΙΣ 24 ΣΤΕΣ 24 ΕΛΕΣΤΕΣ 25 ΙΣ 26
4	 ΤΕΛΕΣΤΕΣ – ΜΑΘΗ 3.1 ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕ 3.2 ΑΛΦΑΡΙΘΜΗΤΙΚΟΙ Τ 3.3 ΤΕΛΕΣΤΕΣ ΣΥΓΚΡΙΣΙ 3.4 ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ 3.5 ΤΕΛΕΣΤΕΣ ΑΝΑΘΕΣΙ 3.6 ΧΡΗΣΗ ΠΑΡΕΝΘΕΣΕ 3.7 ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΥ 3.8 ΑΣΚΗΣΕΙΣ ΚΕΦΑΛΑ ΕΠΙΛΟΓΗ ΚΑΙ ΕΠΑΝ 4.1 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓ <i>Η μορφή If Then Εικ</i> <i>Ενθεση με τη μορφή If Τhe</i> <i>Ενθεση με τη μορφή If</i> 4.2 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 4.4 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 4.5 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 	MATIKEΣ ΣΥΝΑΡΤΗΣΕΙΣ
4	 ΤΕΛΕΣΤΕΣ – ΜΑΘΗ 3.1 ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕ 3.2 ΑΛΦΑΡΙΘΜΗΤΙΚΟΙ Τ 3.3 ΤΕΛΕΣΤΕΣ ΣΥΓΚΡΙΣΙ 3.4 ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ 3.5 ΤΕΛΕΣΤΕΣ ΑΝΑΘΕΣΙ 3.6 ΧΡΗΣΗ ΠΑΡΕΝΘΕΣΕ 3.7 ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΥ 3.8 ΑΣΚΗΣΕΙΣ ΚΕΦΑΛΑ 4 ΕΠΙΛΟΓΗ ΚΑΙ ΕΠΑΝ 4.1 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓ Η μορφή If Then Elss Ένθεση εντολών If The Ένθεση με τη μορφή If Τhe 4.2 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 4.3 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 4.4 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 4.5 Η ΕΝΤΟΛΗ ΕΠΑΝΑΛ 	MATIKEΣ ΣΥΝΑΡΤΗΣΕΙΣ 24 ΣΤΕΣ 24 ΕΛΕΣΤΕΣ 25 ΗΣ 26
3	 ΤΕΛΕΣΤΕΣ – ΜΑΘΗ 3.1 ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕ 3.2 ΑΛΦΑΡΙΘΜΗΤΙΚΟΙ Τ 3.3 ΤΕΛΕΣΤΕΣ ΣΥΓΚΡΙΣΙ 3.4 ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ 3.5 ΤΕΛΕΣΤΕΣ ΑΝΑΘΕΣΙ 3.6 ΧΡΗΣΗ ΠΑΡΕΝΘΕΣΕ 3.7 ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΥ 3.8 ΑΣΚΗΣΕΙΣ ΚΕΦΑΛΑ 4 ΕΠΙΛΟΓΗ ΚΑΙ ΕΠΑΝ 4.1 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓ Η μορφή If Then Els Ένθεση με τη μορφή If The Ένθεση με τη μορφή If 4.2 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 4.3 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 4.4 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 4.5 Η ΕΝΤΟΛΗ ΕΠΑΝΑΛ Ένθεση εντολών For 4.6 Η ΕΝΤΟΛΗ ΕΠΑΝΑΛ 	MATIKES ΣΥΝΑΡΤΗΣΕΙΣ 24 ΣΤΕΣ 24 ΕΛΕΣΤΕΣ 25 ΙΣ 26 ΠΕ 28 ΗΣ ΤΙΜΗΣ 28 ΩΝ - ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΤΕΛΕΣΤΩΝ 29 ΝΑΤΙΚΕΣ 29 ΟΥ 3 31 ΝΑΛΗΨΗ 32 ΗΣ ΙΓ. ΤΗΕΝ 32 <i>μ</i> Γ
3	 ΤΕΛΕΣΤΕΣ – ΜΑΘΗ 3.1 ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕ 3.2 ΑΛΦΑΡΙΘΜΗΤΙΚΟΙ Τ 3.3 ΤΕΛΕΣΤΕΣ ΣΥΓΚΡΙΣΙ 3.4 ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ 3.5 ΤΕΛΕΣΤΕΣ ΑΝΑΘΕΣΙ 3.6 ΧΡΗΣΗ ΠΑΡΕΝΘΕΣΕ 3.7 ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΥ 3.8 ΑΣΚΗΣΕΙΣ ΚΕΦΑΛΑ ΕΠΙΛΟΓΗ ΚΑΙ ΕΠΑΝ 4.1 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 4.1 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 4.1 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 4.1 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 4.2 Η ΕΝΤΟΛΗ ΠΟΛΛΑΓ 4.3 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 4.4 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 4.4 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 4.5 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 4.6 Η ΕΝΤΟΛΗ ΕΠΑΝΑΛ 4.7 Η ΕΝΤΟΛΗ ΕΠΑΝΑΛ 	MATIKEΣ ΣΥΝΑΡΤΗΣΕΙΣ 24 ΣΤΕΣ 24 ΕΛΕΣΤΕΣ 25 ΗΣ 26
3	 ΤΕΛΕΣΤΕΣ – ΜΑΘΗ 3.1 ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕ 3.2 ΑΛΦΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕ 3.3 ΤΕΛΕΣΤΕΣ ΣΥΓΚΡΙΣΙ 3.4 ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ 3.5 ΤΕΛΕΣΤΕΣ ΑΝΑΘΕΣΙ 3.6 ΧΡΗΣΗ ΠΑΡΕΝΘΕΣΕ 3.7 ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΥ 3.8 ΑΣΚΗΣΕΙΣ ΚΕΦΑΛΑΑ ΕΠΙΛΟΓΗ ΚΑΙ ΕΠΑΝ 4.1 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓ Η μορφή If Then End Η μορφή If Then End Η μορφή If Then End Η μορφή If Then Els Ένθεση εντολών If The Ένθεση με τη μορφή If 4.2 Η ΕΝΤΟΛΗ ΠΟΛΛΑΙ 4.3 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 4.4 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 4.5 Η ΕΝΤΟΛΗ ΕΠΑΝΑΛ 4.6 Η ΕΝΤΟΛΗ ΕΠΑΝΑΛ 4.8 Η ΕΝΤΟΛΗ ΕΠΑΝΑΛ 	MATIKEΣ ΣΥΝΑΡΤΗΣΕΙΣ 24 ΣΤΕΣ 24 ΕΛΕΣΤΕΣ 25 ΗΣ 26
3	 ΤΕΛΕΣΤΕΣ – ΜΑΘΗ 3.1 ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕ 3.2 ΑΛΦΑΡΙΘΜΗΤΙΚΟΙ ΤΑΑΕ 3.3 ΤΕΛΕΣΤΕΣ ΣΥΓΚΡΙΣΙ 3.4 ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ 3.5 ΤΕΛΕΣΤΕΣ ΑΝΑΘΕΣΙ 3.6 ΧΡΗΣΗ ΠΑΡΕΝΘΕΣΕ 3.7 ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΥ 3.8 ΑΣΚΗΣΕΙΣ ΚΕΦΑΛΑΕ 4 ΕΠΙΛΟΓΗ ΚΑΙ ΕΠΑΝ 4.1 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ <i>Η μορφή If Then Εικ Ενθεση εντολών If Thee Ενθεση εντολών If Thee Ενθεση εντολών If Thee</i> 4.2 Η ΕΝΤΟΛΗ ΠΟΛΛΑΓ 4.3 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 4.4 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 4.4 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 4.5 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 4.6 Η ΕΝΤΟΛΗ ΕΠΑΝΑΛ 4.7 Η ΕΝΤΟΛΗ ΕΠΑΝΑΛ 4.8 Η ΕΝΤΟΛΗ ΕΠΑΝΑΛ 4.9 Η ΕΝΤΟΛΗ GOTO	MATIKEΣ ΣΥΝΑΡΤΗΣΕΙΣ 24 ΣΤΕΣ 24 ΕΛΕΣΤΕΣ 25 ΙΣ 26 Π 28 Ν ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΤΕΛΕΣΤΩΝ Ν ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΤΕΛΕΣΤΩΝ Ν Α Ν ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΤΕΛΕΣΤΩΝ Ν Α Ν
3	 ΤΕΛΕΣΤΕΣ – ΜΑΘΗ 3.1 ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕ 3.2 ΑΛΦΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕ 3.3 ΤΕΛΕΣΤΕΣ ΣΥΓΚΡΙΣΙ 3.4 ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ 3.5 ΤΕΛΕΣΤΕΣ ΑΝΑΘΕΣΙ 3.6 ΧΡΗΣΗ ΠΑΡΕΝΘΕΣΕ 3.7 ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΥ 3.8 ΑΣΚΗΣΕΙΣ ΚΕΦΑΛΑ ΕΠΙΛΟΓΗ ΚΑΙ ΕΠΑΝ 4.1 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓ <i>Η μορφή If Then Επα</i> <i>Η μορφή If Then Επα</i> <i>Ενθεση εντολών If The</i> <i>Σνθεση εντολών If Τhe</i> <i>Ενθεση εντολών Fo</i> 4.2 Η ΕΝΤΟΛΗ ΠΟΛΛΑΓ 4.3 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 4.4 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 4.4 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 4.5 Η ΕΝΤΟΛΗ ΕΠΙΛΟΓΗ 4.6 Η ΕΝΤΟΛΗ ΕΠΙΛΟΛΑΓ 4.7 Η ΕΝΤΟΛΗ ΕΠΑΝΑΛ 4.8 Η ΕΝΤΟΛΗ ΕΠΑΝΑΛ 4.9 Η ΕΝΤΟΛΗ ΕΠΑΝΑΛ 	MATIKEΣ ΣΥΝΑΡΤΗΣΕΙΣ

5.1	Μονοδιαστατοι πινακές	
5.2	Δισδιαστατοι πινακές	
5.3	ΤΑΞΙΝΟΜΗΣΗ, ΚΑΙ ΑΝΑΖΗΤΗΣΗ ΣΤΟΙΧΕΙΩΝ ΠΙΝΑΚΩΝ	
5.4	ΔΥΝΑΜΙΚΗ ΤΡΟΠΟΠΟΙΗΣΗ ΤΟΥ ΜΕΓΕΘΟΥΣ ΕΝΟΣ ΠΙΝΑΚΑ	
5.5	Ασκήσεις Κεφαλαίου 5	
6 П	ΑΡΑΘΥΡΙΚΕΣ ΕΦΑΡΜΟΓΕΣ	
6.1	ΔΗΜΙΟΥΡΓΙΑ ΠΑΡΑΘΥΡΙΚΩΝ ΕΦΑΡΜΟΓΩΝ	
6.2	Σχεδιασμός του παραθυρού χρηστη	59
Et	πκέτα	60
П	λαίσιο κειμένου	61
П	λήκτρα και πλαίσια επιλογής	
Ke	ατάλογος επιλογής και αναδιπλούμενος κατάλογος	
П	λαίσιο στοιχείων ελέγχου	67
П	λήκτρο	68
6.3	ΔΙΑΣΥΝΔΕΣΗ ΣΥΜΒΑΝΤΩΝ ΕΝΟΣ ΠΑΡΑΘΥΡΟΥ	69
6.4	ΧΡΗΣΗ ΠΟΛΛΑΠΛΩΝ ΠΑΡΑΘΥΡΩΝ ΣΕ ΜΙΑ ΕΦΑΡΜΟΓΗ	71
Eμ	μφάνιση πολλαπλών παραθύρων του ιδίου τύπου	
6.5	Εισόδος δεδομένων με την μέθοδο ΙνρυτΒόχ	
6.6	Εμφανιση μηνυματών στον χρηστη	74
6.7	ΠΑΡΑΔΕΙΓΜΑ ΠΑΡΑΘΥΡΙΚΗΣ ΕΦΑΡΜΟΓΗΣ	
6.8	Ασκήσεις Κεφαλαίου 6	
7 Y	ΠΟΠΡΟΓΡΑΜΜΑΤΑ ΚΑΙ ΜΕΘΟΔΟΙ	
7.1	ΥΠΟΠΡΟΓΡΑΜΜΑΤΑ ΣΕ ΛΕΙΤΟΥΡΓΙΚΕΣ ΜΟΝΑΔΕΣ	
Δι	αδικασίες	
Π	αράμετροι υποπρογραμμάτων	
Σι	ναρτήσεις	
A	ναδρομικές συναρτήσεις	
Σι	τατικές μεταβλητές υποπρογραμμάτων	86
7.2	ΟΡΓΑΝΩΣΗ ΥΠΟΠΡΟΓΡΑΜΜΑΤΩΝ ΚΑΙ ΔΕΔΟΜΕΝΩΝ ΣΕ ΛΕΙΤΟΥΡΓΙΚΕΣ ΜΟΝΑΔΕΣ	
7.3	Μεθοδοι κλάσεων αντικειμένων	
7.4	Ασκήσεις Κεφαλαίου 7	

Εισαγωγή στην Visual Basic Net 2005

1.1 Το περιβάλλον της Visual Basic 2005 Express Edition

Στο μάθημα του Προγραμματισμού διδάσκεται η γλώσσα Visual Basic και χρησιμοποιείται το περιβάλλον ανάπτυξης εφαρμογών Visual Basic 2005 Express Edition. Η έκδοση αυτή της Visual Basic διατίθεται ελεύθερα στο διαδίκτυο στην ηλεκτρονική διεύθυνση: http://msdn.microsoft.com/vstudio/express/vb/download/

Εκκίνηση της Visual Basic

Η εκκίνηση της Visual Basic γίνεται από το εικονίδιο «Microsoft Visual Basic 2005 Express Edition» της επιφάνειας εργασίας ή την επιλογή από το μενού «Έναρξη» «Έναρξη»

«Όλα τα προγράμματα»

«Microsoft Visual Basic 2005 Express Edition»

Δημιουργία νέας εφαρμογής

Η δημιουργία μια νέας εφαρμογής της Visual Basic, γίνεται με την επιλογή «New Project» του μενού «File» (εικόνα 1.1). Εναλλακτικά μπορούμε να χρησιμοποιηθεί η συντόμευση Ctrl-N.



Εικόνα 1.1 Η αρχική σελίδα της Visual Basic

Στην συνέχεια εμφανίζεται ένα πλαίσιο διαλόγου (εικόνα 1.2) με τον Τίτλο «New Project». Με την επιλογή «Console Application» στο πάνω παράθυρο του πλαισίου διαλόγου με τον τίτλο «Template» δημιουργείται μια νέα εφαρμογή τύπου «Console Application». Ο τύπος της εφαρμογής αυτής είναι ο απλούστερος τύπος εφαρμογής και χρησιμοποιείται όταν δεν χρειάζεται το γραφικό περιβάλλον και οι φόρμες των Windows για την επικοινωνία (διεπαφή) με τον χρήστη.

Στο κάτω τμήμα του πλαισίου διαλόγου γίνεται η εισαγωγή του ονόματος της εφαρμογής. (π.χ. lab_1_askisi_1 για την πρώτη άσκηση του πρώτου εργαστηρίου) και στην συνέχεια με το πάτημα του πλήκτρο ΟΚ ολοκληρώνεται η δημιουργία της εφαρμογής (εικόνα 1.2).

Ne	w Project						? X
I	emplates:						0 0 6-6- 0 0 6-6- 6-6-
	Visual Studio i	nstalled templa	tes				
	■VB	WB					
	Windows Application	Class Library	Console Application	My Movie Collecti	Screen Saver Starter Kit		
	My Templates						
	Search Online Templates						
	A project for creati	ng a command-line	application				
N	lame:	lab_1_ask_1					
-						ж	Cancel

Εικόνα 1.2 Καθορισμός ονόματος και τύπου εφαρμογής



Εικόνα 1.3 Η νέα εφαρμογή

Το πρώτο απλό πρόγραμμα

Οι εφαρμογές της Visual Basic αποτελούνται από ένα ή περισσότερα αντικείμενα της Visual Basic (όπως modules, classes, windows forms, dialogs, user controls κλπ). Τα αντικείμενα αυτά μπορούν να επιλεγούν και να εμφανισθούν από το παράθυρο του "Solution Explorer" ο οποίος εμφανίζεται αρχικά στο δεξιό τμήμα της οθόνης.

Το παράθυρο του "Solution Explorer" εμφανίζεται με την επιλογή από το μενού "View" της επιλογής "Solution Explorer" (συντόμευση Ctrl-Alt-L).

Με την δημιουργία μιας εφαρμογής τύπου «Console Application» δημιουργούνται αυτόματα δύο αντικείμενα: Το «My Project» το οποίο χρησιμοποιείται για την αποθήκευση των ρυθμίσεων της εφαρμογής και η λειτουργική μονάδα της Visual Basic «Module1» την οποία μπορούμε να χρησιμοποιήσουμε για να ορίσουμε μεταβλητές, διαδικασίες, και συναρτήσεις.

Με την επιλογή "Properties window" του μενού "View" (συντόμευση F4) μπορούμε να εμφανίσουμε το παράθυρο ιδιοτήτων των αντικειμένων της Visual Basic (εικόνα 1.4).



Εικόνα 1.4 Το παράθυρο ιδιοτήτων

Με το παράθυρο ιδιοτήτων μπορούμε να εμφανίσουμε και να αλλάξουμε το όνομα και τις ιδιότητες των αντικειμένων της Visual Basic. Για παράδειγμα εάν θέλουμε να μετονομάσουμε το αντικειμένου "Module1" σε «FirstProgram», επιλέγουμε το αντικείμενο «Module1» και στην συνέχεια αλλάζουμε την τιμή της ιδιότητας «FileName» από «Module1.vb» σε «FirstProgram.vb»

Παράδειγμα ενός απλού προγράμματος Visual Basic το οποίο εκτυπώνει ένα μήνυμα είναι το παρακάτω:

Module FirstProgram

```
Sub Main()
Console.WriteLine("Welcome")
End Sub
```

End Module

Παράδειγμα 1.1

Το πρόγραμμα αυτό χρησιμοποιεί την εντολή WriteLine της κλάσης αντικειμένων Console για να εκτυπώσει ένα μήνυμα στην οθόνη.

Αποθήκευση και εκτέλεση εφαρμογών

Με την επιλογή «Save All» του μενού «File γίνεται η αποθήκευση των εφαρμογών της Visual Basic (συντόμευση Ctrl-Shift-S).

Στο παράθυρο διαλόγου με το τίτλο «Save Project» (εικόνα 3) το οποίο εμφανίζεται γίνεται ο ορισμός του ονόματος της εφαρμογής, της διαδρομής και του ονόματος του καταλόγου όπου θα αποθηκευθεί η εφαρμογή.

Save Project		? X
<u>N</u> ame:	lab_1_ask_1	
Location:	C:\Documents and Settings\To έγγραφά μου\Visual Studio 2005\Projects	Browse
Solution Name:	lab_1_ask_1 Image: Create directory for solution	
	Save	Cancel

Εικόνα 1.5 Δημιουργία νέας εφαρμογής

Η εκτέλεση της εφαρμογής γίνεται με την επιλογή «Start Debugging» από το μενού «Debug» (συντόμευση F5). Η επιλογή «Start Debugging» χρησιμοποιείται για να ελέγξουμε την λειτουργία ενός προγράμματος και να διορθώσουμε τα λάθη που εμφανίζονται κατά την εκτέλεση του προγράμματος. Εναλλακτικά μπορεί να χρησιμοποιηθεί η επιλογή «Start Without Debugging» από το μενού «Debug» (συντόμευση Ctrl-F5).

Πριν από την εκτέλεση του προγράμματος, γίνεται ο συντακτικός έλεγχος του. Εάν υπάρχουν συντακτικά λάθη, τότε εμφανίζεται ένα πλαίσιο διαλόγου το οποίο ενημερώνει τον χρήστη ότι υπάρχουν λάθη. Με την επιλογή «No» του πλαισίου διαλόγου αναιρείται η εκτέλεση του προγράμματος ώστε να γίνει πρώτα η διόρθωση των συντακτικών λαθών τα οποία εμφανίζονται στο κάτω τμήμα του παραθύρου της Visual Basic με τον τίτλο «Error List».

Το παράθυρο «Error List» εμφανίζει λάθη, προειδοποιήσεις και μηνύματα σχετικά με την σύνταξη του προγράμματος. Με την χρήση του αντίστοιχου εικονιδίου (errors, warnings and messages) γίνεται η επιλογή της εμφάνισης ή της απόκρυψης τους.

Ο έλεγχος της σύνταξης ενός προγράμματος ταυτόχρονα με την σύνταξη του προγράμματος. Τα συντακτικά λάθη υπογραμμίζονται και εμφανίζονται ταυτόχρονα και στο παράθυρο «Error List» (εικόνα 1.6).

📴 lab_1_ask_1 - Microsoft Visual Basic 2005 Express Ed	ition				_8×
File Edit View Project Build Debug Data Tools	Window Community Help				
🛅 🗃 🖽 • 🛃 🕔 🗼 🐘 🚨 📓 🗏 🔊	- (* -) II II 9I (I * *) -	2 🖄 🛠	· 诸 🖕		
firstProgram.vb				-	🗙 Solution Explorer 🛛 👻 🕂 🗙
g 🎼 firstProgram	💌 🎁 (Declarations)				T 🗟 🗿 🖬 🗉
Module firstProgram Sub Hain() Console.WriteLin("Welcome End Sub End Module End Module Fror List Description OMessages		dition vu like to contin Yes	ue and run the	X] Protect	
3 'WriteLin' is not a member of 'System.Console'.	firstProgram.vb	4	9	lab_1_ask_1	
Timmediate Window Cutout					
Build failed					

Εικόνα 1.6 Συντακτικά λάθη

Με την εκτέλεση του προγράμματος του παραδείγματος 1.1 εμφανίζεται το παρακάτω παράθυρο τερματικού (Console Application) με το μήνυμα «Welcome».



Εικόνα 1.7 Το αποτέλεσμα της εκτέλεσης του παραδείγματος 1.1

Η εκτέλεση ενός προγράμματος μπορεί να γίνει βήμα προς βήμα με την επιλογή «Step Into» (συντόμευση F8) του μενού «Debug».

Με την επιλογή «Toggle Breakpoint» του μενού «Debug» ορίζονται ή να αναιρούνται σημεία όπου η εκτέλεση του προγράμματος διακόπτεται. Με την διακοπή του προγράμματος στα συγκεκριμένα σημεία μπορεί γίνει ο έλεγχος της ροή του προγράμματος, των τιμών των μεταβλητών και των ιδιοτήτων των διαφόρων αντικειμένων του προγράμματος. Η εκτέλεση του προγράμματος συνεχίζεται με την επιλογή «Continue» του μενού «Debug» (συντόμευση F5) ή με την επιλογή «Step Into» (συντόμευση F8).

Κατά την διάρκεια της διακοπής της εκτέλεσης του προγράμματος δίνεται η δυνατότητα επιλογής και εμφάνισης των παραθύρων "Locals", "Call Stack", "Watch" και "Immediate".



Εικόνα 1.8 Διακοπή εκτέλεσης προγράμματος

Τα παράθυρα αυτά μπορούν να επιλεγούν με τις επιλογές «Locals», «Call Stack», «Watch» και «Immediate» του μενού «Debug»-«Windows» ή με την επιλογή του τίτλου τους ο οποίος εμφανίζεται στο κάτω αριστερό τμήμα του παραθύρου του περιβάλλοντος της Visual Basic.

Το παράθυρο «Locals» εμφανίζει τις τιμές των μεταβλητών και τα ιδιοτήτων των αντικειμένων του τμήματος του προγράμματος το οποίο εκτελείται.

Το παράθυρο «Call Stack» εμφανίζει τις διαδοχικές κλήσεις μεθόδων και υποπρογραμμάτων οι οποίες έχουν γίνει.

Το παράθυρο «Watch» εμφανίζει τις τιμές των μεταβλητών και τις ιδιότητες των αντικειμένων του υποπρογράμματος το οποίο έχει επιλεγεί με την επιλογή «Add Watch» του μενού «Debug».

Το παράθυρο «Immediate» επιτρέπει την εισαγωγή και εκτέλεση εντολών. Για παράδειγμα (εικόνα 1.8) μπορεί να γίνει η εισαγωγή της παρακάτω εντολής

Console.WriteLine("Welcome again")

και η εκτέλεση της με το πάτημα του πλήκτρου {Enter}.

Κατά την διάρκεια της διακοπής ενός προγράμματος είναι δυνατόν να γίνουν αλλαγές και διορθώσεις στο πρόγραμμα. Με τον τρόπο αυτό μπορεί να γίνει ταυτόχρονα ο έλεγχος και η διόρθωση των λογικών λαθών ενός προγράμματος χωρίς να χρειάζεται η επανεκκίνηση του.

Άλλες επιλογές του Μενού File

Με την εντολή "Close" του μενού "File" κλείνει η τρέχουσα εφαρμογή.

Με την εντολή "Open" του μενού "File" ανοίγει μια εφαρμογή που είχε δημιουργηθεί και αποθηκευθεί παλαιότερα.

Συνέχιση εντολής σε επόμενη γραμμή

Κατά την σύνταξη ενός προγράμματος η κάθε δήλωση ή εντολή καταλαμβάνει μια γραμμή κειμένου. Για να χρησιμοποιηθούν περισσότερες από μια γραμμές θα πρέπει να εισαχθεί ο χαρακτήρας '_ ' με τον οποίο δηλώνεται ότι η σύνταξη της δήλωσης συνεχίζει στην επόμενη γραμμή.

Συνέχιση εντολής στην ίδια γραμμή

Για την εισαγωγή περισσότερων από μία εντολών στην ίδια γραμμή κειμένου χρησιμοποιείται ο χαρακτήρας ': '

```
Console.Write("Εργαστήριο "):Console.WriteLine("Προγραμματισμού")
```

Σχόλια

Τα σχόλια στην Visual Basic εισάγονται με το σύμβολο « ' » ή την λέξη REM.

```
Console.Write("Εργαστήριο ") ' Εκτύπωση μηνύματος
```

1.2 Λειτουργικές μονάδες (Modules)

Τα βασικά στοιχεία μιας λειτουργικής μονάδα (module) της Visual Basic είναι:

- Η δήλωση του ονόματος της. (πχ Module FisrtProgram). Το όνομα μιας λειτουργικής μονάδας πρέπει να είναι το ίδιο με το όνομα του αρχείου το οποίο χρησιμοποιείται για την αποθήκευση της.
- Δηλώσεις μεταβλητών και σταθερών της μονάδος
- Δηλώσεις διαδικασιών. Η δήλωση μιας διαδικασίας με την δεσμευμένη λέξη «Sub» και το όνομα της διαδικασίας. Το τέλος μιας διαδικασίας ορίζεται με την δήλωση «End Sub".
- Δηλώσεις συναρτήσεων. Η δήλωση μιας συνάρτησης γίνεται με την δεσμευμένη λέξη «Function», την δήλωση του ονόματος της και τον τύπο της μεταβλητής την οποία επιστρέφει. Το τέλος μιας συνάρτησης ορίζεται με την δήλωση «End Function».
- Η δήλωση «End module» η οποία ορίζει το τέλος της λειτουργικής μονάδας.

Mod	le I	FirstProgram	'Δήλωση	ονόματος μονάδας
	Sub	Main()	'Διαδικ	ασία main
		Console.WriteLine("Welcome")		
	End	Sub	' Τέλος	διαδικασίας main
End	Modu	le	' Τέλος	module

Τα ονόματα των προγραμμάτων, διαδικασιών, μεταβλητών και άλλων αντικειμένων αρχίζουν με λατινικούς χαρακτήρες ή το χαρακτήρα «_», μπορούν να περιέχουν αριθμητικούς χαρακτήρες (αλλά όχι στην αρχή), και δεν πρέπει περιέχουν κενά ή άλλους ειδικούς χαρακτήρες. Το ονόματα επίσης δεν πρέπει να είναι δεσμευμένες λέξεις της Visual Basic.

Ένα πρόγραμμα μπορεί να έχει πολλές λειτουργικές μονάδες. Η πρόσθεση νέων λειτουργικών μονάδων γίνεται με την επιλογή «Add Module» του μενού «Project». Στο παρακάτω πλαίσιο διαλόγου επιλέγεται ο τύπο του αντικειμένου ο οποίος θα προστεθεί στην εφαρμογή (στην συγκεκριμένη περίπτωση ο προεπιλεγμένος τύπος είναι Module) και το όνομα του.

A	dd New Item - la	b_1_ask_1						? X
	Templates:							00 00
	Visual Studio i	nstalled templa	tes					-
							Tabe	
	Windows Form	Dialog	Explorer Form	MDI Parent Form	About Box	Login Form	Splash Screen	
	V B	1						
	Class	Module	DataSet	SQL Database	User Control	Inherited Form	Inherited User Control	
	Tavt File							
	Texchie							•
	A file for storing gr	oups of functions						
ļ	<u>N</u> ame:	Module1.vb						
							Add	Cancel

Εικόνα 1.9 Διακοπή εκτέλεσης προγράμματος

Οι εφαρμογές τύπου Console Application έχουν μια αρχική λειτουργική μονάδα με την οποία ξεκινά η εκτέλεση τους. Η αρχική λειτουργική μονάδα πρέπει να περιέχει μια διαδικασία με το όνομα «Main» η οποία είναι η πρώτη διαδικασία η οποία θα εκτελεσθεί.

Ο ορισμός της αρχικής μονάδας μιας εφαρμογής γίνεται με την επιλογή (διπλό πάτημα του αριστερού πλήκτρου του ποντικιού) του αντικειμένου "MyProject" στο παράθυρο "Solution Explorer". Στη συνέχεια εμφανίζεται το παράθυρο ρύθμισης των ιδιοτήτων της εφαρμογής (εικόνα 1.9) με το οποίο μπορούμε να επιλέξουμε το αρχικό αντικείμενο εκκίνησης της εφαρμογής.

Application*	Assembly name:	Root namespace:			4	Bab_1_ask_1 Bab_1_ask_1
Iompile	lab_1_ask_1	lab_1_ask_1				🦾 🛅 firstProgram.vb
Debug	Application type:	Icon:	12			
References		(Derault Icon)				
Resources	Startup object: firstProgram	Assembly Informatio	n			
settings	Enable application framework					
iigning	Windows application framework properties —					
Security	Enable XP visual styles					
Publish	Make single instance application					
~	Save My. Settings on Shutdown					
	Authentication mode:	-				Properties
	Chutdows mode					
	When startup form closes	~				Ca. 22
	Splash screen:					
	(None)	View App	lication Events	1		
List	0.Warringa (A) 0.Massagas				~ ₽	×.
Description	o warnings (o nessages	File	Line	Column	Project	
a contraction		1.05	Late 10	sector in the	1 Julyoux	

Εικόνα 1.10 Επιλογή τύπου εφαρμογής και αρχικού αντικειμένου εκκίνησης

1.3 Η κλάση αντικειμένων Console

Σε εφαρμογές τύπου ConsoleApplication είναι διαθέσιμες οι εντολές και οι ιδιότητες της κλάσης αντικειμένων Console. Η κλάση Console αντιπροσωπεύει ένα παράθυρο τερματικού στην οθόνη του Η/Υ. Με την χρήση των μεθόδων και των ιδιοτήτων της κλάσης αυτής μπορεί να γίνει η διαχείριση του παράθυρο τερματικού της εφαρμογής.

Η πρόσβαση στις μεθόδους και τις ιδιότητες αντικειμένων ή κλάσης αντικειμένων γίνεται με την εισαγωγή πρώτα του ονόματος του αντικειμένου ή της κλάσης ακολουθούμενο από μια τελεία και στην συνέχεια το όνομα της μεθόδου ή της ιδιότητας. (πχ Console.WriteLine("Wellcome"))

Οι κυριότερες ιδιότητες και μεθόδοι της κλάσης Console είναι:

- Η μέθοδος Clear() η οποία καθαρίζει τα περιεχόμενα του παραθύρου.
- Η ιδιότητα BackgroundColor η οποία ορίζει το χρώμα υπόβαθρου του παραθύρου και ιδιότητα ForegroundColor η οποία καθορίζει το χρώμα με το οποίο εκτυπώνονται οι χαρακτήρες.
- Οι ιδιότητες CursorLeft και CursorTop οι οποίες τοποθετούν τον δείκτη κειμένου (cursor) του παραθύρου σε συγκεκριμένη στήλη και γραμμή αντίστοιχα.
- Η ιδιότητα Title με την οποία ορίζεται ο τίτλος του παραθύρου.

Το παρακάτω παράδειγμα αλλάζει τον τίτλο, το χρώμα υπόβαθρου και το χρώμα των χαρακτήρων, καθαρίζει το παράθυρο, εκτυπώνει το μήνυμα "Πατήστε οποιοδήποτε πλήκτρο για τερματισμό" και περιμένει με την χρήση της μεθόδου ReadLine να πατηθεί ένας οποιοσδήποτε χαρακτήρας από το πληκτρολόγιο.

```
Sub main()
```

```
Console.Title = "Παράδειγμα 1.2"
Console.BackgroundColor = System.ConsoleColor.White
Console.ForegroundColor = System.ConsoleColor.Black
Console.Clear()
Console.CursorLeft = 30
Console.CursorTop = 10
Console.Write("Πατήστε οποιοδήποτε πλήκτρο για τερματισμό")
Console.Read()
```

End Sub

Παράδειγμα 1.2

Οι μέθοδοι Write και WriteLine

Οι μέθοδοι Write και WriteLine της κλάσης αντικειμένων Console χρησιμοποιούνται για την εκτύπωση μηνυμάτων στην οθόνη. Η διαφορά μεταξύ τους είναι ότι η WriteLine αλλάζει γραμμή κειμένου μετά την εκτύπωση του μηνύματος.

Οι μέθοδοι Write και WriteLine εκτυπώνουν σχεδόν όλους τους βασικούς τύπους δεδομένων όπως ακέραιους, πραγματικούς, λογικές μεταβλητές, χαρακτήρες και αλφαριθμητικές τιμές (Παράδειγμα 1.3)

```
Module Module1

Sub Main()

Console.WriteLine("Welcome")

Console.WriteLine(12.45)

Console.WriteLine(1)

Console.WriteLine(True)

End Sub

End Module
```

Παράδειγμα 1.3

Οι μέθοδοι αυτοί μπορούν να χρησιμοποιηθούν και για την μορφοποίηση της εκτύπωσης των δεδομένων. Εάν η σύνταξη της μεθόδου γίνει με περισσότερες από μία παραμέτρους εισόδου, τότε η πρώτη παράμετρος ορίζει την μορφή με την οποία θα εκτυπωθούν τα δεδομένα τα οποία ακολουθούν. Τα δεδομένα τα οποία θα εκτυπωθούν αριθμούνται ξεκινώντας από το αριθμό 0. Η παράμετρος μορφοποίησης είναι αλφαριθμητική τιμή η οποία χρησιμοποιεί αγκύλες για να δηλώσει την θέση στην οποία θα εκτυπωθούν τα δεδομένα.

Για παράδειγμα η εντολή

```
Console.WriteLine("{0}{1}{2}", 12, 1.32, 45)
```

ορίζει ότι θα εκτυπωθούν οι αριθμοί 12, 1.32 και 25 στις θέσεις $\{0\}$, $\{1\}$ και $\{2\}$ της παραμέτρου μορφοποίησης. Το αποτέλεσμα της εκτύπωσης της μεθόδου είναι:

121,3225

Η παράμετρος μορφοποίησης μπορεί να εμπεριέχει και κείμενο εκτός των αγκυλών που ορίζουν τα δεδομένα τα οποία θα εκτυπωθούν. Για παράδειγμα η εντολή

```
Console.WriteLine("\Delta 1=\{0\} \Delta 2=\{1\} \Delta 3=\{2\}", 12, 1.32, 45)
```

εκτυπώνει το παρακάτω μήνυμα στην οθόνη:

Δ1=12 Δ2=1,32 Δ3=45

Μπορούμε επίσης να καθορισθεί το εύρος εκτύπωσης των δεδομένων και η στοίχισή τους (δεξιά εάν το εύρος είναι θετικός αριθμός και αριστερά εάν το εύρος είναι αρνητικός αριθμός). Για παράδειγμα η εντολή

Console.WriteLine("{0,10}{1,10}{2,10}", 12, 1.32, 45)

εκτυπώνει το παρακάτω μήνυμα στην οθόνη:

12 1,32 45

Μπορεί επίσης να ορισθεί το πλήθος των ψηφίων ακεραίων αριθμών, το πλήθος των δεκαδικών ψηφίων και να εκτυπωθούν ακέραιοι σε δεκαεξαδική μορφή. Για παράδειγμα η εντολή

```
Console.WriteLine("{0:D4} {1:N5} {2 :X}", 12, 1.32, 25)
```

εκτυπώνει το παρακάτω μήνυμα στην οθόνη:

0012 1,32000 19

Τα σύμβολα μορφοποίησης D4, N5 και Χ καθορίζουν ότι:

- {0:D4} ο ακέραιος αριθμός 12 θα εκτυπωθεί με τουλάχιστον 4 ψηφία (αν τα ψηφία του αριθμού δεν επαρκούν θα προστεθούν μηδενικά ψηφία μπροστά από τον αριθμό)
- {1:N5} ο πραγματικός αριθμός 1.32 θα εκτυπωθεί με τουλάχιστον 5 δεκαδικά ψηφία (αν τα δεκαδικά ψηφία του αριθμού δεν επαρκούν θα προστεθούν μηδενικά ψηφία στο τέλος του αριθμού)
- {0:X} ο ακέραιος αριθμός 25 θα εκτυπωθεί σε δεκαεξαδική μορφή.

Οι μέθοδος ReadLine

Η μέθοδος ReadLine διαβάζει μία γραμμή κειμένου η οποία εισάγεται από το πληκτρολόγιο.

Sub Main()

```
Dim x As Double
Console.Write ("Δώστε ένα πραγματικό αριθμό")
x = Console.ReadLine()
Console.WriteLine("Ο αριθμός είναι ο {0}", x)
```

End Sub

Παράδειγμα 1.4

Το παραπάνω πρόγραμμα εκτυπώνει ένα μήνυμα το οποίο ζητά από τον χρήστη να εισάγει ένα πραγματικό αριθμό, διαβάζει τον αριθμό από το πληκτρολόγιο και τον εκτυπώνει.

Το κείμενο το οποίο εισάγεται από το πληκτρολόγιο, διαβάζεται με την μέθοδο ReadLine, μετατρέπεται αυτόματα σε πραγματικό αριθμό και αποθηκεύεται στην μεταβλητή x.

Περισσότερα παραδείγματα χρήσης της ReadLine δίνονται στην επόμενη ενότητα.

<u>Άσκηση 1.1</u>

Δημιουργήστε εφαρμογή με το όνομα Lab_1_Ask_1, γράψτε το πρόγραμμα τους παραδείγματος 1.1, αποθηκεύστε την εφαρμογή και εκτελέστε τη.

Προσθέστε την εντολή Console.Readline() πριν από το τέλος του προγράμματος. Τι παρατηρείτε;

<u>Άσκηση 1.2</u>

Δημιουργήστε εφαρμογή με το όνομα Lab_1_Ask_2, γράψτε το πρόγραμμα τους παραδείγματος 1.2 και εκτελέστε το. Τροποποιήστε το τίτλο του παραθύρου και το χρώμα υπόβαθρου και εκτελέστε ξανά την εφαρμογή.

<u>Άσκηση 1.3</u>

Δημιουργήστε εφαρμογή με το όνομα Lab_1_Ask_3, η οποία ζητά από τον χρήστη να πληκτρολογήσει ένα πραγματικό αριθμό και στην στοιχεία εκτυπώστε τον αριθμό που πληκτρολογήθηκε στην οθόνη. Κάντε χρήση του συμβόλου μορφοποίησης {0:N2} ώστε να εκτυπώσετε τον πραγματικό αριθμό με δύο δεκαδικά ψηφία.

<u>Άσκηση 1.4</u>

Δοκιμάστε να εκτελέσετε την εφαρμογή της προηγούμενης άσκησης βήμα προς βήμα. (πλήκτρο F8).

Επιλέξτε το παράθυρο Locals (Επιλογή Locals του μενού Debug - Windows). Τι εμφανίζεται στο παράθυρο αυτό;

Πηγαίνετε στην τελευταία εντολή του προγράμματος και επιλέξτε «Toggle breakpoint» από το μενού «Debug» (πλήκτρο F9). Τι παρατηρείτε να γίνεται κατά την εκτέλεση του προγράμματος;

Γράψτε στο παράθυρο «Immediate» την μέθοδο Console.WriteLine("Welcome") και εκτελέστε τη.

2 Μεταβλητές και σταθερές

2.1 Δηλώσεις Μεταβλητών

Με τις δηλώσεις των μεταβλητών ορίζονται τα ονόματα και οι τύποι των δεδομένων μιας εφαρμογής και γίνεται η δέσμευση απαραίτητης μνήμης του υπολογιστή για την αποθήκευση και την ανάκτηση των δεδομένων.

Οι δηλώσεις μεταβλητών γίνονται με την δεσμευμένη λέξη **Dim**, το όνομα της μεταβλητής, την λέξη **As** και τον τύπο των δεδομένων. Σε μία δήλωση μπορεί να ορίζονται περισσότερες από μία μεταβλητές όπως φαίνεται και στο παρακάτω παράδειγμα:

Dim i, j as Integer Dim d as Double

2.2 Τύποι Δεδομένων

Στον πίνακα 1 εμφανίζονται οι βασικότεροι τύποι μεταβλητών που υποστηρίζονται από την Visual Basic.

Τύπος	Περιγραφή	Απαιτούμενη Μυτάυτη	Εύρος Τιμών
Boolean	Δουική Μεταβλητη	2 bytes	True é False
Duto	Ακόσοιος	2 bytes	
Shute	Ακεραίος	1 byte	129.605233
Sbyte	Ακεραίος	1 byte	$-128 \cos(127)$ (signed byte)
Char	Χαρακτηρας	2 bytes	$0 \cos \zeta $ 05,555 (unsigned character)
Short	Ακεραιος	2 bytes	$-32,/68 \ \epsilon\omega\zeta \ 32,/6/$
Ushort	Ακέραιος	2 bytes	$0 \cos 65,535$ (unsigned short)
Integer	Ακέραιος	4 bytes	-2,147,483,648 έως 2,147,483,647
Uinteger	Ακέραιος	4 bytes	0 έως 4,294,967,295 (unsigned integer)
Long	Ακέραιος	8 bytes	9,223,372,036,854,775,808 έως
			9,223,372,036,854,775,807
Ulong	Ακέραιος	8 bytes	0 through 18,446,744,073,709,551,615 (unsigned long)
Decimal	Αριθμός	16 bytes	0 έως +/-79,228,162,514,264,337,593,543,950,335 για
			ακέραιους.
			0 +/-7.9228162514264337593543950335
			με δεκαδικά ψηφία
Single	Πραγματικός απλής	4 bytes	-3.4028235E+38 έως -1.401298E-45
	ακρίβειας		(αρνητικές τιμές)
			1.401298Ε-45 έως 3.4028235Ε+38
			(θετικές τιμές)
Double	Πραγματικός	8 bytes	Από -1.79769313486231570Ε+308
	διπλής ακρίβειας		έως -4.94065645841246544Ε-324
			για αρνητικούς αριθμούς
			Από 4 94065645841246544Ε-324
			$E_{\text{E}} = 1.79769313486231570E \pm 308$
			νια θετικούς αριθμούς
String	Αλωαοιθμητική	Μεταβλητή	από 0 έως 2 000 000 000 γαρακτήρες Unicode
String	μεταβλητή	moraphiltil	
Date	Ημερομηνία και	8 bytes	Από 1 Ιανουαρίου 0001 0:0:00 πμ
	ώρα		έως 31 Δεκεμβρίου 9999 11:59:59 μμ
Object	Αντικείμενο	4 bytes	Δείκτης σε αντικείμενα
Structure	Δομή	Μεταβλητή	Αναλόγως της δομής που έχει ορισθεί

Πίνακας 2.1 Βασικοί τύποι μεταβλητών

Ανάθεση τιμής

Η ανάθεση τιμής (αποθήκευση τιμής) σε μεταβλητής γίνεται με τον τελεστή "=". $\tt i=12$

d=1.34

Η ανάθεση τιμής σε μεταβλητές τύπου Date γίνεται με την χρήση των ειδικών χαρακτήρων # Dim startDate as Date startDate = $#1/5/2005 \ 16:55#$

Η ανάθεση τιμής σε μια μεταβλητή μπορεί να γίνει ταυτόχρονα με τον ορισμό της. Dim startDate as Date = #1/5/2005 16:55#

Η ανάθεση τιμής σε μια ακέραια μεταβλητή μπορεί να γίνει και με την χρήση δεκαεξαδικών ή οκταδικών αριθμών. Πρίν από την τιμή θα πρέπει να προηγηθεί η κατάλληλη ακολουθία συμβόλων: &Η για δεκαεξαδικούς και &Ο για οκταδικούς.

```
Dim num1 as Integer = &H64 'Δεκαεξαδικός αριθμός
Dim num2 as Integer = &O144 'Οκταδικόςαριθμός
```

Παράδειγμα προγράμματος το οποίο ορίζει μεταβλητές, αποθηκεύει τιμές σε μεταβλητές και τις εκτυπώνει.

```
Module Module1
    Sub Main()
        'Δηλώσεις μεταβλητών
        Dim i As Integer
        Dim 1 As Long
        Dim s As Short
        Dim b As Byte
        Dim d As Double
        Dim sn As Single
        Dim msg1 As String
        Dim c As Char
        Dim flq As Boolean
        Dim birthDay as Date
        'Ανάθεση τιμών στις μεταβλητές
        i = 12
        1 = 3456789
        s = 45
        b = 213
        d = 1.00000000045
        sn = 1.32
        msg1 = "Welcome"
        c = "e"
        flg = True
        birthDay = #1/5/1994#
        'Εκτύπωση των μεταβλητών
        Console.WriteLine(i)
        Console.WriteLine(1)
        Console.WriteLine(s)
        Console.WriteLine(b)
        Console.WriteLine(d)
        Console.WriteLine(sn)
        Console.WriteLine(msg1)
        Console.WriteLine(c)
        Console.WriteLine(flg)
        Console.WriteLine(birthDay)
    End Sub
End Module
```

Παράδειγμα 2.1

Το αποτέλεσμα της εκτέλεσης του παραπάνω προγράμματος είναι:

```
12
3456789
45
213
1,000000000045
1,32
Welcome
e
True
5/1/1994 12:00:00 πμ
```

Είσοδος δεδομένων από το πληκτρολόγιο

Η εισαγωγή δεδομένων σε εφαρμογές τύπου ConsoleApplication γίνεται με την μέθοδο ReadLine της κλάσης αντικειμένων Console.

Στο παρακάτω παράδειγμα, ορίζεται μια ακέραια μεταβλητή i , και εισάγεται η τιμή της από το πληκτρολόγιο.

```
Dim i as Integer
i =Console.ReadLine()
```

Συνήθως όταν σε μια εφαρμογή χρειάζεται να γίνει η εισαγωγή δεδομένων, εκτυπώνονται πρώτα τα κατάλληλα μηνύματα το οποία ενημερώνουν τον χρήστη για τα στοιχεία τα οποία πρέπει να εισάγει.

Για παράδειγμα, το παρακάτω πρόγραμμα εκτυπώνει το μήνυμα «Δώστε ένα ακέραιο», χρησιμοποιεί την μέθοδο ReadLine για να διαβάσει έναν ακέραιο από το πληκτρολόγιο και στην συνέχεια τον εκτυπώνει στην οθόνη.

```
Sub Main()
```

```
Dim i As Long
Console.Write ("Δώστε ένα ακέραιο")
i = Console.ReadLine()
Console.WriteLine("Ο ακέραιος είναι ο {0}", i)
```

End Sub

Παράδειγμα 2.2

Αν υποθέσουμε ότι πληκτρολογείτζαι ο αριθμός 45, τότε το αποτέλεσμα της εκτέλεσης του προγράμματος είναι το παρακάτω:

```
Δώστε ένα ακέραιο
45
Ο ακέραιος είναι ο 45
```

2.3 Μετατροπή δεδομένων

Η μετατροπή τιμών μεταξύ διαφόρων τύπων δεδομένων γίνεται είτε με την χρήση του τελεστή ανάθεσης τιμής ή με την χρήση ειδικών συναρτήσεων μετατροπής τύπων δεδομένων.

```
Module varConversions

Public Sub Main()

Dim i As Integer

Dim d As Double

d = 14.55

i = d 'Μετατροπή πραγματικού αριθμού σε ακέραιο

Console.WriteLine(i)

End Sub

End Module
```

Παράδειγμα 2.3

Στο παραπάνω παράδειγμα η μετατροπή της πραγματικής μεταβλητής σε ακέραιο (με στρογγυλοποίηση) γίνεται με τον τελεστή ανάθεσης τιμής.

Με την επιλογή «Option Strict On» η οποία μπορεί να δηλωθεί στην αρχή μιας λειτουργικής μονάδας της Visual Basic απενεργοποιούνται οι αυτόματες μετατροπές μεταξύ τύπων δεδομένων (narrowing conversions) όπου τμήμα της μετατρεπόμενης τιμής (π.χ. από πραγματικό σε ακέραιο) χάνεται ή το εύρος τιμών της μετατροπής είναι μικρότερο από το αρχικό εύρος (πχ. από πραγματικό διπλής ακρίβειας σε πραγματικό απλής ακρίβειας).

```
Option Strict On 'Περιορισμός μετατροπών
Module varConversions
Public Sub Main()
Dim i,j As Integer
Dim d As Double
Dim s As Single
d = 14.55
i = d 'Mη επιτρεπτή μετατροπή (Πραγματικός σε ακέραιο)
s = d 'Mη επιτρεπτή μετατροπή (Διπλής ακρίβειας σε απλή ακρίβεια)
j = "45" 'Mη επιτρεπτή μετατροπή (Αλφαριθμητική τιμή σε ακέραιο)
Console.WriteLine("{0} {1:N2} {2}", i, s, j)
End Sub
End Module
```

Παράδειγμα 2.4

Στο παραπάνω παράδειγμα οι μετατροπές που γίνονται με τις εντολές i=d, s=d και i="45" δεν επιτρέπονται και αντιμετωπίζονται ως συντακτικά λάθη του προγράμματος. Εάν δεν υπήρχε η δήλωση «Option Strict On», οι μετατροπές αυτές θα ήταν αποδεκτές.

Η επιλογή «Option Strict On» ή «Off» καθορίζεται καθολικά (για ολόκληρη την εφαρμογή) στις ρυθμίσεις Compile της εφαρμογής (σχήμα 2.1).

Με την επιλογή «Option Explicit Off» δεν είναι απαραίτητη η δήλωση των μεταβλητών. Το παρακάτω παράδειγμα χρησιμοποιεί την μεταβλητή s χωρίς να δηλώσει τον τύπο της μεταβλητής.

```
Option Strict Off
Option Explicit Off
Module varConversions
Public Sub Main()
s = "45"
Console.WriteLine(s)
End Sub
End Module
```

Παράδειγμα 2.5

ſ						
Application	Build output path:				_	Iab_1_ask_2 My Project
Compile	bin\Release\			Browse		- 🎽 firstProgram.vb
Debug						🦢 🐚 Module1.vb
	Option explicit: Option strict:	Option	n c <u>o</u> mpare:		1	
References		L Binary	1]	
Resources	Condition		Notif	ication	a	
Settings	Implicit conversion		None		3	
	Implicit type: object assumed		None		-	
Signing	Lise of variable prior to assignment		Warr	ina 💌		
Security	Euction/Operator without return value		Warr	ing 💻		
Dublich	Unused local variable		Warr	ing 💌		
Publish	Instance variable accesses shared member		Warn	ing 👻	1	
	Recursive operator or property access		Warr	ing 💆	1	Properties
	Duplicate or overlapping catch blocks		Warn	ing 🔄	1	
					_	
	Uşable all warnings					
	I Treat all warnings as errors					
rror List					→ 1	×
	0 Warnings (1) 0 Messages					
🔕 3 Errors 🚺		File	Line	Column	Project	
3 Errors					12/2/	

Εικόνα 2.1 Επιλογή ρυθμίσεων Option Explicit και Option Strict

Συναρτήσεις μετατροπής μεταβλητών

Στον πίνακα 2.1 φαίνονται τα ονόματα συναρτήσεων για την μετατροπή των κυριοτέρων τύπων μεταβλητών.

Όνομα συνάρτησης	Τύπος μεταβλητής ο οποίος επιστρέφεται
Cbool	Boolean
Cbyte	Byte
Cchar	Char
Cdate	Date
CDbl	Double
CDec	Decimal
CInt	Integer
CLng	Long
СОbj	Object
CSByte	SByte
CShort	Short
CSng	Single
CStr	String
CUInt	UInteger
CULng	ULong
CUShort	UShort

Πίνακας 2.2 Συναρτήσεις μετατροπής δεδομένων

Στο παρακάτω παράδειγμα γίνεται η μετατροπή τιμών με την χρήση των συναρτήσεων Cint και CSng.

```
Option Strict On 'Περιορισμός μετατροπών
Module varConversions
Public Sub Main()
Dim i, j As Integer
Dim d As Double
Dim s As Single
d = 14.55
i = CInt(d) '(Πραγματικός σε ακέραιο)
s = CSng(d) '(Διπλής ακρίβειας σε απλή ακρίβεια)
j = CInt("45") '(Αλφαριθμητική τιμή σε ακέραιο)
Console.WriteLine("{0} {1:N2} {2}", i, s, j)
End Sub
End Module
```

Παράδειγμα 2.6

Η συνάρτηση Ctype

Εναλλακτικά αντί των συναρτήσεων του πίνακα 2.2 μπορεί να χρησιμοποιηθεί η συνάρτηση Ctype η οποία δέχεται ως πρώτη παράμετρο τον τιμή η οποία θα μετατραπεί και ως δεύτερη παράμετρο τον τύπο δεδομένων όπου θα γίνει η μετατροπή

```
Public Sub Main()
Dim i As Integer
Dim d As Date
Dim s As Single
i = CType("456", Integer)
s = CType("4,434", Single)
d = CType("1/1/2005", Date)
Console.WriteLine("{0} {1:N2} {2}", i, s, d)
End Sub
```

Παράδειγμα 2.7

Η μέθοδος Parse

Όλοι οι βασικοί τύποι δεδομένων (εκτός του αλφαριθμητικού τύπου String) διαθέτουν την μέθοδο parse η οποία μετατρέπει μια αλφαριθμητική τιμή στον αντίστοιχο τύπο δεδομένων. Στο παρακάτω παράδειγμα φαίνεται ή σύνταξη της μεθόδου Parse.

```
Public Sub Main()
   Dim i As Integer
   Dim d As Date
   Dim s As Single
   i = Integer.Parse("456")
   s = Single.Parse("4,434")
   d = Date.Parse("1/1/2005")
   Console.WriteLine("{0} {1:N2} {2}", i, s, d)
End Sub
```



Η μέθοδος ToString

Όλοι οι βασικοί τύποι δεδομένων (εκτός του αλφαριθμητικού τύπου String) διαθέτουν επίσης την μέθοδο ToString η οποία μετατρέπει την τιμή της μεταβλητής σε αλφαριθμητική τιμή (αντίστροφος της μεθόδου Parse). Η μέθοδος αυτή μπορεί προαιρετικά να πάρει ως παράμετρο όρισμα μορφοποίησης (παρόμοιο της μεθόδου WriteLine)

```
Public Sub Main()
    Dim i As Integer
    Dim d As Date
    Dim s As Single
    i = 456
    s = 4.434
    d = #25/2/2005#
    'Μετατροπή ακέραιου χωρίς παράμετρο μορφοποίησης
    Console.WriteLine("i={0,30}", i.ToString())
    Μετατροπή ακέραιου με εύρος 9 ψηφίων
    Console.WriteLine("i={0,30}", i.ToString("D9"))
    Μετατροπή πραγματικού με 4 δεκαδικά ψηφία
    Console.WriteLine("s={0,30}", s.ToString("N4"))
    'Μετατροπή ημερομηνίας σε πλήρως ανεπτυγμένη μορφή
    Console.WriteLine("d={0,30}", d.ToString("D"))
End Sub
```

Παράδειγμα 2.9

Για την μετατροπή ημερομηνιών μπορούν να χρησιμοποιηθούν προκαθορισμένες ή προσαρμοσμένες μορφές ημερομηνίας. Το παρακάτω πρόγραμμα εκτυπώνει την τρέχουσα ημέρα και ώρα στις κυριότερες προκαθορισμένες μορφές καθώς και σε 4 προσαρμοσμένες μορφές. Οι προσαρμοσμένες μορφές ημερομηνιών ορίζονται με την χρήση των συμβόλων:

- d, dd, ddd, dddd για την ημέρα (πχ 2, 02, Κυρ, Κυριακή)
- Μ, ΜΜ, ΜΜΜ, ΜΜΜΜΜ για τους μήνες (πχ 2, 02, Φεβ, Φεβρουάριος)
- yy, yyyy για το έτος (πχ 07, 2007)
- h, hh για τις ώρες (πχ 6, 06)
- m, mm για τα λεπτά (πχ 5, 05)
- s, ss για τα δευτερόλεπτα (πχ 6, 06)

End Sub

Παράδειγμα 2.10

Το αποτέλεσμα της εκτέλεσης του προγράμματος είναι:

```
Προκαθορισμένες μορφές ημερομηνίαςthe patternD = Kυριακή, 25 Φεβρουαρίου 2007Short Date Pattern (short)d = 25/2/2007Full Date Pattern (short)f = Skuριακή, 25 Φεβρουαρίου 2007 1:47:17 μμFull Date Pattern (short)f = Kυριακή, 25 Φεβρουαρίου 2007 1:47:17 μμGeneral Date Pattern (short)g = 25/2/2007 1:47:17 μμGeneral Date Pattern (short)f = 1:47:17 μμShort Time Pattern for t = 1:47:17 μμDogapuoguéveç μορφές ημερομηνίαςthe dd/MM/yyy h:mm:ss = 25/02/007 01:47:17dd/MM/yyyy h:mm:ss = 25/02/007 01:47:17dd/MM/yyyy h:mm:ss = 25-6εβ-2007 01:47:17dddd, dd MMMMM yyyy h:mm:ss = Kuριακή, 25 Φεβρουαρίου 2007 1:47:17
```

2.4 Σταθερές

Οι δηλώσεις σταθερών σε ένα πρόγραμμα χρησιμοποιούνται για να ορίσουν αριθμούς ή άλλες τιμές δεδομένων οι οποίες παραμένουν σταθερές και αμετάβλητες κατά την διάρκεια εκτέλεσης ενός προγράμματος.

Η δήλωση σταθερών γίνεται με την δεσμευμένη λέξη Const, το όνομα της σταθερής, τον τελεστή ισότητας και την τιμή της σταθερής

```
Const DaysInYear = 365
Const WorkDays = 250
conPi = 3.14159265358979
Const conMaxPlanets As Integer = 9
Const conReleaseDate = #1/1/1995#
```

Προαιρετικά μπορούμε να δηλωθεί και ο τύπος της σταθερής

```
Const conMsg as String = "Hello"
```

2.5 Ασκήσεις κεφαλαίου 2

<u>Άσκηση 2.1</u>

Δημιουργήστε εφαρμογή με όνομα Lab_2_Ask_1 η οποία ζητά από τον χρήστη να πληκτρολογήσει το όνομα, το επώνυμο, την ηλικία, το ύψος και την ημερομηνία γέννησης του. Στην συνέχεια η εφαρμογή εκτυπώνει τα στοιχεία του χρήστη στην οθόνη. Το ύψος εκτυπώνεται με δύο δεκαδικά στοιχεία. Η ημερομηνία εκτυπώνεται με την χρήση των προκαθορισμένων μορφών ημερομηνίας (όπως στο παράδειγμα 2.10)

Ορίστε σημείο διακοπής («Toggle breakpoint» του μενού Debug) στην πρώτη εντολή εκτύπωσης των στοιχείων. Χρησιμοποιήστε το παράθυρο Locals για να δείτε τις τιμές των μεταβλητών πριν την εκτύπωση και το παράθυρο Immediate για να αλλάξετε τις τιμές των δεδομένων.

<u>Άσκηση 2.2</u>

Δημιουργήστε εφαρμογή με όνομα Lab_2_Ask_2 η οποία ζητά από τον χρήστη να εισάγει 6 πραγματικούς αριθμούς και στην συνέχεια τους εκτυπώνει σε 3 γραμμές (δύο αριθμοί σε κάθε γραμμή). Οι αριθμοί εκτυπώνονται με ακρίβεια 4 δεκαδικών στοιχείων και εύρος εκτύπωσης 10 χαρακτήρες.

Χρησιμοποιήστε ως παράμετρο μορφοποίησης: {0,10:N4} {1,10:N4}

<u>Άσκηση 2.3</u>

Δημιουργήστε εφαρμογή με όνομα Lab_2_Ask_3 η οποία ζητά από τον χρήστη να εισάγει μια ημερομηνία και στην συνέχεια εκτυπώνει την ημέρα της εβδομάδος (Κυριακή, Δευτέρα κλπ) η οποία αντιστοιχεί στην ημερομηνία αυτή.

<u>Άσκηση 2.4</u>

Δημιουργήστε εφαρμογή με όνομα Lab_2_Ask_4 η οποία ζητά από τον χρήστη να εισάγει δεδομένα, κάνει χρήση των συναρτήσεων μετατροπής δεδομένων του πίνακα 2.2, της συνάρτησης ctype, και των μεθόδων parse και ToString και εκτυπώνει τα αποτελέσματα των μετατροπών.

3 Τελεστές – Μαθηματικές συναρτήσεις

Η Visual Basic υποστηρίζει τους παρακάτω τύπους τελεστών:

- Αριθμητικοί τελεστές
- Αλφαριθμητικοί τελεστές
- Τελεστές σύγκρισης
- Λογικοί τελεστές

3.1 Αριθμητικοί τελεστές

Τελεστής	Περιγραφή	Παράδειγμα	Αποτέλεσμα
٨	Δύναμη	3 ^ 2	9
-	Αρνητικό πρόσημο	-5	5
*	Πολλαπλασιασμός	2 * 3	6
/	Διαίρεση	5 / 2	2.5
\	Ακέραια διαίρεση	$7 \setminus 3$	2
Mod	Υπόλοιπο διαίρεσης	7 Mod 3	1
+	Πρόσθεση	2+4	6
-	Αφαίρεση	4-2	2

Πίνακας 3.1 Αριθμητικοί τελεστές

Παράδειγμα χρήσης αριθμητικών τελεστών με πραγματικούς αριθμούς για τον υπολογισμό της παράστασης x 3 + 2 x y + 4 y + 24:

```
Public Sub main()
Dim x, y, z As Double
Console.Write("Δώσε ένα πραγματικό αριθμό x ")
x = Console.ReadLine()
Console.Write ("Δώσε ένα δεύτερο πραγματικό αριθμό y ")
y = Console.ReadLine()
z = x ^ 3 + 2 * x * y + 4 * y + 24
Console.WriteLine(" x ^ 3 + 2 * x * y + 4 * y + 24 = {0:N4}", z)
End Sub
```

Παράδειγμα 3.1

Το αποτέλεσμα του παραπάνω προγράμματος εάν εισαχθούν οι αριθμοί 1.45 και 3,321 είναι:

```
Δώσε ένα πραγματικό αριθμό x 1,45
Δώσε ένα δεύτερο πραγματικό αριθμό 3,321
x ^{3} + 2 * x * y + 4 * y + 24 = 49,9635
```

Παράδειγμα χρήσης αριθμητικών τελεστών με ακέραιους αριθμούς:

```
Public Sub main()
Dim i, j As Integer
Console.Write ("Δώσε ένα ακέραιο ")
i = Console.ReadLine()
Console.Write ("Δώσε ένα δεύτερο ακέραιο ")
j = Console.ReadLine()
Console.WriteLine(" {0} + {1} = {2}", i, j, i + j)
Console.WriteLine(" {0} - {1} = {2}", i, j, i - j)
Console.WriteLine(" {0} * {1} = {2}", i, j, i * j)
Console.WriteLine(" {0} / {1} = {2}", i, j, i * j)
Console.WriteLine(" {0} / {1} = {2}", i, j, i / j)
Console.WriteLine(" {0} ^ {1} = {2}", i, j, i ^ j)
Console.WriteLine(" {0} ^ {1} = {2}", i, j, i ^ j)
Console.WriteLine(" {0} ^ {1} = {2}", i, j, i ^ j)
Console.WriteLine(" {0} \ {1} = {2}", i, j, i ^ j)
Console.WriteLine(" {0} Mod {1} = {2}", i, j, i Mod j)
End Sub
```

Παράδειγμα 3.2

Το αποτέλεσμα του παραπάνω προγράμματος εάν εισαχθούν οι αριθμοί 3 και 2 είναι:

```
Δώσε ένα ακέραιο 3

Δώσε ένα δεύτερο ακέραιο 2

3 + 2 = 5

3 - 2 = 1

3 * 2 = 6

3 / 2 = 1,5000

3 ^ 2 = 9

3 \setminus 2 = 1

3 \mod 2 = 1
```

Με την χρήση του υπολοίπου και της διαίρεση ακεραίων μπορεί να υπολογισθούν τα ψηφία ενός αριθμού.

```
Public Sub main()
    Dim i, j, k As Integer
    Console.Write(" Δώστε ένα πενταψήφιο ακέραιο ")
    i = Console.ReadLine()
    j = i Mod 10^4
    Console.WriteLine(" Τα 4 τελευταία ψηφία είναι {0}", j)
    k = j \ 10^3
    Console.WriteLine(" Τα τέταρτο ψηφίο είναι {0}", k)
End Sub
```

```
Παράδειγμα 3.3
```

Εάν πληκτρολογηθεί ο αριθμός 12345 το αποτέλεσμα της εκτέλεσης του προγράμματος είναι:

```
Δώστε ένα πενταψήφιο ακέραιο 12345
Τα 4 τελευταία ψηφία είναι 2345
Τα τέταρτο ψηφίο είναι 2
```

Στο παραπάνω παράδειγμα το υπόλοιπο της διαίρεσης ενός αριθμού με τον αριθμό 10000 υπολογίζει τα τελευταία 4 ψηφία του αριθμού και με την ακέραια διαίρεση των 4 τελευταίων με το αριθμό 1000 υπολογίζεται το τέταρτο ψηφίο του αριθμού.

3.2 Αλφαριθμητικοί τελεστές

Με την χρήση των τελεστών & ή + μπορούμε να ενώσουμε αλφαριθμητικές μεταβλητές. Στο παρακάτω παράδειγμα ο τελεστής & χρησιμοποιείται για τον σχηματισμό του ονοματεπώνυμου από το όνομα και το επώνυμο και ο τελεστής + χρησιμοποιείται για να ενώσει δύο αλφαριθμητικές τιμές και να τις εκτυπώσει.

```
Public Sub main()
Dim onoma, eponymo, onomateponymo As String
Console.Write("Δώστε το όνομα σας ")
onoma = Console.ReadLine()
Console.Write("Δώστε το επώνυμο σας ")
eponymo = Console.ReadLine()
'Χρήση τελεστή & για τον υπολογισμό ονοματεπώνυμο
onomateponymo = onoma & " " & eponymo
'Χρήση τελεστή + για τον υπολογισμό ονοματεπώνυμο
Console.WriteLine("Το ονοματεπώνυμο σας είναι " + onomateponymo)
End Sub
```

Παράδειγμα 3.4

```
Το αποτέλεσμα της εκτέλεσης του προγράμματος είναι:
Δώστε το όνομα σας Ιωάννης
Δώστε το επώνυμο σας Γεωργίου
Το ονοματεπώνυμο σας είναι Ιωάννης Γεωργίου
```

2 2		
5.5	Τελεστες συνκοισης	

Τελεστής	Περιγραφή	Παράδειγμα	Αποτέλεσμα
=	Ίσο	5 = 7	False
\diamond	Διάφορο	5 <> 7	True
<	Μικρότερο	5 < 7	True
<=	Μικρότερο ή ίσο	5 <= 5	True
>	Μεγαλύτερο	5 > 7	False
>=	Μεγαλύτερο ή ίσο	7>=7	True
Is	Είναι το ίδιο αντικείμενο	A Is B	
IsNot	Δεν είναι το ίδιο αντικείμενο	A IsNot B	
TypeOfIs	Ο τύπος του αντικειμένου είναι	TypeOf(A) is B	
Like	Η αλφαριθμητική τιμή μοιάζει με	"En*" Like "Ena"	True

Πίνακας 3.2 Τελεστές σύγκρισης

Παράδειγμα αριθμητικών συγκρίσεων μεταξύ δύο αριθμών που εισάγονται από τον χρήστη:

```
Public Sub main()
    Dim i, j As Integer
    Console.Write("Δώσε ένα ακέραιο ")
    i = Console.ReadLine()
    Console.Write("Δώσε ένα δεύτερο ακέραιο ")
    j = Console.ReadLine()
    Console.WriteLine("{0} = {1} = {2}", i, j, i = j)
    Console.WriteLine("{0} <> {1} = {2}", i, j, i <> j)
    Console.WriteLine("{0} > {1} = {2}", i, j, i > j)
    Console.WriteLine("{0} > {1} = {2}", i, j, i > j)
    Console.WriteLine("{0} << {1} = {2}", i, j, i < j)
    Console.WriteLine("{0} << {1} = {2}", i, j, i < j)
    Console.WriteLine("{0} < {1} = {2}", i, j, i < j)
    Console.WriteLine("{0} < {1} = {2}", i, j, i < j)
    Console.WriteLine("{0} < {1} = {2}", i, j, i < j)
    Console.WriteLine("{0} << {1} = {2}", i, j, i < j)
    Console.WriteLine("{0} << {1} = {2}", i, j, i <= j)
    Console.WriteLine("{0} << {1} = {2}", i, j, i <= j)
    Console.WriteLine("{0} << {1} = {2}", i, j, i <= j)
    Console.WriteLine("{0} << {1} = {2}", i, j, i <= j)
    Console.WriteLine("{0} << {1} = {2}", i, j, i <= j)
    Console.WriteLine("{0} << {1} = {2}", i, j, i <= j)
    Console.WriteLine("{0} << {1} = {2}", i, j, i <= j)
    Console.WriteLine("{0} << {1} = {2}", i, j, i <= j)
    Console.WriteLine("{0} << {1} = {2}", i, j, i <= j)
    Console.WriteLine("{0} << {1} = {2}", i, j, i <= j)
    Console.WriteLine("{0} << {1} = {2}", i, j, i <= j)
    Console.WriteLine("{0} << {1} = {2}", i, j, i <= j)
    Console.WriteLine("{0} << {1} = {2}", i, j, i <= j)
    Console.WriteLine("{0} << {1} = {2}", i, j, i <= j)
    Console.WriteLine("{0} << {1} = {2}", i, j, i <= j)
    Console.WriteLine("{0} << {1} = {2}", i, j, i <= j)
    Console.WriteLine("{0} << {1} = {2}", i, j, i <= j)
    Console.WriteLine("{0} << {1} = {2}", i, j, i <= j)
    Console.WriteLine("{0} << {1} = {2}", i, j, i <= j)
    Console.WriteLine("{0} <= {1} = {2}", i, j, i <= j)
    Console.WriteLine("{0} <= {1} = {2}", i, j, i <= j)
    Console.WriteLine("{0} <= {1} = {2}", i, j, i <= j)
    Console.WriteLine("{0} <= {1} = {2}", i, j, i <= j)
    Console.WriteLine("{0
```

Παράδειγμα 3.5

Εάν εισαχθούν οι αριθμοί 5 και 7 τα αποτελέσματα των συγκρίσεων είναι:

```
  \Delta \omega \sigma \varepsilon 

  ένα ακέραιο 5

  \Delta \omega \sigma \varepsilon 

  ένα δεύτερο ακέραιο 7

5 = 7 = False

5 <> 7 = True

5 > 7 = False

5 >= 7 = False

5 < 7 = True

5 < 7 = True

5 < 7 = True
```

Οι τελεστές σύγκρισης μπορούν να χρησιμοποιηθούν και για την σύγκριση αλφαριθμητικών μεταβλητών ή χαρακτήρων.

Με την οδηγία Option Compare Text στην αρχή της λειτουργικής μονάδος, η σύγκριση γίνεται με βάση της ρυθμίσεις της επιλεγμένης γλώσσας των Windows. Με την επιλογή αυτή οι πεζοί χαρακτήρες θεωρούνται και ταξινομούνται ως ίσοι με τους αντίστοιχους κεφαλαίους χαρακτήρες. Με την οδηγία Option Compare Binary η σύγκριση γίνεται με βάση των κωδικό (αριθμό) που αντιστοιχεί σε κάθε χαρακτήρα. Με την επιλογή Option Compare Binary οι πεζοί χαρακτήρες δεν είναι ισοδύναμοι με τους κεφαλαίους χαρακτήρες. Η επιλογή του τρόπου σύγκρισης των αλφαριθμητικών τιμών μπορεί να γίνει για όλο το πρόγραμμα με την φόρμα ρυθμίσεων της εφαρμογής (όπως και οι επιλογές Option Explicit και Option Strict) Το παρακάτω πρόγραμμα με την επιλογή Option Compare Text εκτυπώνει ως αποτέλεσμα της σύγκρισης true («Welcome» = «welcome»). Το αποτέλεσμα της σύγκρισης θα ήταν false εάν είχαμε επιλέξει Option Compare Binary.

```
Option Compare Text
Module stringComparisonç
    Public Sub main()
        Console.WriteLine("WELCOME" >= "welcome")
        End Sub
End Module
```

Παράδειγμα 3.6

Οι τελεστές Is, IsNot και TypeOf .. Is χρησιμοποιούνται για τις συγκρίσεις αντικειμένων. Παραδείγματα τέτοιων συγκρίσεων δίνονται σε επόμενο κεφάλαιο.

Με τον τελεστή Like μπορούμε να συγκρίνουμε εάν κάποια αλφαριθμητική τιμή μοιάζει ή εμπεριέχει κάποια άλλη. Στις συγκρίσεις μπορούμε να χρησιμοποιήσουμε τους ειδικούς χαρακτήρες ?, *, # ή να χρησιμοποιήσουμε σύνολα η περιοχές χαρακτήρων. Με τον χαρακτήρα ? η σύγκριση αληθεύει για οποιονδήποτε χαρακτήρα υπάρχει στη θέση αυτή. Με τον χαρακτήρα # η σύγκριση αληθεύει για οποιονδήποτε αριθμητικό χαρακτήρα (χαρακτήρες από '0' έως '9') υπάρχει στη θέση αυτή. Με τον χαρακτήρων υπάρχει στη θέση αυτή. Με τον χαρακτήρα * η σύγκριση αληθεύει για οποιονδήποτε ακολουθία χαρακτήρων υπάρχει στη θέση αυτή. Με τον χαρακτήρα * η σύγκριση αληθεύει για οποιονδήποτε ακολουθία χαρακτήρων υπάρχει στη θέση αυτή. Με τον χαρακτήρα * η σύγκριση αληθεύει για οποιαδήποτε ακολουθία χαρακτήρων υπάρχει στη θέση αυτή. Αντί του χαρακτήρα ? μπορεί να χρησιμοποιηθεί ένα σύνολο εναλλακτικών χαρακτήρων (πχ [αο]σημαίνει ότι στην θέση αυτή μπορεί να υπάρχουν ή χαρακτήρες 'a' ή 'o'). Αντί του χαρακτήρων (πχ [αο]σημαίνει ότι στην θέση αυτή μπορεί να υπάρχουν ή χαρακτήρων (πχ [a-z] σημαίνει ότι στη θέση αυτή μπορεί να αποκλείσουμε ένα σύνολο ή μια περιοχή χαρακτήρες του λατινικού αλφαβήτου). Μπορούμε επίσης να αποκλείσουμε ένα σύνολο ή μπορεί να υπάρχει οποιοσδήποτε χαρακτήρα γαρακτήρα ! (πχ [!αο] σημαίνει ότι στην θέση αυτή μπορεί να 'γο')

Παράδειγμα χρήσης του τελεστή like δίνεται στο παρακάτω πρόγραμμα. Το πρόγραμμα ζητά την εισαγωγή ενός ονόματος και στην συνέχεια με την χρήση του τελεστή like κάνει διάφορες συγκρίσεις για το όνομα.

```
Public Sub main()
   Dim s As String
   Console.Write("Δώστε ένα όνομα ")
   s = Console.ReadLine()
    'Είναι οι συγκρινόμενες τιμές είναι ίδιες;
   Console.WriteLine("Το όνομα " & s & " είναι Ηλίας {0}", s Like "Ηλίας")
    'Είναι παρόμοιο με 'Ηλ??α
    Console.WriteLine("To óvoµa " &
         s & " είναι παρόμοιο με 'Ηλ??ς'{0}", s Like "Ηλ??ς")
    'Αρχίζει με την λέξη "Ηλ
   Console.WriteLine("Το όνομα " & s & " αρχίζει από 'Ηλ' {0}", s Like "Ηλ*")
    Έμπεριέχει την λέξη "λία"
   Console.WriteLine("Το όνομα " &
         s & " περιέχει την λέξη 'λία' {0}", s Like "*λία*")
    'Τελειώνει σε τον χαρακτήρα
   Console.WriteLine("Το όνομα " & s & " τελειώνει με 'ς' {0}", s Like "*ς")
    'Η κατάληξη είναι "ας" ή "ος'
   Console.WriteLine("To óvoµa " &
         s & " έχει κατάληξη 'ας' ή 'ος' {0}", s Like "*[αο]ς")
    'Αργίζει με κεφαλαίο φωνήεν
    Console.WriteLine("To όνομα " &
         s & " αρχίζει με κεφαλαίο φωνήεν {0}", s Like "[ΑΕΗΙΥΟΩ]*")
    'Περιέχει τους αριθμητικούς χαρακτήρες 0 έως 9;
    Console.WriteLine("Το όνομα " & s & " περιέχει αριθμούς {0}", s Like "*#*")
    'Δεν περιέχει τους αριθμητικούς χαρακτήρες Ο και
    Console.WriteLine("To óvoµa " &
         s & "δεν περιέχει Ο καί 1 {0}", s Like "*[!01]*")
End Sub
```

Παράδειγμα 3.7

Το αποτέλεσμα της εκτέλεσης του προγράμματος αν δοθεί το όνομα «Ηλίας» ως είσοδος είναι το παρακάτω

Το όνομα Ηλίας είναι Ηλίας True Το όνομα Ηλίας είναι παρόμοιο με 'Ηλ??ς'True Το όνομα Ηλίας αρχίζει από 'Ηλ' True Το όνομα Ηλίας περιέχει την λέξη 'λία' True Το όνομα Ηλίας τελειώνει με 'ς' True Το όνομα Ηλίας έχει κατάληξη 'ας' ή 'ος' True Το όνομα Ηλίας αρχίζει με κεφαλαίο φωνήεν True Το όνομα Ηλίας περιέχει αριθμούς False Το όνομα Ηλίας δεν περιέχει 0 και 1 True

3.4 Λογικοί τελεστές

Τελεστής	Περιγραφή	Παράδειγμα	Αποτέλεσμα
Not	Λογική άρνηση	Not A	False εάν A = True
			True εάν A = False
And	Λογική σύζευξη	A And B	True εάν Α και Β True
Or	Λογική διάζευξη	A or B	True εάν A ή B True
Xor	Αποκλειστική διάζευξη	A Xor B	True εάν
			A True και B False ή
			A False και B True
AndAlso	Λογική σύζευξη	A AndAlso B	True εάν Α και Β True
OrAlso	Λογική διάζευξη	A orAlso B	True εάν A ή B True

Πίνακας 3.3 Λογικοί Τελεστές

Ο λογικός τελεστής AndAlso είναι παρόμοιος με τον τελεστή And όπως και ο τελεστής OrAlso με τον τελεστή Or. Η διαφορά των τελεστών AndAlso και OrAlso από τους τελεστές And και Or βρίσκεται στον τρόπο με τον οποίο υπολογίζουν το αποτέλεσμα της λογικής πράξης. Οι τελεστές AndAlso και OrAlso εξετάζουν την πρώτη λογική τιμή και μόνον εάν χρειάζεται εξετάζουν την δεύτερη λογική τιμή και μόνον εάν χρειάζεται εξετάζουν την δεύτερη λογική τιμή και μόνον εάν χρειάζεται εξετάζουν την δεύτερη λογική τιμή και κάνουν την λογική πράξη (π.χ. στην λογική σύζευξη A AndAlso B εάν A=False επιστρέφεται False χωρίς να υπολογισθεί η τιμή του B). Οι τελεστές AndAlso και OrAlso είναι χρήσιμοι σε περιπτώσεις όπου ο υπολογισμός της δεύτερης λογικής τιμής απαιτεί σημαντικό χρόνο (π.χ. εάν η τιμή αυτή επιστρέφεται από μία συνάρτηση η οποία χρειάζεται σημαντικό χρόνο για την εκτέλεση της).

Οι λογικοί τελεστές χρησιμοποιούνται για τον σχηματισμό πολύπλοκων λογικών εκφράσεων και συγκρίσεων.

Παράδειγμα (x >= 0) And (x <=10) Εξετάζει εάν το x ανήκει στο διάστημα 0..10 (x=1) or (x=3) Εξετάζει εάν το x είναι ίσο με 1 ή 3

3.5 Τελεστές ανάθεσης τιμής

Εκτός του τελεστή ανάθεσης τιμής = μπορεί να χρησιμοποιηθούν και οι παρακάτω τελεστές

Τελεστής	Παράδειγμα	Ισοδύναμο με
=	x=2	
^=	x^=2	$x=x^2$
+=	x+=1	x=x+1
-=	x-=1	x=x-1
=	x=3	x=x*3
/=	x/= 3	x=x/3
\=	x\= 3	x=x\3
&=	x&="test"	x=x&"test"
	(χ αλφαριθμητική μεταβλητή)	

Πίνακας 3.4 Τελεστές ανάθεσης τιμής

3.6 Χρήση παρενθέσεων - Προτεραιότητα τελεστών

Οι παρενθέσεις χρησιμοποιούνται για τον καθορισμό της προτεραιότητας των πράξεων σε πολύπλοκες αριθμητικές ή λογικές παραστάσεις.

Οι πράξεις σε αριθμητικές ή λογικές παραστάσεις εκτελούνται σύμφωνα με την προτεραιότητα των τελεστών η οποία φαίνεται στον παρακάτω πίνακα. Οι πράξεις για τελεστές που έχουν την ίδια προτεραιότητα γίνονται από τα αριστερά προς τα δεξιά.

Τελεστής	Περιγραφή
	Παρένθεση
^	Ύψωση σε δύναμη
-	Αρνητικό πρόσημο
* /	Πολλαπλασιασμός - Διαίρεση
\	Ακέραια διαίρεση
Mod	Υπόλοιπο διαίρεσης
+-	Πρόσθεση, αφαίρεση
&	Αλφαριθμητική ένωση
=, <>, <, <=, >, >=, Like, Is,	Τελεστές σύγκρισης
IsNot, TypeOfIs	
Not	Λογική άρνηση
And, AndAlso	Λογική σύζευξη
Xor, Or, OrAlso	Λογική και αποκλειστική διάζευξη

Πίνακας 3.5 Προτεραιότητας Τελεστών

3.7 Μαθηματικές συναρτήσεις

Η κλάση αντικειμένων Math παρέχει χρήσιμες μαθηματικές συναρτήσεις. Οι κυριότερες από αυτές φαίνονται στον παρακάτω πίνακα.

Όνομα συνάρτησης	Περιγραφή	Παράδειγμα
Abs(x)	Επιστρέφει την απόλυτη τιμή του χ	Abs(23.7) = 23.7
		Abs(0) = 0
		Abs(-23.7) = 23.7
Ceiling(x)	Στρογγυλοποιεί τον αριθμό x στον	Ceiling(9.2) = 10.0
	μικρότερο κοντινό ακέραιο	Ceiling(-9.8) is -9.0
Cos(x)	Επιστρέφει το συνημίτονο του x (x σε	$\cos(0.0) = 1.0$
	ακτίνια)	
Exp(x)	e ^x	$\operatorname{Exp}(1.0) = 2.71828182845905$
		Exp(2.0) = 7.38905609893065
Floor(x)	Στρογγυλοποιεί τον αριθμό x στον	Floor(9.2) = 9.0
	μεγαλύτερο κοντινό ακέραιο	Floor(-9.8) = -10.0
Log(x)	Λογάριθμος του x	Log(2.7182818284590451) = 1.0
		Log(7.3890560989306504) = 2.0
Max(x, y)	Επιστρέφει την μεγαλύτερη τιμή των x	Max(2.3, 12.7) = 12.7
	και γ	Max(-2.3, -12.7) = -2.3
Min(x, y)	Επιστρέφει την μικρότερη τιμή των χ	Min(2.3, 12.7) = 2.3
	και γ	Min(-2.3, -12.7) = -12.7
Pow(x, y)	Επιστρέφει την y δύναμη του x	Pow(2.0, 7.0) = 128.0
		Pow(9.0, .5) = 3.0
Sin(x)	Επιστρέφει το ημίτονο του x (x σε	Sin(0.0) = 0.0
	ακτίνια)	
Sqrt(x)	Επιστρέφει την τετραγωνική ρίζα του x	Sqrt(9.0) = 3.0
		Sqrt(2.0) = 1.4142135623731
Tan(x)	Επιστρέφει την εφαπτομένη του x (x σε	Tan(0.0)=0.0
	ακτίνια)	

Πίνακας 3.6 Μαθηματικές συναρτήσεις

Η κλάση αντικειμένων Math ορίζει επίσης τις σταθερές:

- PI Math.PI = 3.1415926535897931
- E Math.E = 2.7182818284590451

Για την χρήση των συναρτήσεων και των σταθερών αυτών σε αριθμητικές παραστάσεις πρέπει να προηγηθεί πρώτα το όνομα της κλάσης Math ακολουθούμενο από τελεία και το όνομα της συνάρτησης ή της σταθερής.

Το παρακάτω παράδειγμα χρησιμοποιεί μαθηματικές συναρτήσεις για την μετατροπή γωνίας σε ακτίνια και τον υπολογισμό του ημίτονου και του συνημίτονου της γωνίας.

```
Public Sub main()
    Dim x As Double
    Dim xRad As Double 'Γωνία σε ακτίνια
    Dim sinX As Double 'Ημίτονο της γωνίας
    Dim cosX As Double 'Συνημίτονο της γωνίας
    Console.Write ("Δώστε μία γωνία σε μοίρες ")
    x = Console.ReadLine()
    xRad = Math.PI * x / 180 'Μετατροπή γωνίας σε ακτίνια
    sinX = Math.Sin(xRad) 'Υπολογισμός ημίτονου
    cosX = Math.Cos(xRad) 'Υπολογισμός συνημίτονου
    Console.WriteLine(
       "Η γωνία {0:N4} μοιρών είναι {1:N4} ακτίνια ", x, xRad)
    Console.WriteLine(
      "Το ημίτονο της γωνίας {0:Ν4} είναι
                                              {1:N4} ", x, sinX)
    Console.WriteLine(
      "Το συνημίτονο της γωνίας {0:N4} είναι {1:N4} ", x, CosX)
End Sub
```

Παράδειγμα 3.8

Το αποτέλεσμα της εκτέλεσης του παραπάνω προγράμματος εάν δοθεί γωνία 60 μοιρών είναι:

Δώστε μία γωνία σε μοίρες 60 Η γωνία 60,0000 μοιρών είναι 1,0472 ακτίνια Το ημίτονο της γωνίας 60,0000 είναι 0,8660 Το συνημίτονο της γωνίας 60,0000 είναι 0,5000

3.8 Ασκήσεις κεφαλαίου 3

<u>Άσκηση 3.1</u>

Δημιουργήστε εφαρμογή lab_3_ask_1 η οποία ζητά από τον χρήστη να εισάγει την ακτίνα ενός κύκλου και στην συνέχεια υπολογίζει την περίμετρο και το εμβαδόν του κύκλου. Η περίμετρος και το εμβαδόν να εκτυπωθούν με ακρίβεια 4 δεκαδικών στοιχείων.

<u>Άσκηση 3.2</u>

Δημιουργήστε εφαρμογή lab_3_ask_2 η οποία ζητά από τον χρήστη να εισάγει μία από τις γωνίες (σε μοίρες) και το μήκος της υποτείνουσας ενός ορθογωνίου τριγώνου και στην συνέχεια υπολογίζει και εκτυπώνει στην οθόνη τα μήκη των πλευρών του τριγώνου με ακρίβεια 2 δεκαδικών στοιχείων.

<u>Υπόδειξη</u>: Η γωνία του τριγώνου δίνεται σε μοίρες και θα πρέπει να την μετατρέψετε σε ακτίνια (360° αντιστοιχούν σε 2π ακτίνια).

<u>Άσκηση 3.3</u>

Δημιουργήστε εφαρμογή lab_3_ask_3 η οποία ζητά από τον χρήστη να εισάγει το όνομα μιας Ελληνικής πόλης. Η εφαρμογή εξετάζει και εκτυπώνει εάν το όνομα της πόλης περιέχει αριθμούς, πεζούς λατινικούς χαρακτήρες, εάν αρχίζει με κεφαλαίο Ελληνικό χαρακτήρα και εάν η κατάληξη του είναι «ος».

<u>Άσκηση 3.4</u>

Δημιουργήστε εφαρμογή lab_3_ask_4 η οποία ζητά από τον χρήστη να εισάγει τον μισθό ενός υπαλλήλου και στην συνέχεια υπολογίζει και εκτυπώνει πόσα χαρτονομίσματα των 500, 200, 100, 50, 20 και 10 ευρώ θα χρειασθούν για να πληρωθεί το ποσό αυτό.

<u>Υπόδειξη</u>: Χρησιμοποιήστε την ακέραια διαίρεση και το υπόλοιπο της διαίρεσης με τα αντίστοιχα ποσά των χαρτονομισμάτων. Αρχίστε τον υπολογισμό με χαρτονομίσματα των 500 ευρώ.

4 Επιλογή και Επανάληψη

4.1 Η Εντολή Επιλογής if .. then

Η εντολή If .. Then .. χρησιμοποιείται για την λήψη λογικών αποφάσεων σε ένα πρόγραμμα. Η εντολή αυτή έχει διάφορες μορφές σύνταξης οι οποίες περιγράφονται στις παρακάτω ενότητες.

Η μορφή If .. Then .. End If

Η εντολή **If <λογική συνθήκη> Then <τμήμα προγράμματος>** εκτελεί ένα τμήμα προγράμματος εφόσον ικανοποιείται κάποια λογική συνθήκη.

Στο παρακάτω παράδειγμα η εντολή αυτή χρησιμοποιείται για να ελέγξει εάν ο αριθμός ο οποίος εισάγεται από τον χρήστη είναι θετικός. Στην περίπτωση που ο αριθμός είναι αρνητικός, τότε εκτυπώνεται το μήνυμα ότι ο αριθμός είναι αρνητικός και σταματά η εκτέλεση της διαδικασίας με την εντολή exit sub.

```
Public Sub main()
Dim i As Integer
Console.Write("Δώστε ένα θετικό ακέραιο")
i = Console.ReadLine()
'Έλεγχος αν ο αριθμός είναι θετικός
If i <= 0 Then
Console.WriteLine("Λάθος, ο αριθμός είναι αρνητικός")
Exit Sub ' Τέλος της διαδικασίας
End If
'Εκτύπωση τετραγωνικής ρίζας
Console.WriteLine(
"Η τετραγωνική του ρίζα είναι {0:N4}", Math.Sqrt(i))
End Sub
```

```
Παράδειγμα 4.1
```

Στην περίπτωση όπου το τμήμα προγράμματος το οποίο εκτελείται αποτελείται από μία μόνο εντολή, η εντολή μπορεί να συνταχθεί σε μία μόνο γραμμή.

 $\pi.\chi.$ if i <= 0 then Exit Sub

Η μορφή If .. Then .. Else .. End If

Η εντολή **If** <λογική συνθήκη> Then <τμήμα 1> Else <τμήμα 2> End If εκτελεί το πρώτο τμήμα προγράμματος εφόσον η λογική συνθήκη της εντολής είναι αληθής διαφορετικά εκτελεί το δεύτερο τμήμα προγράμματος το οποίο ορίζεται μεταξύ των λέξεων else και end if.

```
Public Sub main()
Dim i As Integer
Console.Write("Δώστε ένα ακέραιο")
i = Console.ReadLine()
'Έλεγχος αν ο αριθμός είναι θετικός ή αρνητικός
If i <= 0 Then
Console.WriteLine("Ο αριθμός είναι αρνητικός")
Else
Console.WriteLine("Ο αριθμός είναι θετικός")
End If
End Sub</pre>
```

Παράδειγμα 4.2

Στο παραπάνω παράδειγμα ελέγχεται εάν ο αριθμός ο οποίος εισάγεται από τον χρήστη είναι μεγαλύτερος ή ίσος του μηδενός και αναλόγως εκτυπώνεται το αντίστοιχο μήνυμα.

Το αποτέλεσμα της εκτέλεσης του προγράμματος εάν εισαχθεί ο αριθμός -45 είναι:

```
Δώστε ένα ακέραιο -45
Ο αριθμός είναι αρνητικός
```

```
Ένθεση εντολών If .. Then
```

Η εντολή επιλογής If .. Then μπορεί να εμπεριέχει ή ενθέτει και άλλες εντολές επιλογής.

Η χρήση ένθετων (εμφωλευμένων) εντολών If .. Then γίνεται όταν υπάρχουν περισσότερες από δύο εναλλακτικές περιπτώσεις διακλάδωσης. Εάν για παράδειγμα χρειάζεται να γίνει έλεγχος ενός αριθμού εάν είναι αρνητικός, μηδενικός ή θετικός, τότε γίνεται χρήση ένθετων εντολών επιλογής if .. then.

```
Public Sub main()
Dim i As Integer
Console.Write("Δώστε ένα ακέραιο ")
i = Console.ReadLine()
'Έλεγχος του αριθμού
If i < 0 Then
Console.WriteLine("Ο αριθμός είναι αρνητικός")
Else
If i = 0 Then
Console.WriteLine("Ο αριθμός είναι μηδενικός")
Else
Console.WriteLine("Ο αριθμός είναι θετικός")
End If
End If
End Sub</pre>
```

Παράδειγμα 4.3

Το αποτέλεσμα της εκτέλεσης του παραπάνω προγράμματος εάν δοθεί ο αριθμός 0 είναι: Δώστε ένα ακέραιο 0 Ο αριθμός είναι μηδενικός

Ένθεση με τη μορφή If .. then .. elseif .. else .. end if

Η σύνταξη των ένθετων εντολών if .. then μπορεί να απλοποιηθεί με την χρήση της λέξης elseIf. Με την χρήση της ένθετης μορφής if .. then .. elseIf .. else .. end if το προηγούμενο παράδειγμα μπορεί να απλοποιηθεί όπως φαίνεται στο παρακάτω πρόγραμμα:

```
Public Sub main()
    Dim i As Integer
    Console.Write("Δώστε ένα ακέραιο ")
    i = Console.ReadLine()
    'Έλεγχος του αριθμού
    If i < 0 Then
        Console.WriteLine("Ο αριθμός είναι αρνητικός")
    ElseIf i = 0 Then
        Console.WriteLine("Ο αριθμός είναι μηδενικός")
    Else
        Console.WriteLine("Ο αριθμός είναι θετικός")
    Else
        Console.WriteLine("Ο αριθμός είναι θετικός")
    Else
        Console.WriteLine("Ο αριθμός είναι θετικός")
</pre>
```

Παράδειγμα 4.4

4.2 Η εντολή πολλαπλής επιλογής Select

Η πολλαπλή επιλογή χρησιμοποιείται για την λήψη λογικών αποφάσεων όταν οι εναλλακτικές περιπτώσεις είναι περισσότερες από δύο. Η εντολή πολλαπλής επιλογής Select εξετάζει την τιμή κάποιας μεταβλητής (Integer, String κλπ) και αναλόγως της τιμής της μεταβλητής εκτελεί διαφορετικό τμήμα προγράμματος.

Το παρακάτω παράδειγμα με την εντολή Select εξετάζει την τιμή ενός μονοψήφιου θετικού ακέραιου ο οποίος εισάγεται από τον χρήστη, υπολογίζει την κατάλληλη αλφαριθμητική τιμή που αντιστοιχεί στον αριθμό και τον εκτυπώνει ολογράφως.

```
Public Sub main()
    Πρόγραμμα το οποίο ζητά έναν αριθμό από 0 έως 9
    και τον τυπώνει ολογράφως
   Dim i As Integer 'Ο αριθμός που θα μετατραπεί
    Dim s As String 'Ο αριθμός ολογράφως
    Console.Write("Δώστε ένα ακέραιο από 0 έως 9 : ")
    i = Console.ReadLine()
    Έλεγχος του αριθμού
    Select Case i
        Case 0
           s = "μηδέν"
        Case 1
           s = "ένα"
        Case 2
           s = "δύο"
        Case 3
           s = "τρία"
        Case 4
           s = "τέσσερα"
        Case 5
           s = "πέντε"
        Case 6
           s = "έξι"
        Case 7
           s = "επτά"
        Case 8
           s = "οκτώ"
        Case 9
           s = "εννέα"
        Case Else 'Εάν ο αριθμός δεν είναι από 0 εώς 9
           s = "****"
    End Select
    Console.WriteLine("Πληκτρολογήθηκε ο αριθμός {0}", s)
End Sub
```

Παράδειγμα 4.5

Εάν δοθεί ο αριθμός 4, τότε το αποτέλεσμα της εκτέλεσης του προγράμματος είναι:

Δώστε ένα ακέραιο από 0 έως 9 : 4 Πληκτρολογήθηκε ο αριθμός τέσσερα

Στην περίπτωση όπου η ίδια επιλογή γίνεται για δύο ή περισσότερες τιμές, μπορούμε να παραθέσουμε τις τιμές μετά την λέξη case με την χρήση κόμματος.

```
Π.χ. αν i είναι η μεταβλητή πολλαπλής επιλογής
Case 1,3,5 'ισοδυναμεί με i=1 or i=2 or i=3
```

Εάν η ίδια επιλογή γίνεται για ένα εύρος τιμών, το εύρος τιμών μπορεί να ορισθεί με την χρήση της αρχικής τιμής, της λέξης Το και της τελική τιμής του εύρους

πχ αν i είναι η μεταβλητή πολλαπλής επιλογής Case 0 To 5 'ισοδυναμεί με i>=0 and i<=5

Επίσης με την χρήση του τελεστή Is μπορούμε να συγκρίνουμε την τιμή της μεταβλητής πολλαπλής επιλογής με άλλες τιμές ή μεταβλητές.

πχ αν ί είναι η μεταβλητή πολλαπλής επιλογής Case Is < 0 'ισοδυναμεί με i<0

Η μεταβλητή η οποία χρησιμοποιείται στην πολλαπλή επιλογή μπορεί να είναι και άλλος τύπος δεδομένων, όπως πραγματικός αριθμός, χαρακτήρας ή αλφαριθμητική μεταβλητή.

```
Public Sub main()
    'Πρόγραμμα το οποίο διαβάζει ένα χαρακτήρα
    και εκτυπώνει την περιγραφή του
    Dim c As Char 'Χαρακτήρας ο οποίος εισάγεται
    Dim s As String 'Περιγραφή του χαρακτήρα
    Console.Write("Δώστε ένα ελληνικό χαρακτήρα : ")
    c = Console.ReadLine()
    ΙΠολλαπλή επιλογή ανάλογα με το χαρακτήρα
    Select Case c
        Case "α" Το "ω"
            s = "ελληνικός πεζός χαρακτήρας"
        Case "ά", "έ", "ό", "ή", "ύ", "ί", "ώ"
            s = "ελληνικός πεζός χαρακτήρας με τόνο"
        Case "A" To "\Omega"
            s = "ελληνικός κεφαλαίος χαρακτήρας"
        Case Is < " "
            s = "ειδικός χαρακτήρας"
        Case Else
            s = "κάποιος άλλος χαρακτήρας"
    End Select
    Console.WriteLine("Πληκτρολογήθηκε {0}", s)
End Sub
```

Παράδειγμα 4.6

Το παραπάνω παράδειγμα χρησιμοποιεί ως μεταβλητή πολλαπλής επιλογής ένα χαρακτήρα ο οποίος εισάγεται από το πληκτρολόγιο και κάνοντας χρήση πολλαπλών τιμών, εύρους τιμών και συγκρίσεων τιμών για την κάθε επιλογή εκτυπώνει την περιγραφή του χαρακτήρα που εισάγεται.

Το αποτέλεσμα της εκτέλεσης του προγράμματος εάν δοθεί ο χαρακτήρας «δ» είναι:

```
Δώστε ένα ελληνικό χαρακτήρα : δ
Πληκτρολογήθηκε ελληνικός πεζός χαρακτήρας
```

4.3 Η εντολή επιλογής ΙΙΓ

Η εντολή ΙΙΓ υπολογίζει μια λογική συνθήκη και αναλόγως επιστρέφει μια τιμή. Η σύνταξη της είναι:

<μεταβλητή> = IIF(<συνθήκη>, <τιμή 1>, <τιμή 2>)

Στο παρακάτω παράδειγμα η εντολή IIF χρησιμοποιείται για να ελέγξει εάν ένας αριθμός είναι άρτιος ή περιττός. Εάν το υπόλοιπο της διαίρεσης με τον αριθμό 2 είναι 0, τότε ο αριθμός είναι άρτιος διαφορετικά είναι περιττός.

```
Public Sub main()
    'Πρόγραμμα το οποίο ελέγχει
    'εάν ένας αριθμός είναι περιττός
    Dim i As Integer
    Dim s As String
    Console.Write("Δώστε ένα ακέραιο : ")
    i = Console.ReadLine()
    s = IIf(i Mod 2 = 0, "άρτιος", "περιττός")
    Console.WriteLine("Ο αριθμός {0} είναι {1}", i, s)
End Sub
```

Παράδειγμα 4.7

Το αποτέλεσμα της εκτέλεσης του προγράμματος εάν δοθεί ο αριθμός 35 είναι:

```
Δώστε ένα ακέραιο : 35
Ο αριθμός 35 είναι περιττός
```

Η εντολή IIF επιστρέφει μια τιμή η οποία είναι τύπου Object. Εάν χρησιμοποιείται η επιλογή Option Strict On, τότε πρέπει να χρησιμοποιείται και η κατάλληλη συνάρτηση ή μέθοδος για την μετατροπή της επιστρεφόμενης τιμής.

Στο προηγούμενο παράδειγμα, εάν ορισθεί η επιλογή Option Strict On, πρέπει να χρησιμοποιηθεί η συνάρτηση μετατροπής Cstr.

s = CStr(IIf(i Mod 2 = 0, "άρτιος", "περιττός"))

4.4 Η εντολή πολλαπλής επιλογής Choose

Η εντολή Choose επιστέφει μια τιμή ανάλογα με την τιμή κάποιας μεταβλητής η οποία χρησιμοποιείται ως δείκτης. Η σύνταξη της εντολής αυτής είναι:

```
<μεταβλητή> = Choose (δείκτης, <τιμή 1>, <τιμή 2>, .. , <τιμή N>)
```

Εάν ο δείκτης έχει την τιμή 1, τότε επιστρέφεται η πρώτη τιμή, εάν έχει τιμή 2 επιστρέφεται η δεύτερη τιμή κ.ο.κ.

Το παρακάτω παράδειγμα χρησιμοποιεί την εντολή choose για να εκτυπώσει ολογράφως την τιμή ενός ακεραίου αριθμού από 0 έως 9.

Παράδειγμα 4.8

Το αποτέλεσμα της εκτέλεσης του προγράμματος εάν δοθεί ο αριθμός 0 είναι:

```
Δώστε ένα ακέραιο από 0 έως 9 : 0
Πληκτρολογήθηκε ο αριθμός μηδέν
```
Η εντολή επανάληψης For 4.5

Η εντολή επανάληψης For επαναλαμβάνει την εκτέλεση ενός τμήματος του προγράμματος με την χρήση μιας μεταβλητής η οποία χρησιμοποιείται ως δείκτης των επαναλήψεων. Η γενική σύνταξη της εντολής είναι:

```
For <μεταβλητή> [As <Τύπος>]= <αρχική τιμή> To <τελική τιμή> [Step <βήμα>]
      [if <συνθήκη> then Continue For]
      [if <συνθήκη> then Exit For]
      <Τμήμα του προγράμματος το οποίο επαναλαμβάνεται>
Next
```

Ο ορισμός του βήματος των επαναλήψεων είναι προαιρετικός και μπορεί να παραλειφθεί. Σε περίπτωση που δεν έχει ορισθεί βήμα, τότε το βήμα είναι ίσο με την μονάδα.

Ο δείκτης των επαναλήψεων στην πρώτη επανάληψη παίρνει την αρχική τιμή και αυξάνεται διαδοχικά με το βήμα των επαναλήψεων μέχρι να φτάσει την τελική τιμή.

```
Public Sub main()
    Dim i As Integer
    For i = 1 To 5
        Console.WriteLine(i)
    Next
End Sub
```

Παράδειγμα 4.9

Στο παραπάνω παράδειγμα η μεταβλητή i χρησιμοποιείται ως δείκτης των επαναλήψεων με αρχική τιμή 1 και τελική τιμή 5. Το αποτέλεσμα της εκτέλεσης του προγράμματος είναι:

5

Το βήμα των επαναλήψεων μπορεί να είναι αρνητικό. Στην περίπτωση αυτή η τελική τιμή πρέπει να είναι μικρότερη από την τελική τιμή.

```
Public Sub main()
    Dim i As Integer
    For i = 5 To 1 Step -1
        Console.WriteLine(i)
    Next
End Sub
```

Παράδειγμα 4.10

Στο παραπάνω παράδειγμα η μεταβλητή i μειώνεται από την αρχική τιμή 5 στην τελική τιμή 1. Το αποτέλεσμα της εκτέλεσης του προγράμματος είναι:

Ο δείκτης των επαναλήψεων μπορεί να είναι οποιοδήποτε τύπος δεδομένων για τον οποίο ορίζονται οι τελεστές <=, >=, + και -. Μπορεί για παράδειγμα να χρησιμοποιηθεί πραγματικός τύπος μεταβλητής διπλής ακρίβειας με βήμα 0.1.

```
Public Sub main()

For x As Double = 10 To 10.5 Step 0.1

Console.WriteLine(x)

Next

End sub

Παράδειγμα 4.11
```

Το αποτέλεσμα της εκτέλεσης του παραπάνω προγράμματος είναι:

10 10,1 10,2 10,3 10,4 10,5

Η δήλωση του μεταβλητής του δείκτη των επαναλήψεων μπορεί να γίνει προαιρετικά στην ίδια γραμμή με την εντολή For.

```
Public Sub main()
    For i As Integer = 1 To 5
        Console.WriteLine(i)
        Next
End Sub
```

Παράδειγμα 4.12

Με την προαιρετική χρήση της εντολής Continue For σταματά η εκτέλεση της τρέχουσας επανάληψης και οι επαναλήψεις συνεχίζονται με την επόμενη τιμή του δείκτη των επαναλήψεων.

```
Public Sub main()
    For i As Integer = 1 To 5
        If i=3 then Continue For
        Console.WriteLine(i)
        Next
End Sub
```

Παράδειγμα 4.13

Στο παραπάνω παράδειγμα σταματά η εκτέλεση της επανάληψης όταν ο δείκτης έχει την τιμή 3. Το αποτέλεσμα της εκτέλεσης του προγράμματος είναι:

Με την προαιρετική εντολή Exit For διακόπτεται η εντολή επανάληψης

```
Public Sub main()
    For i As Integer = 1 To 5
        If i=3 then Exit For
        Console.WriteLine(i)
        Next
End Sub
```

Παράδειγμα 4.14

Στο παραπάνω παράδειγμα διακόπτονται όλες οι επαναλήψεις όταν ο δείκτης έχει την τιμή 3. Το αποτέλεσμα της εκτέλεσης του προγράμματος είναι:

1 2 Ο δείκτης των επαναλήψεων μπορεί να χρησιμοποιηθεί σε διάφορους υπολογισμός όπως για παράδειγμα την εύρεση του αθροίσματος ή του παραγοντικού ενός ακεραίου.

```
Public Sub main()
    'Πρόγραμμα το οποίο υπολογίζει
    'το άθροισμα των αριθμών από 1 έως 100
    Dim sum As Integer = 0
    For i As Integer = 1 To 100
        sum += i'Πρόσθεση του i στο sum
    Next
    Console.WriteLine(sum)
End Sub
```

Παράδειγμα 4.15 Εύρεση του αθροίσματος των αριθμών από 1 έως 100

Το παραπάνω παράδειγμα χρησιμοποιεί μια μεταβλητή sum ως αθροιστή με αρχική τιμή 0. Σε κάθε επανάληψη ο δείκτης επανάληψης ο οποίος παίρνει τιμές από 1 έως 100 προστίθεται στην μεταβλητή sum. Το sum μετά το τέλος των επαναλήψεων είναι ίσο με το άθροισμα των ακεραίων από 1 έως 100.

```
Public Sub main()
    'Πρόγραμμα το οποίο υπολογίζει
    'το παραγοντικό ενός αριθμού
    'n = 1*2*3*..n
    Dim n As Integer ' Ο ακέραιος αριθμός
    Dim p As Integer = 1 ' Το παραγοντικό
    Console.Write("Δώστε έναν ακέραιο: ")
    n = Console.ReadLine()
    For i As Integer = 2 To n
        p *= i ' p = p * i
    Next
    Console.WriteLine("Το παραγοντικό του {0} είναι {1}", n, p)
End Sub
```

Παράδειγμα 4.16 Εύρεση του παραγοντικού ενός ακέραιου

Το παραπάνω παράδειγμα χρησιμοποιεί την μεταβλητή p με αρχική τιμή 1 για να υπολογίσει το παραγοντικό ενός ακέραιου n ο οποίος εισάγεται από τον χρήστη. Σε κάθε επανάληψη το p γίνεται ίσο με γινόμενο της προηγούμενης τιμής του επί τον δείκτη των επαναλήψεων i ο οποίος παίρνει τις τιμές από 2 έως n.

Επανάληψη	Τιμή του n	Τιμή του p
1	2	1*2
2	3	1*2*3
3	4	1*2*3*4
n-1	n	1*2*3*4**n

Το αποτέλεσμα της εκτέλεσης του προγράμματος εάν δοθεί ο αριθμός 6 είναι:

```
Δώστε έναν ακέραιο: 6
Το παραγοντικό του 6 είναι 720
```

Ένθεση εντολών For

Το τμήμα του προγράμματος το οποίο επαναλαμβάνεται με μία εντολή For μπορεί να εμπεριέχει κάποια άλλη εντολή For. Είναι δηλαδή δυνατή η ένθεση εντολών For και γενικότερα εντολών επανάληψης.

```
Public Sub main()
    Dim i, j As Integer
    For i = 1 To 3
        For j = 1 To 2
            Console.WriteLine("i={0} j={1}", i, j)
            Next
    Next
End sub
```

Παράδειγμα 4.17

Το παραπάνω πρόγραμμα ενθέτει μια εντολή επανάληψης με δείκτη j ο οποίος παίρνει τις τιμές από 1 έως 2 σε μια εντολή επανάληψης με δείκτη i ο οποίος παίρνει τις τιμές από 1 έως 3. Η ένθεση έχει ως αποτέλεσμα η εντολή επανάληψης με δείκτη j να εκτελείται για κάθε αλλαγή της τιμής του i. Το αποτέλεσμα της εκτέλεσης του παραπάνω προγράμματος έχει σαν αποτέλεσμα:

i=1 j=1 i=1 j=2 i=2 j=1 i=2 j=2 i=3 j=1 i=3 j=2

Το παρακάτω παράδειγμα χρησιμοποιεί ένθετες εντολές For για να εκτυπώσει ένα πλαίσιο 5 γραμμών και 6 στηλών με τους χαρακτήρες «*». Στο τέλος της ένθετης εντολής επανάληψης με δείκτη j η οποία εκτυπώνει 6 χαρακτήρες «*» στην ίδια γραμμή, εκτελείται η εντολή Console.WriteLine() με την οποία γίνεται η αλλαγή γραμμής για κάθε διαφορετική τιμή του i.

```
Public Sub main()
   Dim i, j As Integer
   For i = 1 To 5
        For j = 1 To 6
            Console.Write("*")
        Next
        Console.WriteLine()
   Next
End Sub
```

Παράδειγμα 4.18

Το αποτέλεσμα της εκτέλεσης του παραπάνω προγράμματος είναι:

***** ***** ****** *****

4.6 Η εντολή επανάληψης Do Loop

Η εντολή επανάληψης Do Loop επαναλαμβάνει διαρκώς την εκτέλεση ενός τμήματος του προγράμματος. Οι επαναλήψεις συνεχίζουν ή σταματούν ανάλογα με την τιμή κάποιας λογικής συνθήκης. Εάν η λογική συνθήκη ελέγχεται στην αρχή τότε η εντολή έχει την μορφή:

```
Do [ While ή Until <συνθήκη> ]
        <εντολή>
        [if <συνθήκη> then Exit Do]
        [if <συνθήκη> then Continue Do]
        <εντολή>
Loop
```

Εάν η λογική συνθήκη ελέγχεται στην τέλος τότε η εντολή έχει την μορφή:

Do <εντολή> [if <συνθήκη> then Exit Do] [if <συνθήκη> then Continue Do] <εντολή> Loop [While ή Until <συνθήκη>]

Εάν χρησιμοποιείται ή δήλωση While οι επαναλήψεις διαρκούν όσο η τιμή της λογικής συνθήκης είναι αληθής. Αντίθετα όταν χρησιμοποιείται ή δήλωση Until οι επαναλήψεις διακόπτονται όταν η τιμή της λογικής συνθήκης γίνει αληθής.

Με την προαιρετική εντολή Exit Do διακόπτονται οι επαναλήψεις, ενώ με την προαιρετική εντολή Continue Do δεν εκτελούνται οι επόμενες εντολές του τμήματος του προγράμματος που επαναλαμβάνεται.

Η εντολή μπορεί επίσης να συνταχθεί χωρίς της χρήση λογικής συνθήκης στην αρχή ή το τέλος της εντολής.

Στην περίπτωση αυτή οι επαναλήψεις θα πρέπει να διακόπτονται με την εντολή Exit Do διαφορετικά οι επαναλήψεις θα συνεχίζουν διαρκώς.

Οι διάφορες μορφές της εντολής Do Loop δίνουν μεγάλη ευελιξία στην σύνταξη τμημάτων του προγράμματος τα οποία επαναλαμβάνονται. Μπορούν να χρησιμοποιηθούν στην θέση της εντολής For, αλλά και για την επανάληψη τμημάτων του προγράμματος όταν δεν είναι γνωστός εκ των προτέρων ο αριθμός των επαναλήψεων.

Τα παρακάτω παραδείγματα είναι ισοδύναμα με το παράδειγμα 4.9 όπου χρησιμοποιείται εντολή For για να εκτυπώσει του αριθμούς από ένα έως 5.

```
Public Sub main()

Dim i As Integer = 1 'Αρχική τιμή

Do While i <= 5 'Τελική τιμή

Console.WriteLine(i)

i += 1 'Βήμα

Loop

End Sub
```

Παράδειγμα 4.19 Έλεγχος συνθήκης στην αρχή με While

```
Public Sub main()

Dim i As Integer = 1 'Αρχική τιμή

Do Until i > 5 'Τελική τιμή

Console.WriteLine(i)

i += 1 'Βήμα

Loop

End Sub
```

Παράδειγμα 4.20 Έλεγχος συνθήκης στην αρχή με Until

```
Public Sub main()

Dim i As Integer = 1 'Αρχική τιμή

Do

Console.WriteLine(i)

i += 1 'Βήμα

Loop While i <= 5 'Τελική τιμή

End Sub
```

Παράδειγμα 4.21 Έλεγχος συνθήκης στην τέλος με While

```
Public Sub main()

Dim i As Integer = 1 'Αρχική τιμή

Do

Console.WriteLine(i)

i += 1 'Βήμα

Loop Until i > 5 'Τελική τιμή

End Sub
```

Παράδειγμα 4.22 Έλεγχος συνθήκης στην τέλος με Until

```
Public Sub main()
    Dim i As Integer = 1 'Αρχική τιμή
    Do
        Console.WriteLine(i)
        i += 1 'Βήμα
        If i > 5 Then Exit Do 'Τελική τιμή
        Loop
End Sub
```

Παράδειγμα 4.23 Χρήση Do Loop χωρίς έλεγχο αρχικής ή τελικής συνθήκης

Σε αρκετές περιπτώσεις ο αριθμός των επαναλήψεων δεν είναι εκ των προτέρων γνωστός. Στο παρακάτω παράδειγμα το πρόγραμμα διαβάζει διαδοχικά ακέραιους αριθμούς τους οποίους εισάγει ο χρήστης. Η εισαγωγή των αριθμών σταματά όταν ο χρήστης πληκτρολογήσει τον αριθμό 0. Το πρόγραμμα υπολογίζει και εκτυπώνει τον αριθμό των αρνητικών αριθμών που δόθηκαν από τον χρήστη.

```
Public Sub main()
        'Πρόγραμμα το οποίο μετρά τον αριθμό
        των αρνητικών αριθμών που εισάγει ο χρήστης
       Dim counter As Integer = 0 'Μετρητής αρνητικών αριθμών
       Dim i As Integer 'Αρχική τιμή
       Do
            Console.Write("Δώστε ένα ακέραιο ή Ο για τέλος: ")
            i = Console.ReadLine()
            If i < 0 Then counter += 1
       Loop While i <> 0
                                                                   ",
        Console.WriteLine("Δώσατε
                                    {0}
                                          αρνητικούς
                                                        αριθμούς
counter)
   End Sub
```

Παράδειγμα 4.24

Το αποτέλεσμα της εκτέλεσης του προγράμματος είναι: Δώστε ένα ακέραιο ή 0 για τέλος: -2 Δώστε ένα ακέραιο ή 0 για τέλος: 4 Δώστε ένα ακέραιο ή 0 για τέλος: -9 Δώστε ένα ακέραιο ή 0 για τέλος: 6 Δώστε ένα ακέραιο ή 0 για τέλος: 0 Δώσατε 2 αρνητικούς αριθμούς

4.7 Η εντολή επανάληψης While

Η εντολή επανάληψης while είναι αντίστοιχη της εντολής Do While Loop. Η σύνταξη της είναι:

```
While <συνθήκη>
     <εντολή>
    [if <συνθήκη> then Exit While]
    [if <συνθήκη> then Continue While]
     <εντολή>
End While
```

και είναι αντίστοιχη της:

```
Do While <συνθήκη>
  <εντολή>
  [if <συνθήκη> then Exit Do]
  [if <συνθήκη> then Continue Do]
  <εντολή>
Loop
```

Στο παρακάτω παράδειγμα η εντολή επανάληψης While χρησιμοποιείται για να υπολογίσει και εκτυπώσει πόσα χρόνια χρειάζονται για τον διπλασιασμό ενός αρχικού κεφαλαίου το οποίο κατατίθεται στην τράπεζα με σταθερό επιτόκιο και ανατοκισμό στο τέλος κάθε έτους.

```
Public Sub main()
    'Πρόγραμμα το οποίο υπολογίζει τα χρόνια που
    'χρειάζονται για το διπλασιασμό ενός κεφαλαίου
    Dim arxikoKefalaio As Single 'Το αρχικό κεφάλαιο
    Dim kefalaio As Single 'Το κεφάλαιο στο τέλος κάθε έτους
    Dim epitokio As Single 'Το επιτόκιο
    Console.Write ("Δώστε αρχικό κεφάλαιο: ")
    arxikoKefalaio = Console.ReadLine()
    Console.Write("Δώστε επιτόκιο: ")
    epitokio = Console.ReadLine()
    Dim metritis As Integer = 0 'Μετρητής ετών
    kefalaio = arxikoKefalaio
    While kefalaio < (2 * arxikoKefalaio)
        metritis += 1 'αύξηση έτους
        Πρόσθεση τόκου στο κεφάλαιο
        kefalaio += kefalaio * epitokio / 100
        Console.WriteLine(
         "Έτος {0} Κεφάλαιο :{1:N2}", metritis, kefalaio)
    End While
End Sub
```

Παράδειγμα 4.25

Το αποτέλεσμα της εκτέλεσης του παραπάνω προγράμματος με αρχικό κεφάλαιο 1000 και επιτόκιο 15 % είναι:

```
Δώστε αρχικό κεφάλαιο: 1000
Δώστε επιτόκιο: 15
Έτος 1 Κεφάλαιο: 1.150,00
Έτος 2 Κεφάλαιο: 1.322,50
Έτος 3 Κεφάλαιο: 1.520,88
Έτος 4 Κεφάλαιο: 1.749,01
Έτος 5 Κεφάλαιο: 2.011,36
Το κεφάλαιο διπλασιάζεται σε 5 έτη
```

4.8 Η εντολή επανάληψης For Each

Η εντολή επανάληψης For Each χρησιμοποιείται για να απαριθμήσει όλα τα μέλη ενός πίνακα ή άλλων συλλογών (Collections) δεδομένων.

Στο παρακάτω παράδειγμα η εντολή For Each χρησιμοποιείται για να εκτυπώσει τους χαρακτήρες μια αλφαριθμητικής μεταβλητής. Οι αλφαριθμητικές μεταβλητές για τις περισσότερες γλώσσες προγραμματισμού είναι πίνακες χαρακτήρων.

```
Public Sub main()
    Dim onoma As String
    Console.Write("Δώστε ένα όνομα: ")
    onoma = Console.ReadLine()
    Dim c As Char
    Console.WriteLine("Οι χαρακτήρες του ονόματος είναι:")
    For Each c In onoma
        Console.WriteLine(c)
    Next
End Sub
```

```
Δώστε ένα όνομα: Άρης
Οι χαρακτήρες του ονόματος είναι:
Ά
ρ
η
ς
```

Περισσότερα παραδείγματα χρήσης της εντολής αυτής θα δοθούν σε επόμενα κεφάλαια.

4.9 Η εντολή goto

Η εντολή goto αν και δεν είναι εντολή επανάληψης ή επιλογής, μπορεί να χρησιμοποιηθεί για την επιλογή ή επανάληψη υποπρογραμμάτων. Τα προγράμματα που χρησιμοποιούν εντολές goto γίνονται με δυσκολία κατανοητά και είναι ιδιαίτερα εύκολα να χαθεί ο έλεγχος της ροής του προγράμματος.

Η εντολή goto δίνει την εντολή στο πρόγραμμα να συνεχίσει την εκτέλεση του σε κάποιο άλλη γραμμή η σημείο του προγράμματος. Η εντολή goto παίρνει ως παράμετρο τον αριθμό γραμμής ή το όνομα ετικέτας (label) του προγράμματος στο οποίο θα γίνει η μετάβαση.

Ετικέτες μπορούμε να ορίσουμε σε οποιοδήποτε σημείο του προγράμματος με την χρήση του ονόματος της ετικέτας και τον χαρακτήρα «:».

Το παρακάτω πρόγραμμα εκτυπώνει τους αριθμούς από 1 έως 5 στην οθόνη.

```
Public Sub main()
Dim i As Integer = 1
arxi:
Console.Writeline(i)
i += 1
If i <= 5 Then GoTo arxi
End Sub
```

Η χρήση της εντολής αυτής δεν ενδείκνυται παρά μόνο σε ελάχιστες περιπτώσεις όπως όταν χρειάζεται να διακοπούν ένθετες εντολές επανάληψης και ο έλεγχος να δοθεί σε συγκεκριμένο σημείο του προγράμματος η για να αντιμετωπισθούν λάθη κατά την εκτέλεση του προγράμματος.

4.10 Ασκήσεις κεφαλαίου 4

<u>Άσκηση 4.1</u>

Δημιουργήστε εφαρμογή lab_4_ask_1 η οποία ζητά από τον χρήστη να εισάγει δύο πραγματικούς αριθμούς x και y και στην συνέχεια:

- εξετάζει και εκτυπώνει εάν ο αριθμός x ανήκει στο διάστημα [1..10]
- εκτυπώνει τον μεγαλύτερο αριθμό.

<u>Άσκηση 4.2</u>

Δημιουργήστε εφαρμογή lab_4_ask_2 η οποία ζητά από τον χρήστη να εισάγει έναν ακέραιο από 0 έως 9999 και στην συνέχεια κάνοντας χρήση της εντολής επιλογής Select εκτυπώνει τον αριθμό των ψηφίων του αριθμού (μονοψήφιος, διψήφιος, τριψήφιος ή τετραψήφιος).

<u>Άσκηση 4.3</u>

Δημιουργήστε εφαρμογή lab_4_ask_3 η οποία ζητά από τον χρήστη να εισάγει τα συνολικό εισόδημα του και στην συνέχεια υπολογίζει και εκτυπώνει το φόρο τον οποίο θα πρέπει να πληρώσει.

Ο φορολογικός συντελεστής είναι:

- Για τα πρώτα 8.400 Ευρώ 0%
- Για τα επόμενα 5.000 Ευρώ 15%
- Για τα επόμενα 10.000 Ευρώ 30%
- Για επιπλέον εισόδημα 40%

Пχ.

Για εισόδημα 20.000 Έυρώ ο φόρος είναι 8400*0%+5000*15%+(20000-13400)*30%=2730

<u>Άσκηση 4.4</u>

Δημιουργήστε εφαρμογή lab_4_ask_4 η οποία ζητά από τον χρήστη να εισάγει τρεις πραγματικούς αριθμούς και την συνέχεια τους εκτυπώνει στην οθόνη σε αύξουσα σειρά.

Υπόδειξη: Χρησιμοποιήστε ένθετες εντολές Ιf για να ελέγξετε όλες τις περιπτώσεις

<u>Άσκηση 4.5</u>

Δημιουργήστε εφαρμογή lab_4_ask_5 η οποία κάνοντας χρήση εντολών επανάληψης εκτυπώνει τις αντιστοιχία σε Ευρώ των 100,200,300 έως 1000 δραχμών

<u>Άσκηση 4.6</u>

Δημιουργήστε εφαρμογή lab_4_ask_6 η οποία υπολογίζει και εκτυπώνει το άθροισμα των περιττών ακεραίων από 1 έως 199.

<u>Άσκηση 4.7</u>

Δημιουργήστε εφαρμογή lab_4_ask_7 η οποία ζητά από τον χρήστη να εισάγει ένα ακέραιο αριθμό N και στην συνέχεια εκτυπώνει τις τιμές των παρακάτω παραστάσεων:

- 1+2+3+..+N
- $1^2 + 2^2 + 3^2 + ... + N^2$
- 1*2*3* .. *N

<u>Άσκηση 4.8</u>

Δημιουργήστε εφαρμογή lab_4_ask_8 η οποία με την χρήση ένθετων εντολών επανάληψης εμφανίζει στην οθόνη το παρακάτω σχήμα:

<u>Άσκηση 4.9</u>

Δημιουργήστε εφαρμογή lab_4_ask_9 η οποία ζητά από τον χρήστη να εισάγει διαδοχικά 30 ακεραίους αριθμούς και στην συνέχεια εκτυπώνει το άθροισμα τους, τον μέσο όρο τους, τον μεγαλύτερο αριθμό και το πλήθος των αρνητικών αριθμών.

<u>Άσκηση 4.10</u>

Δημιουργήστε εφαρμογή lab_4_ask_10 η οποία κάνοντας χρήση εντολής επανάληψης Do Loop, υπολογίζει και εκτυπώνει τα ημίτονα και συνημίτονα των γωνιών από 10 έως 20 μοίρες με βήμα 0.1

<u>Άσκηση 4.11</u>

Δημιουργήστε εφαρμογή lab_4_ask_11 η οποία διαβάζει διαδοχικά ακεραίους αριθμούς τους οποίους εισάγει ο χρήστης. Η εισαγωγή των αριθμών σταματά όταν ο χρήστης πληκτρολογήσει τον αριθμό 0 και στην συνέχεια το πρόγραμμα εκτυπώνει στην οθόνη το άθροισμα τους, τον μικρότερο αριθμό, το πλήθος των περιττών αριθμών και το πλήθος των αριθμών οι οποίοι είναι μεγαλύτεροι του 20.

<u>Άσκηση 4.12</u>

Δημιουργήστε εφαρμογή lab_4_ask12 η υπολογίζει την μικρότερη τιμή του N για την οποία η παρακάτω παράσταση είναι μεγαλύτερη του 65000. $1^2 + 2^2 + 3^2 + .. + N^2$

<u>Άσκηση 4.13</u>

Σφαίρα αφήνεται να πέσει από αρχικό ύψος x. Σε κάθε αναπήδηση η σφαίρα χάνει το 10% του προηγούμενου ύψους. Η σφαίρα σταματά την κίνηση της όταν το ύψος της γίνεται <= 0.01 . Γράψτε εφαρμογή lab_4_ask_13 ζητά από τον χρήστη να δώσει το αρχικό ύψος x και στην συνέχεια υπολογίζει και εκτυπώνει τον αριθμό των αναπηδήσεων που θα κάνει η σφαίρα.

<u>Άσκηση 4.14</u>

Δημιουργήστε εφαρμογή lab_4_ask_14 η οποία ζητά από τον χρήστη να εισάγει δύο αριθμούς x και y και στην συνέχεια υπολογίζει το υπόλοιπο της διαίρεσης x / y με την χρήση διαδοχικών αφαιρέσεων του y από το x.

<u>Άσκηση 4.15</u>

Δημιουργήστε εφαρμογή lab_4_ask_15 η οποία εκτυπώνει τους Πυθαγόρειους αριθμούς από το 1 μέχρι το 100. Τρεις ακέραιοι αριθμοί α, β, γ λέγονται Πυθαγόρειοι αν ισχύει : $\alpha^2 + \beta^2 = \gamma^2$

Υπόδειξη: Χρησιμοποιήστε τρεις ένθετες εντολές επανάληψης με μεταβλητές a,b και c.

<u>Άσκηση 4.16</u>

Δημιουργήστε εφαρμογή lab_4_ask_16 η οποία ζητά από τον χρήστη να εισάγει έναν ακέραιο αριθμό, εκτυπώνει τους διαιρέτες του αριθμού και εκτυπώνει εάν ο αριθμός αυτός είναι πρώτος ή όχι.

Υπόδειξη: Πρώτος αριθμός είναι ο αριθμός ο οποίος διαιρείται ακριβώς μόνο με την μονάδα και τον εαυτό του. Χρησιμοποιήστε εντολή επανάληψης για να εκτυπώσετε και να μετρήσετε τους διαιρέτες του αριθμού. Ανάλογα με το αποτέλεσμα της μέτρησης εκτυπώστε εάν ο αριθμός είναι πρώτος ή όχι.

5 Πίνακες δεδομένων

Η Visual Basic υποστηρίζει μονοδιάστατους, δισδιάστατους ή και πίνακες περισσοτέρων διαστάσεων. Τα στοιχεία ενός πίνακα ανήκουν στον ίδιο τύπο δεδομένων.

5.1 Μονοδιάστατοι πίνακες

Ο ορισμός ενός μονοδιάστατου πίνακα τύπου δεδομένων γίνεται με την δήλωση:

```
Dim <όνομα πίνακα> (<πλήθος στοιχείων - 1 >) as <τύπος δεδομένων>
```

π.χ. η δήλωση

Dim m(6) As Integer

ορίζει ένα μονοδιάστατο πίνακα 7 ακεραίων με δείκτες στοιχείων του πίνακα από 0 έως 6.

Η αναφορά σε ένα στοιχείο του πίνακα (για αποθήκευση ή ανάγνωση της τιμής του) γίνεται με την χρήση του ονόματος του πίνακα ακολουθούμενο από τον δείκτη του στοιχείου εντός παρενθέσεων.

Π.χ.

m(0) = 45 ' Απόδοση τιμής στο πρώτο στοιχείο του πίνακα

Console.WriteLine(m(0)) ' Εκτύπωση της τιμής του πρώτου στοιχείου

Το πρώτο στοιχείο ενός μονοδιάστατου πίνακα έχει πάντα τον δείκτη 0 και το τελευταίο στοιχείο του πίνακα έχει ως δείκτη το πλήθος των στοιχείων ελαττωμένο κατά μία μονάδα.

Ο ορισμός ενός πίνακα μπορεί να γίνει και με τον ταυτόχρονο καθορισμό των τιμών των στοιχείων του πίνακα. Στην περίπτωση αυτή παραλείπεται ο ορισμός του πλήθους των στοιχείων του πίνακα.

Π.χ.

Dim m() As Integer = {10, 11, 45, 23, 45, 8}

Η επεξεργασία των στοιχείων ενός πίνακα γίνεται συνήθως με την χρήση εντολών επανάληψης. Εντολές επανάληψης χρησιμοποιούνται επίσης για την εισαγωγή των στοιχείων από το πληκτρολόγιο, στην περίπτωση κατά την οποία τα στοιχεία του πίνακα δεν ορίζονται με εντολές του προγράμματος αλλά είναι δυναμικά και εισάγονται από το πληκτρολόγιο κάθε φορά που γίνεται η εκτέλεση του προγράμματος.

Στο παρακάτω παράδειγμα 5.1 ορίζεται ένας πίνακας 6 ακεραίων , γίνεται η εισαγωγή των στοιχείων του από το πληκτρολόγιο, εκτυπώνονται τα στοιχεία του πίνακα και υπολογίζεται και εκτυπώνεται ο μέσος όρος, το μικρότερο στοιχείο, το μεγαλύτερο στοιχείο και το πλήθος των μηδενικών στοιχείων του πίνακα.

```
Public Sub main()

Πρόγραμμα το οποίο ορίζει πίνακα 6 ακεραίων,
'διαβάζει τα στοιχεία από το πληκτρολόγιο,
'εκτυπώνει τον πίνακα και υπολογίζει το μέσο όρο,
'το μικρότερο , το μεγαλύτερο και το πλήθος των
'μηδενικών στοιχείων του πίνακα

Dim m(6) As Integer 'Ορισμός πίνακα
Dim i As Integer 'Δείκτης επανάληψης
Dim sum As Integer 'Άθροισμα στοιχείων πίνακα
Dim min As Integer 'Μικρότερο στοιχείο πίνακα
```

```
Dim n As Integer 'Πλήθος μηδενικών στοιχείων
   'Εισαγωγή στοιχείων από το πληκτρολόγιο
   Console.WriteLine("Εισαγωγή στοιχείων πίνακα")
   Console.WriteLine("-----")
   For i = 0 To 5
       Console.Write("Δώστε το στοιχείο m({0}): ", i)
       m(i) = Console.ReadLine()
   Next
   'Εκτύπωση στοιχείων πίνακα
   Console.WriteLine()
   Console.WriteLine("Τα στοιχεία του πίνακα είναι:")
   Console.WriteLine("-----")
   For i = 0 To 5
       Console.WriteLine(" m({0}) = {1,10} ", i, m(i))
   Next
   Console.WriteLine()
   Console.WriteLine("Επεξεργασία στοιχείων πίνακα")
   Console.WriteLine("-----")
   'Έυρεση μέσου όρου
   sum = 0
   For i = 0 To 5
       sum += m(i)
   Next
                                       = \{0, 8: N2\}", sum / 6)
   Console.WriteLine("Méσoç ópoç
   'Έυρεση μικρότερου στοιχείου
   \min = m(0)
   For i = 0 To 5
       If m(i) < \min Then \min = m(i)
   Next
   Console.WriteLine("Μικρότερο στοιχείο = {0,8}", min)
   'Ευρεση μεγαλύτερου στοιχείου
   max = m(0)
   For i = 0 To 5
       If m(i) > max Then max = m(i)
   Next
   Console.WriteLine("Μεγαλύτερο στοιχείο= {0,8}", max)
   'Πλήθος μηδενικών στοιχείων
   n = 0
   For i = 0 To 5
       If m(i) = 0 Then n \neq 1
   Next
   Console.WriteLine("Πλήθος μηδενικών = \{0, 8\}", n)
End Sub
```

Παράδειγμα 5.1

Το αποτέλεσμα της εκτέλεσης του παραδείγματος 5.1 είναι το παρακάτω:

```
Εισαγωγή στοιχείων του πίνακα
-----
Δώστε το στοιχείο m(0): 9
Δώστε το στοιχείο m(1): 0
Δώστε το στοιχείο m(2): -23
```

Δώστε	το	στο	ιχε	ίo	m(3)	:	12	
Δώστε	το	στο	ιχε	ίo	m(4)	:	0	
Δώστε	το	στο	ιχε	ίo	m(5)	:	45	
Τα στο	οιχε	εία	του	πίν	νακα	εί	ναι:	
m(0)	=			9				
m(1)	=			0				
m(2)	=			-23				
m(3)	=			12				
m(4)	=			0				
m (5)	=			45				
Επεξερ	ογαα	σία	στο	ιχεί	ίων π	ίv	ακα	
 Μέσος	όρα	 >ς			=		7,17	
Μικρότ	ερα	στ	οιχ	είο	=		-23	
Μεγαλύ	ίτερ	οσ	τοι	χεία	> =		45	
Πλήθος	μι	ιδεν	ικώ	v	=		2	

5.2 Δισδιάστατοι πίνακες

Ο ορισμός ενός δισδιάστατου πίνακα τύπου δεδομένων γίνεται με την δήλωση:

Dim <όνομα πίνακα> (<γραμμές - 1 > <στήλες -1>) as <τύπος δεδομένων>

π.χ. η δήλωση

Dim m(2, 2) As Double

ορίζει ένα δισδιάστατο πίνακα 3x3 ακεραίων.

Πίνακες περισσοτέρων διαστάσεων ορίζονται με την δήλωση των δεικτών της τρίτης, της τέταρτης διάστασης κ.ο.κ.

Π.χ. η δήλωση:

Dim m(2, 2, 2) As Double

ορίζει ένα τρισδιάστατο πίνακα 3x3x3 ακεραίων

Η αναφορά στα στοιχεία ενός δισδιάστατου του πίνακα γίνεται με την χρήση δύο δεικτών, ο πρώτος καθορίζει την γραμμή και ο δεύτερος την στήλη του πίνακα. Η αρίθμηση των γραμμών και των στηλών ξεκινά με την τιμή 0.

Π.χ.

m(0, 0) = 45 ' Απόδοση τιμής στο στοιχείο m₀₀ του πίνακα

Console.WriteLine(m(0, 0)) ' Εκτύπωση της τιμής του στοιχείου m₀₀

Η απόδοση τιμών στα στοιχεία ενός δισδιάστατου πίνακα μπορεί να γίνει ταυτόχρονα με τον ορισμό του όπως φαίνεται στο παρακάτω παράδειγμα.

Π.χ.

```
Dim m(,) As Double = { {1, 12, 3}, _
{4, 5, 6}, _
{7, 18, 9}}
```

Στο παρακάτω παράδειγμα 5.2 ορίζεται ένας πίνακας 3x3 ακεραίων και όπως και στο παράδειγμα 5.1 για μονοδιάστατους πίνακες, γίνεται η εισαγωγή των στοιχείων του από το πληκτρολόγιο, εκτυπώνονται τα στοιχεία του πίνακα και υπολογίζεται και εκτυπώνεται ο μέσος όρος, το μικρότερο στοιχείο, το μεγαλύτερο στοιχείο και το πλήθος των μηδενικών στοιχείων του πίνακα.

```
Public Sub main()
    'Πρόγραμμα το οποίο ορίζει πίνακα 3Χ3 πραγματικών,
    'διαβάζει τα στοιχεία από το πληκτρολόγιο,
    'εκτυπώνει τον πίνακα και υπολογίζει το μέσο όρο,
    'το μικρότερο, το μεγαλύτερο και το πλήθος των
    'μηδενικών στοιχείων του πίνακα
   Dim m(2, 2) As Double 'Ορισμός πίνακα
   Dim i As Integer 'Δείκτης επανάληψης γραμμών
   Dim j As Integer 'Δείκτης επανάληψης στηλών
   Dim sum As Double 'Άθροισμα στοιχείων πίνακα
   Dim min As Double 'Μικρότερο στοιχείο πίνακα
   Dim max As Double 'Μεγαλύτερο στοιχείο πίνακα
   Dim n As Integer 'Πλήθος μηδενικών στοιχείων
    'Εισαγωγή στοιχείων από το πληκτρολόγιο
   Console.WriteLine("Εισαγωγή στοιχείων πίνακα")
   Console.WriteLine("-----")
   For i = 0 To 2
       For j = 0 To 2
           Console.Write("Δώστε το στοιχείο m({0}, {1}): ", i,j)
           m(i, j) = Console.ReadLine()
       Next
   Next
    'Εκτύπωση στοιχείων πίνακα
   Console.WriteLine()
   Console.WriteLine("Τα στοιχεία του πίνακα είναι:")
   Console.WriteLine("-----")
   For i = 0 To 2
       For j = 0 To 2
           Console.Write("{0,8:N2} ", m(i, j))
       Next
       Console.WriteLine()
   Next
   Console.WriteLine()
   Console.WriteLine("Επεξεργασία στοιχείων πίνακα")
   Console.WriteLine("-----")
    'Έυρεση μέσου όρου
   sum = 0
   For i = 0 To 2
       For j = 0 To 2
           sum += m(i, j)
       Next
   Next
   Console.WriteLine("Mέσος όρος
                                  = \{0, 8: N2\}", sum / 9)
    'Έυρεση μικρότερου στοιχείου
   \min = m(0, 0)
   For i = 0 To 2
       For j = 0 To 2
           If m(i, j) < min Then min = m(i, j)
       Next
```

```
Next
    Console.WriteLine("Μικρότερο στοιχείο = {0,8:N2}", min)
    'Έυρεση μεγαλύτερου στοιχείου
    \max = m(0, 0)
    For i = 0 To 2
        For j = 0 To 2
            If m(i, j) > max Then max = m(i, j)
        Next
    Next
    Console.WriteLine("Μεγαλύτερο στοιχείο= {0,8:N2}", max)
    'Πλήθος μηδενικών στοιχείων
    n = 0
    For i = 0 To 2
        For j = 0 To 2
            If m(i, j) = 0 Then n \neq 1
        Next
    Next
    Console.WriteLine("Πλήθος μηδενικών = \{0, 8\}", n)
End Sub
```

Παράδειγμα 5.2

Το αποτέλεσμα της εκτέλεσης του παραπάνω προγράμματος είναι:

```
Εισαγωγή στοιχείων πίνακα
```

```
_____
Δώστε το στοιχείο m(0,0): 1,23
Δώστε το στοιχείο m(0,1): -4,876
Δώστε το στοιχείο m(0,2): 2
Δώστε το στοιχείο m(1,0): -98,2
Δώστε το στοιχείο m(1,1): 1
Δώστε το στοιχείο m(1,2): 0
Δώστε το στοιχείο m(2,0): 2,4
Δώστε το στοιχείο m(2,1): 45
Δώστε το στοιχείο m(2,2): Ο
Τα στοιχεία του πίνακα είναι:
------
 1,23 -4,88 2,00
-98,20 1,00 0,00
2,40 45,00 0,00
Επεξεργασία στοιχείων πίνακα
 _____
Μέσος όρος = -5,72
Μικρότερο στοιχείο = -98,20
Μεγαλύτερο στοιχείο= 45,00
Πλήθος μηδενικών = 2
```

Συχνά είναι αναγκαίος ο ορισμός και η απόδοση τιμών σε ένα πίνακα με την χρήση κάποιων σχέσεων ή ιδιοτήτων. Στο παρακάτω παράδειγμα δημιουργείται ένας τριγωνικός πίνακας 8x8 και εκτυπώνεται. Το πρόγραμμα δίνει την τιμή 1 στο στοιχεία του πίνακα τα οποία βρίσκονται πάνω ή μετά την κύρια διαγώνιο του πίνακα και την τιμή 0 στα υπόλοιπα.

```
Public Sub main()
    Πρόγραμμα το οποίο δημιουργεί ένα τριγωνικό πίνακα
    και τον εκτυπώνει
    Dim m(7, 7) As Integer 'Ορισμός πίνακα 8X8
    Dim i, j As Integer 'Δείκτες πίνακα
    'Απόδοση τιμής στα στοιχεία του πίνακα
    For i = 0 To 7
        For j = 0 To 7
            Έλεγχος εάν το στοιχείο είναι
            'πάνω η μετά την κύρια διαγώνιο
            If j >= i Then
                m(i, j) = 1
            Else
                m(i, j) = 0
            End If
        Next
    Next
    'Εκτύπωση πίνακα
    For i = 0 To 7
        For j = 0 To 7
            Console.Write("{0,2}", m(i, j))
        Next
        Console.WriteLine()
    Next
End Sub
```

5.3 Ταξινόμηση, και αναζήτηση στοιχείων πινάκων

Η Visual Basic παρέχει διάφορες χρήσιμες μεθόδους και συναρτήσεις για τον χειρισμό, και επεξεργασία πινάκων.

Η μέθοδος Length επιστρέφει το πλήθος των στοιχείων ενός πίνακα (μίας η περισσοτέρων διαστάσεων).

Οι μέθοδοι GetLowerBound και GetUpperBound επιστρέφουν τον αρχικό και τον τελικό δείκτη της διάστασης η οποία δίνεται ως παράμετρος εισόδου (0 για την πρώτη διάσταση, 1 για την δεύτερη διάσταση κ.ο.κ.).

Π.χ. αν ο m είναι ένας δισδιάστατος πίνακας, η παρακάτω εντολή εκτυπώνει τον τελικό δείκτη της δεύτερης διάστασης του πίνακα (των στηλών).

Console.WriteLine(m.GetUpperBound(1))

Εναλλακτικά, αντί των μεθόδων GetLowerBound και GetUpperBound μπορούν να χρησιμοποιηθούν οι συναρτήσεις LBound and Ubound αντίστοιχα. Οι συναρτήσεις αυτές δέχονται ως πρώτη παράμετρο τον πίνακα και ως δεύτερη παράμετρο την διάσταση (1 για την πρώτη διάσταση, 2 για την δεύτερη διάσταση κ.ο.κ.).

Πχ. η παρακάτω εντολή εκτυπώνει τον τελικό δείκτη της δεύτερης διάστασης του δισδιάστατου πίνακα m.

Console.WriteLine(Ubound(m,2))

Ειδικά για τους μονοδιάστατους πίνακες υπάρχουν και οι μέθοδοι IndexOf και Sort της κλάσης αντικειμένων Array.

Η μέθοδος IndexOf επιστρέφει τον δείκτη ενός στοιχείου του πίνακα ή τον αριθμό -1 εάν το στοιχείο δεν ανήκει στον πίνακα. Π.χ. εάν m είναι μονοδιάστατος πίνακας ακεραίων, η παρακάτω εντολή εκτυπώνει τον δείκτη του στοιχείου με τιμή 2

```
Console.WriteLine(Array.IndexOf(m, 2))
```

Η μέθοδος Sort ταξινομεί τα στοιχεία ενός μονοδιάστατου πίνακα. Π.χ. εάν m είναι μονοδιάστατος πίνακας, η παρακάτω εντολή ταξινομεί τα στοιχεία του.

```
Array.Sort(m)
```

Στο παρακάτω παράδειγμα δίνονται παράδειγμα χρήσης των μεθόδων αυτών για ένα μονοδιάστατο πίνακα m1 και για ένα δισδιάστατο πίνακα m2.

```
Public Sub main()
    'Παράδειγμα μεθόδων μονοδιάστατου πίνακα
    Dim m1() As Double = \{12, 22.3, 4.1, 6.7, 9, 12\}
    Dim i As Integer
    Console.WriteLine("\Pi \lambda \eta \theta \circ \varsigma \sigma \tau \circ \iota \chi \epsilon (\omega v m 1 = \{0\})", m1.Length)
    Console.WriteLine(
      "Άρχικός δείκτης στοιχείων m1 = {0}", m1.GetLowerBound(0))
    Console.WriteLine(
      "Τελικός δείκτης στοιχείων m1 = {0}", m1.GetUpperBound(0))
    Console.WriteLine("")
    'Εύρεση δείκτη στοιχείου πίνακα
    Console.WriteLine(
      "Το 4.1 είναι στην θέση = {0} ", Array.IndexOf(m1, 4.1))
    Console.WriteLine("")
    'Ταξινόμιση μονοδιάστατου πίνακα
    Console.WriteLine("Τα στοιχεία πριν την ταξινόμιση είναι:")
    For i = 0 To m1.GetUpperBound(0)
        Console.Write("{0,6:N2} ", m1(i))
    Next
    Console.WriteLine("")
    Array.Sort(m1)
    Console.WriteLine("Τα στοιχεία μετά την ταξινόμιση είναι:")
    For i = 0 To m1.GetUpperBound(0)
        Console.Write("{0,6:N2} ", m1(i))
    Next
    Console.WriteLine("")
    Console.WriteLine("")
    'Παράδειγμα μεθόδων δισδιάστατου πίνακα
    Dim m2(,) As Double = {{1, 12}, _
                             {4, 51},
                             \{7, 18\}\}
    Console.WriteLine(
     "Δείκτης πρώτης γραμμής m2 = {0}", m2.GetLowerBound(0))
    Console.WriteLine(
     "Δείκτης τελευταίας γραμμής m2 = \{0\}", m2.GetUpperBound(0))
```

Παράδειγμα 5.3

Πίνακες

Το αποτέλεσμα της εκτέλεσης του παραπάνω προγράμματος είναι:

```
Πλήθος στοιχείων m1 = 6
Άρχικός δείκτης στοιχείων m1 = 0
Τελικός δείκτης στοιχείων m1 = 5
Το 4.1 είναι στην θέση = 2
Τα στοιχεία πριν την ταξινόμιση είναι:
12,00 22,30 4,10 6,70 9,00 12,00
Τα στοιχεία πριν την ταξινόμιση είναι:
4,10 6,70 9,00 12,00 12,00 22,30
Δείκτης πρώτης γραμμής m2 = 0
Δείκτης πρώτης στήλης m2 = 0
Δείκτης τελευταίας στήλης m2 = 1
Πλήθος στοιχείων m2 = 6
```

5.4 Δυναμική τροποποίηση του μεγέθους ενός πίνακα

Το πλήθος των στοιχείων ενός πίνακα είναι σταθερό. Εάν όμως χρειάζεται να τροποποιηθεί δυναμικά (κατά την εκτέλεση ενός προγράμματος) το πλήθος των στοιχείων ενός πίνακα, τότε χρησιμοποιείται η δήλωση *Redim* ή η δήλωση *Redim Preserve*. Με την δήλωση *Redim Preserve* οι τιμές των στοιχείων του πίνακα διατηρούνται μετά την αλλαγή του πλήθους των στοιχείων.

Στο παρακάτω παράδειγμα ο χρήστης ορίζει δυναμικά το πλήθος των στοιχείων ενός πίνακα και στην συνέχεια εισάγει τα στοιχεία του πίνακα.

```
Public Sub main()
    'Παράδειγμα δυναμικής τροποποίησης του μεγέθους ενός πίνακα
    Dim m() As Double
    Dim i, n As Integer
    Console.Write("Δώσε πλήθος στοιχείων: ")
    n = Console.ReadLine()
    'Αλλαγή μεγέθους πίνακα
    ReDim Preserve m(n - 1)
    'Εισαγωγή στοιχείων
    For i = 0 To m.GetUpperBound(0)
        Console.Write("Δώσε στοιχείο m({0}): ", i)
        m(i) = Console.ReadLine()
    Next
    'Εκτύπωση των στοιχείων με την εντολή for each
    Console.Write("Τα στοιχεία που δώσατε είναι:")
    For Each d As Double In m
        Console.Write("{0,6:N2} ", d)
```

Next Console.WriteLine()

End Sub

Παράδειγμα 5.4

Το αποτέλεσμα της εκτέλεσης του παραπάνω προγράμματος είναι:

Δώσε πλήθος στοιχείων: 3 Δώσε στοιχείο m(0): 1 Δώσε στοιχείο m(1): 2,345 Δώσε στοιχείο m(2): 1,23 Τα στοιχεία που δώσατε είναι: 1,00 2,35 1,23

5.5 Ασκήσεις Κεφαλαίου 5

<u>Άσκηση 5.1</u>

Δημιουργήστε εφαρμογή lab_5_ask_1 η οποία ορίζει μονοδιάστατο πίνακα 8 πραγματικών αριθμών, ζητά από τον χρήστη να εισάγει τα στοιχεία του πίνακα, και εκτυπώνει:

- τον πίνακα
- το άθροισμα, τον μέσο όρο, το μικρότερο και το μεγαλύτερο στοιχείο του πίνακα
- το πλήθος των αρνητικών του στοιχείων του πίνακα

<u>Άσκηση 5.2</u>

Δημιουργήστε εφαρμογή lab_5_ask_2 η οποία ορίζει μονοδιάστατο πίνακα 6 ακεραίων, ζητά από τον χρήση να εισάγει τα στοιχεία του πίνακα, ταξινομεί τον πίνακα, εκτυπώνει τον ταξινομημένο πίνακα, και στην περίπτωση που ο πίνακας περιέχει τον αριθμό 25 εκτυπώνει την θέση του (τον δείκτη του στοιχείου).

<u>Άσκηση 5.3</u>

Δημιουργήστε εφαρμογή lab_5_ask_3 ζητά από τον χρήστη να εισάγει σε πίνακα πραγματικών αριθμών τις ελάχιστες θερμοκρασίες 7 διαδοχικών ημερών μιας εβδομάδος και εκτυπώνει:

- Το μέσο όρο της ελάχιστης θερμοκρασίας
- Το πλήθος των ημερών με ελάχιστη θερμοκρασία άνω των 4 βαθμών
- Το πλήθος των ημερών με ελάχιστη θερμοκρασία κάτω του μηδενός
- Τον μέσο όρο της ελάχιστης θερμοκρασίας των ημερών με θερμοκρασία κάτω του μηδενός

<u>Άσκηση 5.4</u>

Δημιουργήστε εφαρμογή lab_5_ask_4 η οποία ορίζει μονοδιάστατο πίνακα 7 χαρακτήρων, ζητά από τον χρήση να εισάγει τα στοιχεία του πίνακα, εκτυπώνει τον πίνακα και στην συνέχεια εξετάζει εάν όλοι οι χαρακτήρες του πίνακα είναι κεφαλαίοι λατινικοί χαρακτήρες.

Υπόδειξη: Μετρήστε πόσους κεφαλαίους λατινικούς χαρακτήρες περιέχει ο πίνακας.

<u>Άσκηση 5.5</u>

Δημιουργήστε εφαρμογή lab_5_ask_5 η οποία ορίζει έναν πίνακα πραγματικών αριθμών 2X3, ζητά από τον χρήστη να εισάγει τα στοιχεία του πίνακα και στην συνέχεια εκτυπώνει:

- τον πίνακα
- το μικρότερο, το μεγαλύτερο και τον μέσο όρο των στοιχείων του πίνακα

<u>Άσκηση 5.6</u>

Δημιουργήστε εφαρμογή lab_5_ask_6 η οποία αποθηκεύει με την χρήση ένθετων εντολών επανάληψης σε πίνακα 5X5 ακεραίων τις παρακάτω τιμές:

1	2	3	4	5	
2	4	6	8	10	
3	6	9	12	15	
4	8	12	16	20	
5	10	15	20	25	

και εκτυπώνει τον πίνακα στην οθόνη του Η/Υ

<u>Άσκηση 5.7</u>

Δημιουργήστε εφαρμογή lab_5_ask_7 η οποία ορίζει έναν δισδιάστατο πίνακα ακεραίων 4X4, ζητά από τον χρήστη να εισάγει τα στοιχεία του πίνακα, εξετάζει εάν τα στοιχεία του πίνακα είναι όλα αρνητικοί αριθμοί και εκτυπώνει με την χρήση εντολών επανάληψης τα αθροίσματα των γραμμών του πίνακα

<u>Άσκηση 5.8</u>

Δημιουργήστε εφαρμογή lab_5_ask_8 η οποία ορίζει ένα τετραγωνικό δισδιάστατο πίνακα NxN (αριθμός γραμμών ίσος με τον αριθμό των στηλών του πίνακα) χαρακτήρων, δίνει την τιμή '*' στα στοιχεία της κύριας και δευτερεύουσας διαγωνίου του πίνακα, την τιμή ' ' (κενός χαρακτήρας) στα υπόλοιπα στοιχεία του πίνακα και εκτυπώνει τον πίνακα στην οθόνη. Η διάσταση N του πίνακα δίνεται από τον χρήστη και το μέγεθος του πίνακα καθορίζεται δυναμικά με την δήλωση ReDim.

6 Παραθυρικές εφαρμογές

6.1 Δημιουργία παραθυρικών εφαρμογών

Η δημιουργία μιας παραθυρικής εφαρμογής γίνεται με την επιλογή «New Project» του μενού «File». Στη συνέχεια εμφανίζεται ένα παράθυρο διαλόγου με τίτλο «New Project» (εικόνα 6.1) όπου πρέπει να επιλεγεί ο τύπος εφαρμογής «Windows Application» και να καθορισθεί το όνομα της εφαρμογής. Η δημιουργία της εφαρμογής ολοκληρώνεται με το πάτημα του πλήκτρο «OK».

New Project						? X
Templates:						0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Visual Studio i	nstalled templat	es				
■VB	₩					
Windows Application	Class Library	Console Application	My Movie Collecti	Screen Saver Starter Kit		
My Templates	5					
Search Online Templates						
A project for great		uith a Wiedows us	or interface			
A project for creat	ing an application v	viuri a vvinuuvvs us	erinteriate		1	
<u>N</u> ame:	lab_6_ask_1					
					ОК	Cancel

Εικόνα 6.1 Δημιουργία παραθυρικής εφαρμογής

Η παραθυρική εφαρμογή (εικόνα 6.2) αρχικά αποτελείται από ένα κενό παράθυρο χρήστη με όνομα και τίτλο Form1.

Με την χρήση του παραθύρου ιδιοτήτων (κάτω δεξιό τμήμα του περιβάλλοντος ανάπτυξης) μπορεί να γίνει η αλλαγή των ιδιοτήτων του παραθύρου. Το παράθυρο ιδιοτήτων μπορεί να εμφανισθεί με την επιλογή «Properties Window» από το μενού «View» ή το πλήκτρο συντόμευσης F4.

Βασικές ιδιότητες, μέθοδοι και συμβάντα του παραθύρου χρήστη είναι:

Ιδιότητα	Περιγραφή	Παράδειγμα	
Name	Το όνομα του παραθύρου	FrmFirstApp	
Text	Ο τίτλος του παραθύρου	Η πρώτη εφαρμογή	
		μου	
BackColor	Το χρώμα υπόβαθρου		
Size	Οι διαστάσεις (πλάτος και ύψος) του παράθυρου	474;303	
Μέθοδος	Περιγραφή	Παράδειγμα	
Close	Κλείσιμο του παραθύρου	frmFirstApp.Close()	
Hide	Απόκρυψη του παραθύρου	frmFirstApp.Hide()	
Show	Εμφάνιση του παραθύρου	frmFirstApp.Show()	
Συμβάν	Περιγραφή	Παράδειγμα	
Load	Φόρτωση του παραθύρου. Το συμβάν αυτό	FrmFirstApp_Load	
	ενεργοποιείται κατά την φόρτωση του παράθυρου.		

Οι διαστάσεις του παραθύρου μπορούν να αλλάξουν και με την χρήση του ποντικιού. Η αλλαγή του μεγέθους του παραθύρου γίνεται με την τοποθέτηση του δείκτη του ποντικιού στην κάτω αριστερή

γωνία του παραθύρου οπότε και η εμφάνιση (εικόνα) του δείκτη του ποντικιού αλλάζει σε διπλό βέλος. Με το σταθερό πάτημα του αριστερού πλήκτρου του ποντικιού και την ταυτόχρονη μετακίνηση του ποντικιού, μπορεί να μεταβληθεί το μέγεθος του παραθύρου.

🖳 lab_6_ask_1 - Microsoft Visual Basic 2005 Express Edition					_ <u>-</u> ×
<u>File Edit View Project Build Debug Data Tools Window Com</u>	munity <u>H</u> elp				
Image: A market and A marke	■ • • • • • • • • • • • • • • • • • • •	2 2 3 3	* 🖧 ₌		
Form1.vb[Design]					Solution Explorer Solution
Error List O Errors ① 0 Warnings ① 0 Messages Description	File	Line	Column	Project	Solution Explorer Index Form1.vb File Properties Build Action Compile Copy to Output I Do not copy Custom Tool I Custom Tool I Custom Tool I Custom Tool I Build Action Bui
Fror List To Index Results for Int function Immediate Window Output Ready	1				How the file relates to the build and deployment processes.

Εικόνα 6.2 Αρχική παραθυρική εφαρμογή

Οι ιδιότητες ενός παραθύρου, εκτός από τον ορισμός τους κατά την διάρκεια σχεδίασης της εφαρμογής, μπορούν να ορισθούν δυναμικά κατά την διάρκεια της εκτέλεσης της εφαρμογής με την χρήση εντολών. Για την απόδοση τιμής σε κάποια ιδιότητα ενός παραθύρου χρησιμοποιείται η λέξη «Me» για να δηλώσει ότι η ιδιότητα αυτή ανήκει στο παράθυρο (και όχι σε κάποιο άλλο στοιχείο ελέγχου ή αντικείμενο του παραθύρου) και το όνομα της ιδιότητας. Με την παρακάτω εντολή ορίζεται δυναμικά ο τίτλος ενός παραθύρου δίνοντας την τιμή «Η πρώτη εφαρμογή μου» στην ιδιότητα Text του παραθύρου.

Me.Text = "Η πρώτη εφαρμογή μου"

6.2 Σχεδιασμός του παραθύρου χρήστη

Ο σχεδιασμός ενός παραθύρου γίνεται με την προσθήκη στοιχείων ελέγχου στο παράθυρο, τον καθορισμό των ιδιοτήτων τους και την διασύνδεση των διαφόρων συμβάντων του παραθύρου και των στοιχείων ελέγχου του με μεθόδους οι οποίες εκτελούνται για το κάθε συμβάν.

Στοιχείο Ελέγχου	Περιγραφή
Ετικέτα	Στοιχείο ελέγχου το οποίο απεικονίζει κείμενο (χωρίς
(Label)	δυνατότητα αλλαγής από τον χρήστη) ή εικόνα
Πλαίσιο κειμένου	Στοιχείο ελέγχου το οποίο χρησιμοποιείται για την εισαγωγή
(TextBox)	κειμένου από τον χρήστη
Πλήκτρο επιλογής	Πλήκτρο με το οποίο ο χρήστης μπορεί να κάνει κάποια επιλογή
(Radio Button)	

Τα βασικά στοιχεία ελέγχου ενός παραθύρου είναι:

Πλαίσιο επιλογής	Πλήκτρο με το οποίο ο χρήστης μπορεί να κάνει κάποια επιλογή		
(CheckBox)	(χρησιμοποιείται συνήθως σε ερωτήσεις τύπου ΝΑΙ/ΟΧΙ)		
Κατάλογος Επιλογής	Στοιχείο ελέγχου το οποίο απεικονίζει έναν κατάλογο		
(List Box)	αντικείμενων. Ο χρήστης μπορεί να επιλέξει ένα η περισσότερα		
	αντικείμενα.		
Αναδιπλούμενος Κατάλογος Αναδιπλούμενος κατάλογος επιλογής.			
Επιλογής (ComboBox)			
Πλαίσιο στοιχείων ελέγχου	Πλαίσιο στο οποίο μπορούν να τοποθετηθούν και να		
(Panel)	ομαδοποιηθούν άλλα στοιχεία ελέγχου		
Πλήκτρο	Πλήκτρο το οποίο χρησιμοποιείται για την κλήση κάποιου		
(Button)	υποπρογράμματος ή μεθόδου		



Εικόνα 6.3 Εργαλειοθήκη – Στοιχεία Ελέγχου

Η επιλογή στοιχείων ελέγχου και η τοποθέτηση τους στο παράθυρο χρήστη γίνεται με την χρήση της εργαλειοθήκης (Mevoú «View» επιλογή «ToolBox» ή συντόμευση Ctrl-Alt-X). Τα συνηθέστερα στοιχεία ελέγχου μπορούν να επιλεχθούν από την κατηγορία Common Controls.

Ετικέτα

Η ετικέτα απεικονίζει κείμενο (χωρίς δυνατότητα αλλαγής από τον χρήστη) ή εικόνα. Η ετικέτα χρησιμοποιείται κυρίως σε συνδυασμό με άλλα στοιχεία ελέγχου για να δηλώσει τον τίτλο τους ή να επεξηγήσει το περιεχόμενο τους ή την λειτουργία τους.

Στο παρακάτω παράδειγμα χρησιμοποιούνται ετικέτες σε συνδυασμό με πλαίσια κειμένου για την εισαγωγή του ονόματος, του επιθέτου, της διεύθυνσης ηλεκτρονικού ταχυδρομείου και της διεύθυνσης ενός σπουδαστή.

🔡 Παράδειγμ	α στοιχείων ελέγχου	<u> </u>
Όνομα	Γεώργιος	
Επίθετο	Γεωργίου	
Email	george@teilam.gr	
Διεύθυνση	θερμοπυλών 10 35100 Λαμία	A P

Εικόνα 6.4 Ετικέτες και πλαίσια κειμένου

Τα κυριότερα χαρακτηριστικά μίας ετικέτας είναι:

Ιδιότητα	Περιγραφή	Παράδειγμα
Name	Το όνομα της ετικέτας	LabelSurname
Text	Κείμενο το οποίο εμφανίζεται (Τίτλος)	Επίθετο
TextAlign	Στοίχιση του κείμενου	Left, Right ή Center
Font	Γραμματοσειρά	Microsoft Sans Serif 8.25pt
BackColor	Το χρώμα υπόβαθρου	
BorderStyle	Τύπος πλαισίου	Fixed3D, Fixed Single ή None
Image	Εικόνα η οποία εμφανίζεται	
Visible	Ιδιότητα η οποία καθορίζει εάν το στοιχείο	True ή False
	ελέγχου εμφανίζεται στο παράθυρο	(ορατό ή κρυφό)

Πλαίσιο κειμένου

Το πλαίσιο κειμένου χρησιμοποιείται για την εισαγωγή κειμένου από τον χρήστη. Οι κυριότερες ιδιότητες ενός πλαισίου κειμένου είναι:

Ιδιότητα	Περιγραφή	Παράδειγμα
Name	Το όνομα του πλαισίου κειμένου	TextBoxSurname
Text	Κείμενο το οποίο εμφανίζεται ή έχει εισαχθεί	
	από τον χρήστη	
TextAlign	Στοίχιση του κείμενου	Left, Right ή Center
BackColor	Το χρώμα υπόβαθρου	
BorderStyle	Τύπος πλαισίου	Fixed3D, Fixed Single ή None
Font	Γραμματοσειρά	Microsoft Sans Serif 8.25pt
Visible	Ιδιότητα η οποία καθορίζει εάν το στοιχείο	True ή False
	ελέγχου εμφανίζεται στο παράθυρο	(ορατό ή κρυφό)
Multiline	Καθορίζει εάν το κείμενο εκτείνεται σε	True ή False
	περισσότερες από μία γραμμές	
ScrollBars	Καθορίζει εάν το πλαίσιο κειμένου έχει	None ή
	οριζόντιες ή κάθετες μπάρες κύλισης	Horizontal ή
		Vertical ή
		Both
Συμβάν	Περιγραφή	Παράδειγμα
TextChanged	Αλλαγή του κειμένου	TextBoxSurname_TextChanged

Με την χρήση της ιδιότητας Text μπορούμε να τροποποιήσουμε το κείμενο ενός πλαισίου κειμένου η να διαβάσουμε το κείμενο το οποίο εισήγαγε ο χρήστης. Για παράδειγμα, η εντολή

```
TextBoxSurName.Text = ""
```

καθαρίζει το κείμενο του πλαίσιο κειμένου του επιθέτου του σπουδαστή. Το διάβασμα του επιθέτου το οποίο πληκτρολογήθηκε από τον χρήστη και η αποθήκευση του σε μεταβλητή με το όνομα strSurname γίνεται με τις παρακάτω εντολές

```
Dim strSurname As String
strSurname = TextBoxSurname.Text
```

Πλήκτρα και πλαίσια επιλογής

Το πλήκτρα επιλογής (RadioButtons) χρησιμοποιούνται σε επιλογές μεταξύ εναλλακτικών απαντήσεων. Στο παρακάτω παράδειγμα χρησιμοποιούνται δύο πλήκτρα επιλογής για την επιλογή του φύλο του σπουδαστή.

Τα πλαίσια επιλογής (CheckBoxes) χρησιμοποιούνται συνήθως σε ερωτήσεις τύπου NAI/OXI. Στο παρακάτω παράδειγμα το πλήκτρο επιλογής χρησιμοποιείται για την εισαγωγή της πληροφορίας εάν ο σπουδαστής επιθυμεί να ενημερώνεται με ηλεκτρονικό ταχυδρομείο ή όχι.

🔡 Παράδειγμ	α στοι	ε νωί з	λέγχου	
Όνομα				
Επίθετο				
Email				
Διεύθυνση				×
Φύλο	• A	Св	Επιθυμεί να ενημερώνεται με	email 🗖

Εικόνα 6.5 Πλήκτρα και πλαίσια επιλογής

Κύριες ιδιότητες ενός πλήκτρου επιλογής είναι:

Ιδιότητα	Περιγραφή	Παράδειγμα
Name	Το όνομα του πλήκτρου επιλογής	RadioButtonMale
Text	Κείμενο το οποίο εμφανίζεται	Α
Font	Γραμματοσειρά	Microsoft Sans Serif 8.25pt
BackColor	Το χρώμα υπόβαθρου	
RightToLeft	Καθορίζει εάν το κείμενο	True ή False
	εμφανίζεται δεξιά του πλήκτρο	
Checked	Ιδιότητα η οποία καθορίζει εάν το	True ή False
	πλήκτρο επιλογής είναι επιλεγμένο	
Συμβάν	Περιγραφή	Παράδειγμα
CheckedChanged	Αλλαγή επιλογής	RadioButtonMale CheckedChanged

Η βασικότερη ιδιότητα το πλήκτρου επιλογής είναι η ιδιότητα Checked. Εάν κατά την σχεδίαση του παραθύρου δώσουμε την τιμή True στην ιδιότητα Checked, τότε η το πλήκτρο αυτό γίνεται η προεπιλεγμένη επιλογή. Κατά την διάρκεια της εκτέλεσης του προγράμματος με τον έλεγχο της τιμής της ιδιότητα αυτής προσδιορίζεται εάν ο χρήστης επέλεξε το πλήκτρο αυτό.

Στην περίπτωση όπου χρησιμοποιούνται περισσότερες από μία ομάδες επιλογών στο ίδιο παράθυρο (π.χ. επιλογή φύλου και επιλογή αγαπημένου αθλήματος) τότε τα πλήκτρα επιλογής πρέπει να τοποθετηθούν πάνω σε διαφορετικά πλαίσια στοιχείων ελέγχου.

Ιδιότητα	Περιγραφή	Παράδειγμα
Name	Το όνομα του πλήκτρου επιλογής	CheckBoxMale
Text	Κείμενο το οποίο εμφανίζεται	Α
Font	Γραμματοσειρά	Microsoft Sans Serif 8.25pt
BackColor	Το χρώμα υπόβαθρου	
RightToLeft	Καθορίζει εάν το κείμενο εμφανίζεται δεξιά του πλασίου	True ή False
Checked	Ιδιότητα η οποία καθορίζει εάν το πλαίσιο επιλογής είναι επιλεγμένο	True ή False (Ναι/Οχι)
ThreeState	Ιδιότητα η οποία καθορίζει εάν το πλαίσιο επιλογής έχει τρεις καταστάσεις: επιλεγμένο, μη επιλεγμένο και μη καθορισμένο	True ή False
CheckedState	Η ιδιότητα αυτή καθορίζει την κατάσταση του πλαισίου επιλογής στην περίπτωση που έχουν επιλεχθεί τρεις καταστάσεις για το πλαίσιο επιλογής (ThreeState=True)	Checked ή Unchecked ή Indeterminate (Ναι/Οχι/Άγνωστο)
Συμβάν	Περιγραφή	Παράδειγμα
CheckedChanged	Αλλαγή επιλογής	CheckBoxMale_CheckedChanged

Κύριες ιδιότητες ενός πλαισίου επιλογής είναι:

Το πλαίσιο επιλογής έχει συνήθως δύο καταστάσεις: επιλεγμένο (Checked=true) και μη επιλεγμένο (Checked = false). Με την χρήση της ιδιότητας Checked κατά την διάρκεια της σχεδίασης καθορίζεται η προεπιλεγμένη κατάσταση και κατά την διάρκεια της εκτέλεσης ελέγχεται η επιλογή του χρήστη.

Με την χρήση της ιδιότητας ThreeState είναι δυνατό να ορισθεί μια επιπλέον ενδιάμεση κατάσταση όπου ο χρήστης δεν έχει κάνει καμία επιλογή. Για την χρήση και της τρίτης κατάστασης η τιμή της ιδιότητας ThreeState πρέπει να είναι true. Ο έλεγχος της προεπιλεγμένης ή τρέχουσας κατάστασης γίνεται με την ιδιότητα CheckedState η οποία έχει ως δυνατές τιμές checked, unchecked και indeterminate.

Κατάλογος επιλογής και αναδιπλούμενος κατάλογος

Με τον κατάλογο επιλογής (ListBox) και το αναδιπλούμενο κατάλογο επιλογής (ComboBox) δίνεται η δυνατότητα στον χρήστη της επιλογής ενός ή περισσοτέρων αντικειμένων από έναν κατάλογο.

Στο παρακάτω παράδειγμα χρησιμοποιείται ένας κατάλογος επιλογής για τον καθορισμό του τμήματος του σπουδαστή και ένας αναδιπλούμενος κατάλογος επιλογής για τον καθορισμό του εξαμήνου σπουδών του.

🔡 Παράδειγμ	α στοιχείων ελέγχου
Όνομα	Γεώργιος
Επίθετο	Γεωργίου
Email	george@teilam.gr
Διεύθυνση	Θερμοπυλών 30 35 100 Λαμία
обуФ	💿 Α 🔘 Θ 🛛 Επιθυμεί να ενημερώνεται με email 🗹
Τμήμα	Ηπεκτρονικής Ηπεκτροπογίας Ππηροφορικής

Εικόνα 6.6 Κατάλογος επιλογής και αναδιπλούμενος κατάλογος

Οι βασικότερες ιδιότητες του καταλόγου επιλογής είναι:

Ιδιότητα	Περιγραφή	Παράδειγμα
Name	Το όνομα του καταλόγου επιλογής	ListBoxDepartment
Font	Γραμματοσειρά	Microsoft Sans Serif 8.25pt
BackColor	Το χρώμα υπόβαθρου	
BorderStyle	Τύπος πλαισίου	Fixed3D, Fixed Single ή None
RightToLeft	Καθορίζει την στοίχιση των	True ή False
	αντικειμένων	
SelectionMode	Ιδιότητα η οποία καθορίζει τον	None ή
	αριθμό των επιλέξιμων	One ή
	αντικειμένων	MultiSimple ή
		MultiExtended
Items	Συλλογή (Collection) η οποία	Ηλεκτρονικής
	περιέχει τα αντικείμενα του	Ηλεκτρολογίας
	καταλόγου	Πληροφορικής
Sorted	Η ιδιότητα αυτή καθορίζει εάν τα	True ή False
	αντικείμενα ταξινομούνται	
	αλφαβητικά	
SelectedIndex	Ιδιότητα ή οποία καθορίζει ή	ListBoxDepartment.SelectedIndex
	επιστρέφει τον δείκτη του	
	επιλεγμένου αντικείμενο. Η	
	ιδιότητα αυτή δεν είναι διαθέσιμη	
	κατά την σχεδίαση της εφαρμογής.	
SelectedItem	Επιστρέφει το επιλεγμένο	ListBoxDepartment.SelectedItem
	αντικείμενο. Η ιδιότητα αυτή δεν	
	είναι διαθέσιμη κατά την σχεδίαση	
	της εφαρμογής.	
SelectedItems	Επιστρέφει μια συλλογή	ListBoxDepartment.SelectedItems
	(Collection) με τα επιλεγμένα	
	αντικείμενα του καταλόγου	
SelectedIndexes	Επιστρέφει μια συλλογή	ListBoxDepartment.SelectedIndexes
	(Collection) με τους επιλεγμένους	
	δείκτες των αντικειμένων του	

	καταλόγου	
Μέθοδοι	Περιγραφή	Παράδειγμα
GetSelected	Δέχεται ως παράμετρο εισόδου τον δείκτη του αντικειμένου και επιστρέφει True ή False ανάλογα εάν το αντικείμενο είναι επιλεγμένο ή όχι	GetSelected(0)
SetSelected	Δέχεται ως πρώτη παράμετρο τον δείκτη του αντικειμένου και ως δεύτερη True ή False ή οποία καθορίζει εάν το αντικέιμενο είναι επιλεγμένο ή μη όχι	SetSelected(0,False) (Επιλέγει το πρώτο αντικείμενο του καταλόγου)
Συμβάν	Περιγραφή	Παράδειγμα
SelectedIndexChanged	Αλλαγή επιλογής	ListBoxDepartment SelectedIndexChanged

Στην περίπτωση που η ιδιότητα SelectionMode έχει την τιμή None, ο χρήστης δεν έχει την δυνατότητα επιλογής αντικειμένων. Όταν έχει την τιμή One, μπορεί να επιλέξει μόνο ένα αντικείμενο, όταν έχει την τιμή MultiSimple μπορεί να επιλέξει περισσότερα από ένα αντικείμενα και όταν έχει την τιμή MultiExtended μπορεί με το πάτημα του πλήκτρου Shift και την ταυτόχρονη χρήση των πλήκτρων μετακίνησης του πληκτρολογίου(πάνω ή κάτω βέλος) ή την χρήση του ποντικιού να επιλέξει περιοχές αντικειμένων.

Οι ιδιότητες SelectedIndex και SelectedItem καθορίζουν ή επιστρέφουν τον δείκτη του επιλεγμένου αντικειμένου και το επιλεγμένο αντικείμενο του καταλόγου αντίστοιχα. Οι ιδιότητες αυτές δεν είναι διαθέσιμες κατά την σχεδίαση της εφαρμογής και βρίσκουν χρήση σε επιλογές ενός μόνο αντικειμένου (SelectionMode=One).

Για παράδειγμα η παρακάτω εντολή προεπιλέγει το πρώτο αντικείμενο του καταλόγου

Me.ListBoxDepartment.SelectedIndex = 0

και η παρακάτω εντολή αποθηκεύει το επιλεγμένο αντικείμενο στην μεταβλητή strDepartment.

Dim strDepartment As String = Me.ListBoxDepartment.SelectedItem

Εάν δεν έχεο επιλεγεί κανένα αντικείμενο του καταλόγου επιλογής, τότε η τιμή της ιδιότητας SelectedIndex είναι ίση με -1. Η παρακάτω εντολή ελέγχει εάν έχει επιλεγεί κάποιο αντικείμενο από τον κατάλογο επιλογής:

If Me.ListBoxDepartment.SelectedIndex = -1 then ...

Για την επιλογή πολλαπλών αντικειμένων χρησιμοποιούνται οι ιδιότητες SelectedItems και SelectedIndexes οι οποίες επιστρέφουν μια συλλογή (Collection) αντικειμένων και δεικτών αντικειμένων αντίστοιχα. Μία συλλογή είναι ένας πίνακα αντικειμένων ο οποίος έχει την δυνατότητα δυναμικής προσθήκης και διαγραφής αντικειμένων.

Οι κυριότερες μέθοδοι και ιδιότητες συλλογών αντικειμένων είναι :

- Add με την οποία μπορούμε να προσθέσουμε ένα νέο αντικείμενο στην συλλογή
- Clear με την οποία μπορούμε να διαγράψουμε όλα τα στοιχεία της συλλογής
- Remove με την οποία μπορούμε να αφαιρέσουμε από την συλλογή ένα συγκεκριμένο αντικείμενο το οποίο δίνεται ως παράμετρος της μεθόδου.
 - πχ Me.ListBoxDepartment.Items.Remove("Ηλεκτρολογίας")
- Count Ιδιότητα η οποία επιστρέφει τον αριθμό των στοιχείων της συλλογής

Η πρόσβαση στα στοιχεία μιας συλλογής μπορεί να γίνει με την χρήση δεικτών όπως γίνεται και με τους πίνακες.

Τα στοιχεία ενός καταλόγου επιλογής μπορούν να καθορισθούν κατά την σχεδίαση της εφαρμογής ή να καθορισθούν δυναμικά κατά την διάρκεια εκτέλεσης της εφαρμογής με την χρήση της ιδιότητας

Items. Το παρακάτω παράδειγμα απαλείφει όλα τα στοιχεία ενός καταλόγου, προσθέτει δυναμικά τρία αντικείμενα στον κατάλογο και προεπιλέγει το πρώτο αντικείμενο.

```
'Καθαρισμός καταλόγου
Me.ListBoxDepartment.Items.Clear()
'Δυναμική προσθήκη στοιχείων στον κατάλογο
Me.ListBoxDepartment.Items.Add("Ηλεκτρονικής")
Me.ListBoxDepartment.Items.Add("Πληροφορικής")
'Προεπιλόγή τμήματος Ηλεκτρονικής
Me.ListBoxDepartment.SetSelected(0, True)
```

Παράδειγμα 6.1 Δυναμικός καθορισμός των στοιχείων καταλόγου επιλογής

Οι βασικότερες ιδιότητες του αναδιπλούμενου καταλόγου επιλογής είναι:

Ιδιότητα	Περιγραφή	Παράδειγμα
Name	Το όνομα του αναδιπλούμενου	ComboBoxSemester
	καταλόγου επιλογής	
Font	Γραμματοσειρά	Microsoft Sans Serif 8.25pt
BackColor	Το χρώμα υπόβαθρου	
RightToLeft	Καθορίζει την στοίχιση των	True ή False
	αντικειμένων	
DropDownStyle	Ιδιότητα η οποία καθορίζει τον	Simple ή
	τρόπο με τον οποίο ο χρήστης	DropDown ή
	επιλέγει αντικείμενα	DropDownList
Items	Τα αντικείμενα του καταλόγου	A B Γ Δ Ε ΣΤ Η και Ζ
		(γραμμένα το καθένα σε διαφορετική
		γραμμή)
Sorted	Η ιδιότητα αυτή καθορίζει εάν	True ή False
	τα αντικείμενα ταξινομούνται	
	αλφαβητικά	
SelectedIndex	Ιδιότητα ή οποία καθορίζει ή	ComboBoxSemester.SelectedIndex()
	επιστρέφει το δείκτη του	
	επιλεγμένου αντικείμενο. Η	
	ιδιότητα αυτή δεν είναι	
	διαθέσιμη κατά την σχεδίαση	
	της εφαρμογής.	
SelectedItem	Επιστρέφει το επιλεγμένο	ComboBoxSemester.SelectedItem()
	αντικείμενο. Η ιδιότητα αυτή	
	δεν είναι διαθέσιμη κατά την	
	σχεδίαση της εφαρμογής.	
Text	Καθορίζει ή επιστρέφει το	ComboBoxSemester.Text
	κείμενο το οποίο εμφανίζεται	
	στον αναδιπλούμενο κατάλογο	
Συμβάν	Περιγραφή	Παράδειγμα
SelectedIndexChanged	Αλλαγή επιλογής	ComboBoxSemester
		_SelectedIndexChanged

Ο αναδιπλούμενος κατάλογος χρησιμοποιείται για την επιλογή ενός μόνο αντικειμένου από τον χρήστη. Εάν δεν έχει επιλεγεί κάποιο αντικείμενο του καταλόγου τότε το εμφανιζόμενο κείμενο (ιδιότητα Text) είναι κενό. Εάν προεπιλεγεί κάποια αντικείμενο ή ο χρήστης επιλέξει κάποιο αντικείμενο από τον κατάλογο, τότε το αντικείμενο αυτό εμφανίζεται ως το κείμενο του αναδιπλούμενου κατάλογου.

Ανάλογα με την τιμή της ιδιότητας DropDownStyle διαφέρει και ο τρόπος επιλογής αντικειμένων. Εάν έχει επιλεγεί ως DropDownStyle η τιμή SimpleCombo, ο χρήστης επιλέγει κάποιο αντικείμενο με την εισαγωγή κειμένου ή με την χρήση των πλήκτρων (πάνω και κάτω βέλος του πληκτρολογίου). Εάν έχει επιλεγεί η τιμή DropDown, ο χρήστης μπορεί να αναπτύξει και να αναδιπλώσει τον κατάλογο αντικειμένων και να επιλέξει κάποιο στοιχείο του καταλόγου με την χρήση του ποντικιού. Τέλος, εάν έχει επιλεγεί η τιμή DropDownList, ο χρήστης δεν έχει την δυνατότητα ελεύθερης εισαγωγής κειμένου αλλά πρέπει αναγκαστικά να επιλέξει κάποιο από τα υπάρχοντα αντικείμενα του καταλόγου.

Οι ιδιότητες SelectedIndex και SelectedItem είναι αντίστοιχες με τις ιδιότητες του καταλόγου επιλογής. Για παράδειγμα η παρακάτω εντολή προεπιλέγει το πρώτο αντικείμενο ως τιμή του αναδιπλούμενου καταλόγου

```
Me.ComboBoxSemester.SelectedIndex = 0
```

και η παρακάτω εντολή αποθηκεύει το επιλεγμένο αντικείμενο στην μεταβλητή strSemester.

Dim strSemester As String = Me.ComboBoxSemester.SelectedItem

Εναλλακτικά μπορεί να χρησιμοποιηθεί η ιδιότητα Text.

Dim strSemester As String = Me.ComboBoxSemester.Text

Για την αποεπιλογή του επιλεγμένου αντικειμένου μπορεί να χρησιμοποιηθεί η εντολή:

```
Me.ComboBoxSemester.SelectedIndex = -1
```

Πλαίσιο στοιχείων ελέγχου

Το πλαίσιο στοιχείων ελέγχου (Panel) μπορεί να επιλεγεί από την κατηγορία Containers της εργαλειοθήκης. Το πλαίσιο στοιχείων ελέγχου μπορεί να χρησιμοποιηθεί για την τοποθέτηση και ομαδοποίηση άλλων στοιχείων ελέγχου εντός του πλαισίου. Η ομαδοποίηση αυτή διευκολύνει τον σχεδιασμό ενός παραθύρου, την ομαδική μετακίνηση των στοιχείων ελέγχου που περιέχει και την ομαδοποίηση πλήκτρων επιλογής όταν απαιτείται να διαχωριστούν σε διαφορετικές ομάδες επιλογής.

Στο παρακάτω παράδειγμα χρησιμοποιούνται δύο πλαίσια στοιχείων ελέγχου, το πρώτο για να ομαδοποιήσει όλα τα στοιχεία ελέγχου τα οποία είναι σχετικά με την εισαγωγή των στοιχείων ενός σπουδαστή και το δεύτερο για να ομαδοποιήσει τα πλήκτρα επιλογής για τον καθορισμό του φύλου του σπουδαστή.

🔛 Παράδειγμ	α στοιχείων ελέγχου
Ονομα	Γεώργιος
Επίθετο	Γεωργίου
Email	george@teilam.gr
Διεύθυνση	Θερμοπυλών 30 35 100 Λαμία
οΓύΦ	Eπιθυμεί να ενημερώνεται με email
Τμήμα	Ηλεκτρονικής Ηλεκτρολογίας Πληροφορικής

Εικόνα 6.7 Πλαίσια στοιχείων ελέγχου

Τα κυριότερα χαρακτηριστικά ενός πλαισίου στοιχείων ελέγχου είναι:

Ιδιότητα	Περιγραφή	Παράδειγμα
Name	Το όνομα του πλαισίου στοιχείων ελέγχου	panelStudentData
BackColor	Το χρώμα υπόβαθρου	
BorderStyle	Τύπος πλαισίου	Fixed3D, Fixed Single ή
		None
Visible	Ιδιότητα η οποία καθορίζει εάν το στοιχείο ελέγχου	True ή False
	εμφανιζεται στο παραθυρο	

Πλήκτρο

Τα πλήκτρα (Buttons) δίνουν την δυνατότητα στον χρήστη επιλογής ενεργειών οι οποίες εκτελούν κάποιο υποπρόγραμμα, τροποποιούν ιδιότητες αντικειμένων ή καλούν μεθόδους αντικειμένων.

Στο παράδειγμα 6.8 χρησιμοποιούνται τρία πλήκτρα τα οποία δίνουν την δυνατότητα στο χρήστη να καθαρίσει τα δεδομένα του παραθύρου, να εμφανίσει τα στοιχεία τα οποία έχει εισάγει και να κλείσει το παράθυρο της εφαρμογής.

🔛 Παράδειγμ	α στοιχείων ελέγχου
Ονομα	Γεώργιος
Επίθετο	Γεωργίου
Email	george@teilam.gr
Διεύθυνση	Βερμοπυλών 30 35 100 Λαμία
οĥὺΦ	● A ● θ En:θυμεί να ενημερώνεται με email
Τμήμα	Η <mark>λεκτρονικής</mark> Εξάμηνο Α Ηλεκτρολογίας Πληροφορικής
Καθαρισμό	ς Εμφάνιση ΟΚ

Εικόνα 6.8 Πλήκτρα

Τα κυριότερα χαρακτηριστικά ενός πλήκτρου είναι:

Ιδιότητα	Περιγραφή	Παράδειγμα
Name	Το όνομα του πλήκτρου	ButtonClear
BackColor	Το χρώμα υπόβαθρου	
Text	Ο τίτλος του πλήκτρου	Καθαρισμός
Visible	Ιδιότητα η οποία καθορίζει εάν το πλήκτρο	True ή False
	εμφανίζεται στο παράθυρο	(ορατό ή κρυφό)
Enabled	Ενεργοποιεί ή απενεργοποιεί (δεν είναι δυνατή η	True ή False
	επιλογή του από τον χρήστη) ένα πλήκτρο	
Συμβάν	Περιγραφή	Παράδειγμα
Click	Πάτημα του πλήκτρου από το χρήστη	ButtonClear_Click

Βασικό χαρακτηριστικό ενός πλήκτρου είναι το συμβάν Click το οποίο προκαλείται όταν πατηθεί το πλήκτρο από τον χρήστη.

6.3 Διασύνδεση συμβάντων ενός παραθύρου

Η υλοποίηση της λειτουργικότητας ενός παραθύρου γίνεται με την διασύνδεση συμβάντων του παραθύρου ή των στοιχείων ελέγχου του με την κλήση μεθόδων του παραθύρου.

Σε εφαρμογές του τερματικού (Console Application) η εκτέλεση του προγράμματος ξεκινά με την κλήση της διαδικασίας main. Αντίθετα σε μία παραθυρική εφαρμογή, η εκτέλεση του προγράμματος συνήθως ξεκινά με την εμφάνιση ενός αρχικού παράθυρου της εφαρμογής. Τα παράθυρο εκκίνησης καθορίζεται με την χρήση του παραθύρου ιδιοτήτων της εφαρμογής.

Σε μία παραθυρική εφαρμογή, η επικοινωνία με τον χρήστη γίνεται με το πάτημα πλήκτρων της εφαρμογής, την επιλογή εντολών από τα μενού της εφαρμογής και την αλληλεπίδραση γενικότερα με τα παράθυρα και τα στοιχεία ελέγχου της εφαρμογής. Οι ενέργειες του χρήστη προκαλούν συμβάντα τα οποία με την σειρά τους προκαλούν την εκτέλεση μεθόδων οι οποίες είναι συνδεδεμένες προγραμματιστικά με τα συμβάντα αυτά.

Το παράθυρο του παραδείγματος 6.8 χρησιμοποιεί τα παρακάτω συμβάντα:

- Load για το παράθυρο της εφαρμογής FormStudentData
- Click για το πλήκτρο ButtonClear (Καθαρισμός)
- Click για το πλήκτρο ButtonShow (Εμφάνιση)
- Click $\gamma_{1\alpha}$ to $\pi\lambda\eta\kappa\tau\rho_0$ ButtonOK $\mu\epsilon$ (OK)

	Properties	×
	FormStudentData System.Windows.Forms.Form	n 🔹
	Enter FortChanged	
	ForeColorChanged	
/	FormClosing	
	GiveFeedback	
	HelpButtonClicked	
	HelpRequested	Εικονίδιο
Εικονίδιο	ImeModeChanged	Συμβάντα
Ιδιότητες	InputLanguageChanged	
	InputLanguageChanging	
	KeyDown	
	KeyPress	
	КеуUр	
	Layout	
	Leave	
	Load FormStudentDa	ta_Load 💽
	LocationChanged	
	Load Occurs whenever the user loads the form.	
	p-	

Εικόνα 6.9 Επιλογή συμβάντων στο παράθυρο ιδιοτήτων

Η αντιστοίχιση συμβάντων με μεθόδους γίνεται με την χρήση του παραθύρου ιδιοτήτων. Με το πάτημα του εικονιδίου συμβάντα (δεξιό τμήμα της εικόνας 6.9) εμφανίζονται τα διαθέσιμα συμβάντα ενός αντικειμένου της εφαρμογής. Η επιστροφή στις ιδιότητες ενός αντικειμένου γίνεται με το εικονίδιο ιδιότητες.

Η διασύνδεση ενός συμβάντος με μία μέθοδο γίνεται με το διπλό πάτημα του αριστερού πλήκτρου του ποντικιού πάνω σε κάποια ιδιότητα. Εναλλακτικά, για τις βασικότερα συμβάντα ενός στοιχείου ελέγχου ή παραθύρου, η διασύνδεση μπορεί να γίνει με το διπλό πάτημα του αριστερού πλήκτρου του ποντικιού πάνω στο στοιχείο ελέγχου ή το παράθυρο της εφαρμογής. Όταν επιλεγεί κάποια συμβάν για διασύνδεση, το περιβάλλον της VisualBasic εμφανίζει το αρχείο κώδικα της VisualBasic το οποίο αντιστοιχεί στο παράθυρο της εφαρμογής και εισάγει αυτόματα τον ορισμό της μεθόδου η οποία διασυνδέεται με το συμβάν (εικόνα 6.10).

Η μέθοδος αρχικά είναι κενή. Ο προγραμματιστής πρέπει να καθορίσει τις εντολές οι οποίες εκτελούνται για κάθε συμβάν. Για το συγκεκριμένο παράδειγμα ορίζονται οι παρακάτω μέθοδοι για τα συμβάντα:

lab_6_	_ask_1 - Mici	osoft Visual Basic 2005 Express Edition			X
Eile Edi	it <u>V</u> iew Pr	sject <u>B</u> uild <u>D</u> ebug D <u>a</u> ta <u>T</u> ools <u>W</u> indow <u>C</u> ommunity <u>H</u> elp			
1	i 📴 🛪 🛃 i	J X 🖻 🛍 📓 🗏 🤊 • (* - * 🗉 💷 🗐 💭 🕾 🖄 👷			
FormSt	udentData.	b* How to: Close Dtain User Input lab_6_ask_1 FormStudentData.vb [Design]* frmFirstApp.vb	₹×	Solution Explorer 🔷 🤘	ł ×
⊘ ∂Butt	tonOK	🔽 🐓 Click	+		
Pub	lic Class	FormStudentData	A	Iab_6_ask_1 My Project FormStudentData.vb FormFirstApp.vb	
G	End Sub	Sub Formstudentbata_Load(byval sender xs system.object, byval e xs system	1. EV		
þ	Private	Sub ButtonClear_Click(ByVal sender & System.Object, ByVal e & System.Ev	rent		
	Private		ntA:	Solution Explorer	
ē	Private End Sub	Sub ButtonOK_Click(ByVal sender & System.Object, ByVal e & System.Event	Arg	Properties I ButtonOK_Click Attributes	+ ×
End	Class			Unit and the second sec	
2		000000000000000000000000000000000000000	•		
Immedi	iate Window] Output			
eady		Ln 17	Col 9	Ch 9 INS	

Εικόνα 6.10 Επιλογή συμβάντων στο παράθυρο ιδιοτήτων

Συμβάν FormStudentData Load

Παράδειγμα 6.2

Συμβάν ButtonClear Click

```
Private Sub ButtonClear_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ButtonClear.Click
'Kαθαρισμός πλαισίων κειμένου
Me.TextBoxName.Text = ""
Me.TextBoxSurname.Text = ""
Me.TextBoxEmail.Text = ""
Me.TextBoxAddress.Text = ""
'Kαθαρισμός πλήκτρων επιλογής
Me.RadioButtonFemale.Checked = False
Me.RadioButtonMale.Checked = False
```



Παράδειγμα 6.3

Συμβάν ButtonOk Click

```
Private Sub ButtonOK_Click(ByVal sender As System.Object, _______
ByVal e As System.EventArgs) Handles ButtonOK.Click
'Τερματισμός εφαρμογής
End
End Sub
```

Παράδειγμα 6.4

6.4 Χρήση πολλαπλών παραθύρων σε μία εφαρμογή

Μια παραθυρική εφαρμογή μπορεί να χρησιμοποιεί περισσότερα του ενός παράθυρα. Η πρόσθεση ενός νέου παράθυρου σε μία εφαρμογή γίνεται με την επιλογή «Add Windows Form» από το μενού «Project». Στο παράθυρο διαλόγου το οποίο εμφανίζεται (εικόνα 6.11) επιλέγεται ως τύπος αντικειμένου «WindowsForm» και στο πλαίσιο κειμένου εισάγεται το όνομα του παράθυρου. Η διαδικασία δημιουργίας τελειώνει με την πάτημα του πλήκτρου «Add».

A	dd New Item - lat	0_6_ask_1						? X
Ī	emplates:							000 0-0- -0-0 000
	Visual Studio ir	nstalled templa	ites					-
		10.00			18		Tida	
	Windows Form	Dialog	Explorer Form	MDI Parent Form	About Box	Login Form	Splash Screen	
	V _B	1						
	Class	Module	DataSet	SQL Database	User Control	Inherited Form	Inherited User Control	
	Text File							-1
A blank Windows Form								
Ŀ	lame:	FormShowData	.vb					
							Add	Tancel

Εικόνα 6.11 Δημιουργία νέου παραθύρου

Παράδειγμα πρόσθεσης ενός νέου παραθύρου σε μία εφαρμογή είναι το παράθυρο με όνομα FormShowData (εικόνα 6.12). Το παράθυρο αυτό χρησιμοποιείται για την εμφάνιση των στοιχείων ενός σπουδαστή και αποτελείται από ένα πλαίσιο κειμένου με όνομα TextBoxData και ένα πλήκτρο με όνομα ButtonOK. Το πλαίσιο κειμένου εμφανίζει τα στοιχεία του σπουδαστή τα οποία εισήγαγε ο χρήστης και το πλήκτρο ButtonOK χρησιμοποιείται για το κλείσιμο του παραθύρου.

🔛 Στοιχεία σπουδασή 📃 🗖 🗙							
Όνομα	Γεώργιος						
Επώνυμο	Γεωργίου						
Email	george@teilam.gr						
Φύλο	A						
Ενημέρωση	NAI						
Τμήμα	Ηλεκτρονικής						
Εξάμηνο	A						
1							
	ок						

Εικόνα 6.12 Το παράθυρο εμφάνισης των στοιχείων του σπουδαστή

Η ιδιότητα Multiline του πλαισίου κειμένου TextBoxData έχει την τιμή True και η ιδιότητα του Font έχει την τιμή Courier New; 9,75pt.

Το πλήκτρο ButtonOK ορίζει το συμβάν Click το οποίο διασυνδέεται με την παρακάτω μέθοδο η οποία κλείνει το παράθυρο FormShowData.

```
Private Sub ButtonOK_Click(ByVal sender As System.Object, ______
ByVal e As System.EventArgs) Handles ButtonOK.Click
Me.Close()
End Sub
```

Παράδειγμα 6.5

Το παράθυρο FormShowData εμφανίζεται με το πάτημα του πλήκτρου ButtonShowData του παράθυρου FormStudentData. Το πλήκτο διασυνδέεται με την παρακάτω μέθοδο η οποία αρχικά εμφανίζει το παράθυρο FormShowData, αποθηκεύει τα στοιχεία του σπουδαστή σε μια μεταβλητή με όνομα strData και τέλος εμφανίζει τα δεδομένα αυτά στο πλαίσιο TextBoxData του παραθύρου FormShowData.

```
Private Sub ButtonShow Click (ByVal sender As System.Object,
                  ByVal e As System. EventArgs) Handles ButtonShow. Click
    'Εμφάνιση παραθύρου στοιχείων σπουδαστή
    FormShowData.Show()
    'Στοιχεία πλαισίων κειμένου
    Dim strName As String = Me.TextBoxName.Text
    Dim strSurname As String = Me.TextBoxSurname.Text
    Dim strEmail As String = Me.TextBoxEmail.Text
    'Στοιχεία πλήκτρων επιλογής
    Dim strSex As String
    If Me.RadioButtonMale.Checked Then
        strSex = "A"
    ElseIf Me.RadioButtonFemale.Checked Then
        strSex = "\Theta"
    Else
        strSex = ""
    End If
    'Στοιχεία πλαίσιου επιλογής
    Dim strSendMail As String
    If Me.CheckBoxMail.Checked Then
        strSendMail = "NAI"
    Else
        strSendMail = "OXI"
    End If
    'Στοιχεία καταλόγων επιλογής
    Dim strDepartment As String = Me.ListBoxDepartment.SelectedItem
    Dim strSemester As String = Me.ComboBoxSemester.SelectedItem
    'Μορφοποίηση στοιχείων
    Dim strData As String = ""
```
```
strData &= "Όνομα " + strName + vbCrLf
strData &= "Eπώνυμο " + strSurname + vbCrLf
strData &= "Email " + strEmail + vbCrLf
strData &= "Φύλο " + strSex + vbCrLf
strData &= "Ενημέρωση " + strSendMail + vbCrLf
strData &= "Τμήμα " + strDepartment + vbCrLf
strData &= "Εξάμηνο " + strSemester + vbCrLf
'Eμφάνιση στοιχείων
FormShowData.TextBoxStudentData.Text = strData
```

End Sub

Παράδειγμα 6.6

Εμφάνιση πολλαπλών παραθύρων του ιδίου τύπου

Με την προσθήκη ενός νέου τύπου παραθύρου σε μία εφαρμογή δημιουργείται ταυτόχρονα και μία νέα κλάση αντικειμένων με το ίδιο όνομα η οποία χρησιμοποιείται για τον ορισμό των ιδιοτήτων, μεταβλητών, μεθόδων και συμβάντων του παραθύρου. Εάν για παράδειγμα δημιουργηθεί ένα νέο παράθυρο με όνομα FormShowData, τότε δημιουργείται μία νέα κλάση αντικειμένων με το όνομα FormShowData (αρχείο FormShowData.Vb της εικόνας 6.10).

Η εμφάνιση ενός παραθύρου γίνεται με την μέθοδο Show. Στο παράδειγμα 6.6 το άνοιγμα του παραθύρου FormShowData γίνεται με την εντολή:

FormShowData.Show()

Η εντολή αυτή εμφανίζει το παράθυρο το οποίο αντιστοιχεί στην κλάση αντικειμένων FormShowData.

Μία κλάση αντικειμένων μπορεί να έχει αντικείμενα τα οποίαν ανήκουν στην κλάση αυτή. Τα αντικείμενα αυτά έχουν τα ίδια χαρακτηριστικά, ιδιότητες, μεθόδους και συμβάντα με την κλάση αντικειμένων που ανήκουν. Η δημιουργία νέων αντικειμένων μίας κλάσης γίνεται με την εντολή new.

Στο παρακάτω παράδειγμα δημιουργείται ένα νέο αντικείμενο με όνομα newForm το οποίο ανήκει στην κλάση αντικειμένων FormShowData και εμφανίζεται στην οθόνη με την εντολή Show.

```
Dim newForm As FormShowData
newForm = New FormShowData
newForm.Show()
```

Παράδειγμα 6.7

Με την χρήση νέων αντικειμένων τα οποία ανήκουν στην κλάση ενός παραθύρου μπορούμε να εμφανίσουμε πολλαπλά παράθυρα του ιδίου τύπου σε μία εφαρμογή. Τα παράθυρα αυτά είναι πανομοιότυπα αλλά το καθένα έχει τα δικά του δεδομένα και συμπεριφορά.

6.5 Είσοδος δεδομένων με την μέθοδο InputBox

Η μέθοδος InputBox μπορεί να χρησιμοποιηθεί για την είσοδο δεδομένων σε παραθυρικές εφαρμογές. Η μέθοδος αυτή εμφανίζει ένα παράθυρο το οποίο ζητά από τον χρήση να εισάγει δεδομένα. Για παράδειγμα οι εντολές

```
Dim strName As String
strName = InputBox("Δώστε ένα όνομα")
```

εμφανίζουν το παρακάτω παράθυρο το οποίο ζητά από τον χρήση να εισάγει ένα όνομα.

lab_6_ask_1	×
Δώστε ένα όνομα	ОК
	Cancel
<u> </u>	

Εικόνα 6.13 Παράθυρο εισόδου δεδομένων

Η εισαγωγή τελειώνει με το είτε με το πάτημα του πλήκτρου OK. Η μέθοδος InputBox με το πάτημα του πλήκτρου OK επιστρέφει το κείμενο το οποίο πληκτρολόγησε ο χρήστης και το αποθηκεύει στο συγκεκριμένο παράδειγμα στην μεταβλητή strName. Εάν πατηθεί το πλήκτρο Cancel, τότε η εισαγωγή στοιχείων ακυρώνεται και επιστρέφεται κείμενο μηδενικού μήκος (strName = "").

Η πρώτη παράμετρος της μεθόδου InputBox ορίζει το μήνυμα το οποίο εμφανίζεται στον χρήστη κατά την εισαγωγή. Προαιρετικά μπορεί να δοθεί δεύτερη παράμετρος η οποία ορίζει τον τίτλο του παραθύρου και τρίτη παράμετρος η οποία εμφανίζει μία αρχική τιμή στο πλαίσιο κειμένου.

Για παράδειγμα η εντολή

```
strName = InputBox("Δώστε ένα όνομα", "Εισαγωγή δεδομένων", "George")
```

εμφανίζει το παρακάτω παράθυρο

Εισαγωγή δεδομένων	×
Δώστε ένα όνομα	ОК
	Cancel
George	

Εικόνα 6.14 Παράθυρο εισόδου δεδομένων

Η μέθοδος InputBox με την χρήση κατάλληλων συναρτήσεων μετατροπής μπορεί να χρησιμοποιηθεί για την είσοδο αριθμητικών δεδομένων.

Пχ.

```
Dim x As Double
x = CDbl(InputBox("Δώστε πραγματικό αριθμό"))
```

Η συνάρτησης μετατροπής δεν είναι απαραίτητη εάν έχει ορισθεί Option Strict Off

6.6 Εμφάνιση μηνυμάτων στον χρήστη

Η εμφάνιση μηνυμάτων στον χρήστη γίνεται με την χρήση της μεθόδου Show της κλάσης αντικειμένων MessageBox.

Στην απλούστερη μορφή της η μέθοδος Show χρησιμοποιείται με παράμετρο το μήνυμα το οποίο εμφανίζεται στον χρήστη.

Π.χ. η εντολή

MessageBox.Show("Σφάλμα Εφαρμογής")

εμφανίζει το παρακάτω παράθυρο μηνυμάτων στον χρήστη

×
Σφάλμα Εφαρμογής
OK

Εικόνα 6.15 Παράθυρο μηνύματος

Προαιρετικά μπορεί να καθορισθεί ο τίτλος του παραθύρου μηνύματος με την εντολή:

```
MessageBox.Show("Σφάλμα Εφαρμογής", "Λάθος")
```

Μπορεί επίσης να καθορισθεί ο τύπος των πλήκτρων και το εικονίδιο του παραθύρου.

Πχ. Η εντολή

```
MessageBox.Show("Σφάλμα Εφαρμογής", _
"Λάθος", _
MessageBoxButtons.OK, _
MessageBoxIcon.Exclamation)
```

εμφανίζει το παρακάτω παράθυρο:

Λάθος	×
1	Σφάλμα Εφαρμογής
1	ОК

Εικόνα 6.16 Παράθυρο μηνύματος

Το παράθυρο μηνυμάτων στον χρήστη, όταν χρησιμοποιείται με περισσότερα του ενός πλήκτρα (AbortRetryIgnore, OkCancel, RetryCancel, YesNo και YesNoCancel), μπορεί να χρησιμοποιηθεί για την λήψη αποφάσεων αναλόγως της προτίμησης του χρήστη.

Το παρακάτω παράδειγμα ζητά επιβεβαίωση για τον τερματισμό της εφαρμογής. Η μεταβλητή result χρησιμοποιείται για την αποθήκευση της προτίμησης του χρήστη και σε περίπτωση που έχει πατηθεί το πλήκτρο Ok, τότε η εκτέλεση της εφαρμογής σταματά.

Παράδειγμα 6.8

6.7 Παράδειγμα παραθυρικής εφαρμογής

Η παρακάτω εφαρμογή με όνομα Calculator δίνει την δυνατότητα στον χρήστη να κάνει βασικές αριθμητικές πράξεις. Η εφαρμογή αποτελείται από το παράθυρο FormCalculator το οποίο φαίνεται στο σχήμα 6.17.

🔛 Αριθμομηχαν	m _ 🗆 🗙
Αριθμός 1	45,12
Αριθμός 2	23,456
Πράξη	+ · X /
Αποτέλεσμα	68,576
	ζοfiàT

Εικόνα 6.17 Η εφαρμογή της αριθμομηχανής

Το παράθυρο έχει τίτλο «Αριθμομηχανή» και περιέχει τα παρακάτω στοιχείο ελέγχου:

Όνομα στοιχείο ελέγχου	Περιγραφή	Χαρακτηριστικά
LabelNumber1	Ετικέτα πρώτου αριθμού	Text = "Αριθμός 1"
LabelNumber2	Ετικέτα δεύτερου αριθμού	Text = "Αριθμός 2"
LabelOperation	Ετικέττα πράξης	Text = «Πράξη»
LabelResult	Ετικέτα αποτελέσματος	Text = "Αποτέλεσμα"
TextBoxNumber1	Πλαίσιο κειμένου για την	Text = ""
	εισαγωγή του πρώτου	TextAlign =Right
	πραγματικού αριθμού	
TextBoxNumber2	Πλαίσιο κειμένου για την	Text = ""
	εισαγωγή του δεύτερου	TextAlign =Right
	πραγματικού αριθμού	
TextBoxResult	Πλαίσιο κειμένου για την	Text = ""
	εμφάνιση του αποτελέσματος	TextAlign =Right
ButtonAdd Πλήκτρο πρόσθεσης		Event Click
ButtonSubtract	Πλήκτρο αφαίρεσης	Event Click
ButtonMultiply	Πλήκτρο πολλαπλασιασμού	Event Click
ButtonDivide	Πλήκτρο διαίρεσης	Event Click
ButtonOk	Πλήκτρος τέλους εφαρμογής	Event Click

Οι ιδιότητες των στοιχείων ελέγχου φαίνονται στην στήλη του παραπάνω πίνακα με τίτλο «Χαρακτηριστικά». Επιπλέον έχουν καθορισθεί τα συμβάντα Click για όλα τα πλήκτρα της εφαρμογής.

Η μέθοδος η οποία αντιστοιχεί στο πάτημα του πλήκτρου της πρόσθεσης είναι:

```
Private Sub ButtonAdd_Click(ByVal sender As System.Object, _____
ByVal e As System.EventArgs) Handles ButtonAdd.Click
```

```
'Ορισμός πραγματικών μεταβλητών
Dim num1 As Double = 0
Dim num2 As Double = 0
```

'Μετατροπή και αποθήκευση πρώτης τιμής

```
If Me.TextBoxNumber1.Text <> "" Then
    num1 = CDbl(Me.TextBoxNumber1.Text)
End If
'Mετατροπή και αποθήκευση δεύτερης τιμής
If Me.TextBoxNumber1.Text <> "" Then
    num2 = CDbl(Me.TextBoxNumber2.Text)
End If
'Yπολογισμός και εμφάνιση αποτελέσματος
Me.TextBoxResult.Text = num1 + num2
```

End Sub

Παράδειγμα 6.9

Σε περίπτωση που το αποτέλεσμα πρέπει να εμφανισθεί με συγκεκριμένη μορφή (πχ δεκαδικά στοιχεία), μπορεί να χρησιμοποιηθεί η μέθοδος Parse (Κεφάλαιο 2). Π.χ. η εντολή

Me.TextBoxResult.Text = (num1 + num2).ToString("N4")

εκτυπώνει το αποτέλεσμα της πρόσθεσης με ακρίβεια 4 δεκαδικών ψηφίων.

Αντίστοιχες με την μέθοδο της πρόσθεσης είναι και οι μέθοδοι για την υλοποίηση των άλλων αριθμητικών πράξεων.

Για τον τερματισμό της εφαρμογής χρησιμοποιείται η παρακάτω μέθοδος:

Private Sub ButtonOk_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonOk.Click End End Sub

Παράδειγμα 6.10

6.8 Ασκήσεις Κεφαλαίου 6

<u>Άσκηση 6.1</u>

Δημιουργήστε παραθυρική εφαρμογή lab_6_ask_1 η οποία εμφανίζει ένα παράθυρο με τίτλο Άσκηση 6.1. Στο παράθυρο της εφαρμογής εμφανίζονται τρία πλήκτρα με τίτλους (ιδιότητα text) «1», «2» και «3» αντίστοιχα. Με το πάτημα του αντίστοιχου πλήκτρου εμφανίζεται τα μηνύματα «1», «2» και «3» με την χρήση της μεθόδου show της κλάσης αντικειμένων Message Box.

Προσθέστε πλήκτρο για τον τερματισμό της εφαρμογής (παράδειγμα 6.8)

🔜 Άσκηση 6.1 📃 🗖 🗙	
1 2 3	Μήνυμα 🔀
Τερματισμός	1

<u>Άσκηση 6.2</u>

Δημιουργήστε παραθυρική εφαρμογή lab_6_ask_2 η οποία με το πάτημα ενός πλήκτρου με τίτλο «Υπολογισμός εμβαδού κύκλου» ζητά από τον χρήστη να εισάγει την ακτίνα ενός κύκλου (με την μέθοδο InputBox) και εκτυπώνει με την χρήση της μεθόδου show της κλάσης αντικειμένων MessageBox το εμβαδόν του κύκλου.

Χρησιμοποιήστε την επιλογή Option Strict On και τις κατάλληλες συναρτήσεις μετατροπής τύπων μεταβλητών.

<u>Άσκηση 6.3</u>

Δημιουργήστε παραθυρική εφαρμογή lab_6_ask_3 η οποία εμφανίζει το παρακάτω παράθυρο.

🔡 Хокηση 6.3	
Ακτίνα Κύκλου Περίμετρος Κύκλου	
	Υπολογισμός Περιμέτρου

Δώστε το όνομα FormLab6Ask4 στο παράθυρο της εφαρμογής, και εισάγετε τα παρακάτω στοιχεία ελέγχου.

- Ετικέτα με όνομα labelRadius και ιδιότητα Text = «Ακτίνα Κύκλου»
- Ετικέτα με όνομα labelPerimeter και ιδιότητα Text = «Περίμετρος Κύκλου»
- Πλαίσιο κειμένου με όνομα textBoxRadius
- Πλαίσιο κειμένου με όνομα textBoxPerimeter
- Πλήκτρο με όνομα buttonCalculatePerimeter και ιδιότητα Text = «Υπολογισμός Περιμέτρου»

Ο χρήστης εισάγει την ακτίνα του κύκλου στο πλαίσιο κειμένου textBoxRadius και με το πάτημα του πλήκτρου buttonCalculatePerimeter το πρόγραμμα υπολογίζει την περίμετρο του κύκλου και την εμφανίζει στο πλαίσιο κειμένου textBoxPerimeter.

<u>Άσκηση 6.4</u>

Τροποποιήστε την εφαρμογή lab_6_ask_3 προσθέτοντας πλήκτρο με το όνομα ButtonAbout και τίτλο «Σχετικά» το οποίο εμφανίζει ένα νέο παράθυρο όνομα FormAbout και τίτλο «Σχετικά». Το παράθυρο αυτό εμφανίζει ένα πλαίσιο κειμένου με το όνομα της εφαρμογής, την ημερομηνία που δημιουργήθηκε και το όνομα σας.

Πως μπορείτε να εμφανίζεται ταυτόχρονα πολλαπλά παράθυρα FormAbout ;

<u>Άσκηση 6.5</u>

Δημιουργήστε παραθυρική εφαρμογή lab_6_ask_5 η οποία εμφανίζει ένα παράθυρο με τίτλο «Άσκηση 6.5» το οποίο περιέχει ένα αναδιπλούμενο κατάλογο επιλογής (comboBox) με όνομα comboBoxSchool και ένα κατάλογο επιλογής (listBox) με όνομα listBoxDepartment. Ο αναδιπλούμενος κατάλογος επιλογής εμφανίζει τις σχολές «Σ.Τ.ΕΦ.» και «Σ.Ε.Υ.Π». Όταν ο χρήστης επιλέξει κάποια από τις δύο σχολές, τότε εμφανίζονται τα τμήματα της επιλεγμένης σχολής στον κατάλογο επιλογής listBoxDepartment (π.χ. εάν επιλεγεί η σχολή Σ.Τ.Ε.Φ τότε εμφανίζονται τα τμήματα Ηλεκτρονικής, Ηλεκτρολογίας και Πληροφορικής).

<u>Άσκηση 6.6</u>

Δημιουργήστε παραθυρική εφαρμογή lab_6_ask_6 η οποία εμφανίζει ένα παράθυρο με τίτλο «Άσκηση 6.6» το οποίο περιέχει έναν κατάλογο επιλογής (listBox) με όνομα listBoxNames και τρία πλήκτρα με τίτλους «Προσθήκη Ονόματος», «Διαγραφή Ονόματος» και «Πλήθος ονομάτων ». Με το πάτημα του πλήκτρου «Προσθήκη Ονόματος» η εφαρμογή ζητά από τον χρήστη να εισάγει έναν όνομα με την μέθοδο InputBox και τον προσθέτει στον κατάλογο επιλογής. Με το πάτημα του πλήκτρου «Διαγραφή Ονόματος» διαγράφεται το επιλεγμένο όνομα. Με το πάτημα του πλήκτρου «Πλήθος ονομάτων» εμφανίζεται μήνυμα με το πλήθος των στοιχείων του πίνακα επιλογής.

<u>Άσκηση 6.7</u>

Δημιουργήστε εφαρμογή με όνομα lab_6_ask_7 ή οποία απεικονίζει σε παράθυρο με όνομα FormListBox έναν κατάλογο επιλογής με όνομα ListBoxNumbers. Προσθέστε πλήκτρα ButtonAdd, ButtonSum και ButtonOK με τίτλους «Πρόσθεση στοιχείων», «Άθροισμα στοιχείων» και «Τέλος» αντίστοιχα. Το πλήκτρο ButtonAdd ζητά από τον χρήστη να εισάγει διαδοχικά 10 αριθμούς (με την χρήση εντολής επανάληψης και της μεθόδου InputBox) και τους προσθέτει στον κατάλογο επιλογής, το πλήκτρο ButtonSum υπολογίζει το άθροισμα των στοιχείων του καταλόγου και εμφανίζει σχετικό μήνυμα στον χρήστη και το πλήκτρο ButtonOK σταματά την εκτέλεση της εφαρμογής.

<u>Άσκηση 6.8</u>

Δημιουργήστε εφαρμογή με όνομα lab_6_ask_8 ή οποία λειτουργεί ως ηλεκτρονική κλειδαριά. Το παράθυρο της εφαρμογής έχει 10 πλήκτρα με τους αριθμούς από 0 έως 10. Ο χρήστης πρέπει να πληκτρολογήσει τον κατάλληλο αριθμό (πχ 456741) ώστε να εμφανισθεί το μήνυμα ότι δόθηκε ο σωστός αριθμός. Σε κάθε λάθος πάτημα πλήκτρου εμφανίζεται μήνυμα λάθους.

Υπόδειξη: Χρησιμοποιήστε ένα κρυφό πλαίσιο κειμένου (ιδιότητα Visible = False) για την αποθήκευση των αριθμών που εισάγει ο χρήστης. Ελέγξτε σε κάθε πάτημα κάποιου πλήκτρου εάν το κείμενο του πλαισίου κειμένου ταιριάζει με το αριθμό κλειδί '456741'

<u>Άσκηση 6.9</u>

Δημιουργήστε παραθυρική εφαρμογή lab_6_ask_9. Δώστε το όνομα FormStudentData στο αρχικό παράθυρο της εφαρμογής και εισάγετε τα στοιχεία ελέγχου τα οποία φαίνονται στο εικόνα 6.6.

Δώστε τα ονόματα TextBoxName, TextBoxSurname, TextBoxEmail και TextBoxAddress στα πλαίσια κειμένου του παραθύρου που αντιστοιχούν στα στοιχεία του ονόματος, επωνύμου, email και διεύθυνσης του σπουδαστή.

Ορίστε την ιδιότητα Multiline = True για το πλαίσιο κειμένου TextBoxAddress.

Δώστε τα ονόματα RadioButtonMale και RadioButtonFemale στα πλήκτρα επιλογής του φύλου του σπουδαστή και δώστε στην ιδιότητα Text τις τιμές «Α» και «Θ» αντίστοιχα. Δώστε στην ιδιότητα RightToLeft την τιμή Νο και για τα δύο πλήκτρα επιλογής.

Δώστε το όνομα CheckBoxMail και τον τίτλο «Επιθυμεί να ενημερώνεται με email» στο πλαίσιο επιλογής.

Δώστε το όνομα ListBoxDepartment στον κατάλογο επιλογής και εισάγετε στην ιδιότητα Items τις τιμές που φαίνονται στην εικόνα 6.6.

Δώστε το όνομα ComboBoxSemester στον αναδιπλούμενο κατάλογο επιλογής και εισάγετε στην ιδιότητα Items τα εξάμηνα από Α έως Η.

<u>Άσκηση 6.10</u>

Τοποθετήστε στο παράθυρο FormStudentData της εφαρμογής lab_6_ask_9 πλήκτρα με ονόματα ButtonClear, ButtonShow και ButtonOk (εικόνα 6.8) και ορίστε τον τίτλο των πλήκτρων σε «Καθαρισμός», «Εμφάνιση» και «ΟΚ» αντίστοιχα.

Προσθέστε ένα νέο παράθυρο στην εφαρμογή lab_6_ask_9 με όνομα FormDataShow (Εικόνα 6.12). Δώστε το όνομα TextBoxData στο πλαίσιο κειμένου και το όνομα ButtonOK στο πλήκτρο του παραθύρου. Δώστε την ιδιότητα Multiline=True στο πλαίσιο κειμένου TextBoxData και ορίστε τον τίτλο του πλήκτρου σε «OK».

Ορίστε τα παρακάτω συμβάντα και τις μεθόδους των (αντιγράψτε τα παραδείγματα 6.2, 6.3, 6.4, 6.5 και 6.6):

- Load για το παράθυρο FormStudentData
- Click για το πλήκτρο ButtonClear του παράθυρου FormStudentData
- Click για το πλήκτρο ButtonShow του παράθυρου FormStudentData
- Click για το πλήκτρο ButtonOk του παράθυρου FormStudentData
- Click για το πλήκτρο ButtonOk του παράθυρου FormShowData

Εκτελέστε το πρόγραμμα και ελέγξτε την λειτουργικότητα των παραθύρων που δημιουργήσατε.

<u>Άσκηση 6.11</u>

Τροποποιήστε την εφαρμογή lab_6_ask_9:

Εισάγετε τις κατάλληλες εντολές στην μέθοδο FormStudentData_Load ώστε ο καθορισμός των αντικειμένων του κατάλογου επιλογής ListBoxDepartment και του αναδιπλούμενου κατάλογου ComboSemester να γίνεται δυναμικά κατά το άνοιγμα του παραθύρου FormStudentData. (Παράδειγμα 6.1). Ζητήστε επίσης με την χρήση της μεθόδου InputBox ένα όνομα από τον χρήστη και ορίστε το όνομα αυτό ως τον τίτλο του παραθύρου FormStudentData. Ο τίτλος του παραθύρου δεν πρέπει να αλλάζει εάν ακυρωθεί η εισαγωγή του ονόματος.

 Τροποποιήστε την μέθοδο ButtonShow_Click η οποία εμφανίζει τα στοιχεία ενός σπουδαστή ώστε να εμφανίζει με κάθε πάτημα του πλήκτρου ένα νέο παράθυρο με τα στοιχεία του σπουδαστή (Παράδειγμα 6.7)

<u>Άσκηση 6.12</u>

Δημιουργήστε εφαρμογή με όνομα Calculator όπως περιγράφεται στο υποκεφάλαιο 6.7 και ελέγξτε την λειτουργικότητα της.

Στην περίπτωση που η διαίρεση δεν είναι δυνατή (Ο δεύτερος αριθμός είναι μηδενικός) εμφανίστε μήνυμα το οποίο ενημερώνει τον χρήστη ότι η διαίρεση δεν είναι δυνατή.

Προσθέστε πλήκτρο ButtonClear με τίτλο «Καθαρισμός» το οποίο καθαρίζει όλα τα πλαίσια κειμένου του παραθύρου (τους αριθμούς και το αποτέλεσμα της πράξης).

Τροποποιήστε την μέθοδο ButtonOk_Click ώστε να ζητά την επιβεβαίωση από τον χρήστη για το κλείσιμο της εφαρμογής.

Τοποθετήστε ετικέτα με τίτλο «Ακρίβεια Δεκαδικών» και πλαίσιο κειμένου με όνομα TextBoxPrecision και αρχική τιμή «4». Η τιμή του πλαισίου κειμένου καθορίζει τον αριθμό των δεκαδικών ψηφίων του αποτελέσματος.

(Υπόδειξη: Χρησιμοποιήστε για την μορφοποίηση του αποτελέσματος:

Me.TextBoxResult.Text = (num1 + num2).ToString("N"&TextBoxPrecision.Text)

<u>Άσκηση 6.13</u>

Δημιουργήστε εφαρμογή με όνομα lab_6_ask_13 ή οποία εμφανίζει παράθυροο με δύο πλαίσια κειμένου textBoxNumber και textBoxDay. Όταν ο χρήστης εισάγει ένα αριθμό από το 1 έως το 7 στο πλαίσιο κειμένου textBoxNumber, τότε το πρόγραμμα εμφανίζει στο πλαίσιο κειμένου textBoxDay την αντίστοιχη ημέρα της εβδομάδος (1 – Δευτέρα, 2 – Τρίτη, ..., 7 – Κυριακή)

<u>Άσκηση 6.14</u>

Δημιουργήστε εφαρμογή με όνομα lab_6_ask_14 ή οποία εμφανίζει παράθυρο με πλαίσιο κειμένου textBoxPassword και πλήκτρο ButtonOK. Ο χρήστης πρέπει να εισάγει τον κατάλληλο κωδικό (π.χ. 4567) στο πλαίσιο κειμένου και να πατήσει το πλήκτρο ButtonOK για να γίνει ο έλεγχος του κωδικού από πρόγραμμα. Σε περίπτωση λάθους κωδικού εμφανίζεται μήνυμα λάθους στον χρήστη. Σε τρεις διαδοχικές προσπάθειες με λάθος κωδικό το πρόγραμμα τερματίζει την λειτουργία του. (Χρησιμοποιήστε ένα κρυφό πλαίσιο κειμένου με όνομα textBoxCounter για να αποθηκεύετε τον αριθμό των λάθος προσπαθειών)

<u>Άσκηση 6.15</u>

Δημιουργήστε εφαρμογή με όνομα lab_6_ask_15 ή οποία ζητά από τον χρήστη να απαντήσει σε 4 ερωτήσεις πολλαπλής επιλογής και αξιολογεί την απάντηση του. Για την κάθε ερώτηση υπάρχουν 4 δυνατές επιλογές οι οποίες υλοποιούνται με ομαδοποιημένα πλήκτρα επιλογής σε πλαίσιο στοιχείων ελέγχου (RadioButtons τοποθετημένα σε διαφορετικά Panels). Για την κάθε ερώτηση υπάρχει μία μόνο σωστή απάντηση η οποία βαθμολογείται με μία μονάδα. Οι λάθος απαντήσεις δεν βαθμολογούνται.

<u> Άσκηση 6.16</u>

Δημιουργήστε εφαρμογή με όνομα lab_6_ask_16 ή οποία ζητά από τον χρήστη να απαντήσει σε 6 ερωτήσεις NAI/OXI οι οποίες υλοποιούνται με πλαίσια επιλογής και τον αξιολογεί. Η κάθε ερώτηση έχει τρεις καταστάσεις (Nai, Oχι και Αναπάντητη οι οποίες αντιστοιχούν στις καταστάσεις checked, unchecked και indeterminate του πλαισίου επιλογής). Για κάθε σωστή απάντηση δίνεται ένας βαθμός και για κάθε λάθος απάντηση αφαιρείται ένας βαθμός. Οι αναπάντητες ερωτήσεις δεν βαθμολογούνται.

7 Υποπρογράμματα και μέθοδοι

7.1 Υποπρογράμματα σε λειτουργικές μονάδες

Μια εφαρμογή τύπου Console Application αποτελείται τουλάχιστον από μία λειτουργική μονάδα η οποία περιέχει ένα υποπρόγραμμα με την ονομασία main.

Module Module1

```
Sub Main()
Console.WriteLine("Hello")
End Sub
```

```
End Module
```

Παράδειγμα 7.1

Υπάρχουν δύο τύποι υποπρογραμμάτων: οι διαδικασίες και οι συναρτήσεις. Μία λειτουργική μονάδα μπορεί να περιέχει εκτός του υποπρογράμματος Main και άλλες διαδικασίες ή συναρτήσεις.

Διαδικασίες

Οι δηλώσεις των διαδικασιών αρχίζουν με την λέξη Sub (συντομογραφία του Subprocedure) και τελειώνουν με την δήλωση End Sub. Στο παρακάτω παράδειγμα ορίζεται μια διαδικασία με το όνομα PrintLine η οποία εκτυπώνει μια γραμμή κειμένου στην οθόνη τερματικού.

Module Module1

End Module

Παράδειγμα 7.2

Η διαδικασία PrintLine καλείται δύο φορές από την διαδικασία Main. Σε κάθε κλήση της διαδικασίας PrintLine, η εκτέλεση της διαδικασίας Main διακόπτεται προσωρινά μέχρι να ολοκληρωθεί η εκτέλεση της διαδικασίας PrintLine. Το αποτέλεσμα της εκτέλεσης του παραδείγματος 7.2 είναι το παρακάτω:

Ένα υποπρόγραμμα έχει την δυνατότητα να ορίζει τις δικές του τοπικές μεταβλητές όπως φαίνεται στο παράδειγμα 7.3. Οι μεταβλητές αυτές είναι διαθέσιμες μόνο κατά την διάρκεια εκτέλεσης του υποπρογράμματος. Με το τέλος του υποπρογράμματος αποδεσμεύεται η μνήμη του υπολογιστή που χρησιμοποιήθηκε για τις μεταβλητές αυτές. Σε κάθε νέα κλήση του υποπρογράμματος οι μεταβλητές αυτές δημιουργούνται εκ νέου.

```
Sub Printline()
Dim i As Integer
For i = 1 To 18
Console.Write("*")
Next
Console.WriteLine()
End Sub
```

Παράδειγμα 7.3 Η διαδικασία PrintLine με τοπικές μεταβλητές και εντολή επανάληψης

Παράμετροι υποπρογραμμάτων

Η κλήση ενός υποπρογράμματος μπορεί να γίνει και με την χρήση παραμέτρων. Για παράδειγμα μπορεί να καθορισθεί το πλήθος και ο χαρακτήρας τον οποίο τυπώνει η διαδικασία PrintLine (Παράδειγμα 7.4).

```
Module Module1
```

```
Sub Main()
    Printline(18, ".")
    Console.WriteLine("Τμήμα Ηλεκτρονικής")
    Printline(18, ".")
End Sub
Sub Printline(ByVal n As Integer, ByVal c As Char)
    Dim i As Integer
    For i = 1 To n
        Console.Write(c)
    Next
    Console.WriteLine()
End Sub
```

End Module



Οι παράμετροι ενός υποπρογράμματος δηλώνονται εντός παρενθέσεων μετά την δήλωση του ονόματος του υποπρογράμματος. Οι παράμετροι ενός υποπρογράμματος μπορούν να θεωρηθούν ως ειδικού τύπου μεταβλητές στις οποίες δίνεται τιμή κατά την κλήση του υποπρογράμματος.

Στο παραπάνω παράδειγμα κατά την κλήση του υποπρογράμματος με την εντολή PrintLine(18, ".") δίνεται η τιμή 18 στην παράμετρο η και η τιμή "." στην παράμετρο c. Η παράμετρος η καθορίζει το πλήθος των χαρακτήρων που θα εκτυπωθούν και η παράμετρος c τον χαρακτήρα ο οποίος θα εκτυπωθεί.

Οι παράμετροι, όπως και οι τοπικές μεταβλητές ενός υποπρογράμματος είναι προσβάσιμες μόνον από το υποπρόγραμμα και είναι διαθέσιμες μόνο κατή την εκτέλεση του.

Όταν μια παράμετρος χρησιμοποιείται μόνο για την είσοδο δεδομένων σε ένα υποπρόγραμμα (όπως στο παράδειγμα 7.4), τότε δηλώνεται με την λέξη ByVal (ByValue). Στην περίπτωση αυτή, ακόμη κι αν το καλούμενο υποπρόγραμμα τροποποιήσει την τιμή της παραμέτρου, η αλλαγή δεν θα επηρεάσει τιμές των μεταβλητών του υποπρογράμματος που έκανε την κλήση.

Εάν η παράμετρος ενός υποπρογράμματος ορισθεί με την λέξη ByRef τότε η παράμετρος αυτή μπορεί να χρησιμοποιείται τόσο για την είσοδο όσο και για την έξοδο δεδομένων. Στο παράδειγμα 7.5 η διαδικασία IncNumber αυξάνει την τιμή της μεταβλητής number1 η οποία δίνεται ως παράμετρος με την εντολή IncNumber(number1). Κατά την κλήση της διαδικασίας IncNumber, αντί να δοθεί η τιμή της μεταβλητής number1 στην παράμετρο n όπως θα γινόταν με την δήλωση ByVal, η μεταβλητή number1 και η παράμετρος n αποκτούν την ίδια διεύθυνση μνήμης και γίνονται πρακτικά ίδιες μεταβλητές η οποίες αναφέρονται με διαφορετικό όνομα από το υποπρόγραμμα που κάνει την κλήση και το υποπρόγραμμα το οποίο καλείται.

```
Module Module1
```

```
Sub Main()
    Dim number1 As Integer = 0
    IncNumber(number1)
    Console.WriteLine(number1)
End Sub
Sub IncNumber(ByRef n As Integer)
    n = n + 1
End Sub
```

End Module

Παράδειγμα 7.5

Το αποτέλεσμα της εκτέλεσης του παραπάνω προγράμματος είναι ο αριθμός 1

Ένα υποπρόγραμμα μπορεί να έχει προαιρετικές παραμέτρους όπως φαίνεται στο παράδειγμα 7.6. Οι παράμετροι n c δηλώνονται ως προαιρετικές και δίνονται σε αυτές αρχικές τιμές 18 και "." αντίστοιχα.

```
Sub Printline(Optional ByVal n As Integer = 18, _____
Optional ByVal c As Char = "*")
Dim i As Integer
For i = 1 To n
Console.Write(c)
Next
Console.WriteLine()
End Sub
```

Παράδειγμα 7.6

Συναρτήσεις

Οι δηλώσεις των συναρτήσεων αρχίζουν με την λέξη Function και τελειώνουν με την λέξη End Function. Οι συναρτήσεις σε αντίθεση με τις διαδικασίες μετά το τέλος της εκτέλεσης τους επιστρέφουν κάποια τιμή. Ο τύπος δεδομένων ή του αντικειμένου το οποίο επιστρέφει μία συνάρτηση ορίζεται μετά την δήλωση του ονόματος της συνάρτησης. Η τιμή μιας συνάρτησης επιστρέφεται με την εντολή Return.

Στο παράδειγμα 7.7 ορίζεται η συνάρτηση DegreesToRad η οποία μετατρέπει την τιμή μιας γωνίας από μοίρες σε ακτίνια. Η διαδικασία Main, η οποία υπολογίζει το ημίτονου μιας γωνίας η οποία δίνεται από τον χρήστη, καλεί την συνάρτησης DegreesToRad για να μετατρέψει την γωνία σε ακτίνια.

```
Module Module1
```

```
Function DegreesToRad(ByVal angle As Double) As Double
Return angle * Math.PI / 180
End Function
```

```
End Module
```

Παράδειγμα 7.7 Συνάρτηση μετατροπής γωνίας σε ακτίνια

Ένα άλλο παράδειγμα συνάρτησης για τον υπολογισμό του παραγοντικού ενός αριθμού δίνεται από το παρακάτω πρόγραμμα:

```
Module Module1
```

```
End Module
```

Παράδειγμα 7.8 Συνάρτηση υπολογισμού του παραγοντικού ενός αριθμού

Αναδρομικές συναρτήσεις

Στην περίπτωση που κάποια συνάρτηση είναι αναδρομική (όπως το παραγοντικό ενός αριθμού) τότε για την υλοποίηση της μπορεί να χρησιμοποιηθεί αναδρομή, δηλαδή η τιμή της συνάρτησης να υπολογισθεί με την χρήση των τιμών προηγούμενων όρων της συνάρτησης.

Στο παρακάτω παράδειγμα 7.9 χρησιμοποιείται αναδρομή (αντί της εντολής επανάληψης του παραδείγματος 7.8) για να υπολογισθεί το παραγοντικό ενός αριθμού. Το παραγοντικό ενός αριθμού Ν, εάν ο αριθμός είναι μηδέν είναι ο αριθμός ένα, διαφορετικό είναι το γινόμενο του αριθμού με το παραγοντικό του N - 1.

```
Function Paragontiko(ByVal n As Integer)
If n > 1 Then
Return n * Paragontiko(n - 1)
Else
Return 1
End If
End Function
```

Παράδειγμα 7.9 Αναδρομική συνάρτηση

Κατά τον υπολογισμό του παραγοντικού ενός αριθμού, για παράδειγμα του αριθμού 4 (εικόνα 7.1), η συνάρτηση θα καλέσει τον εαυτό της για τον υπολογισμό του παραγοντικού του 3, στην συνέχεια θα κληθεί νέα συνάρτηση για τον του παραγοντικού του αριθμού 2, κ.ο.κ. έως ότου η αναδρομή σταματήσει στον υπολογισμό του παραγοντικού του αριθμού 0. Στην συνέχεια το αποτέλεσμα του

υπολογισμού του παραγοντικού του αριθμού 0 θα χρησιμοποιηθεί για τον υπολογισμό του παραγοντικού του αριθμού 1, κατόπιν για τον υπολογισμό του παραγοντικού του αριθμού 2 κ.ο.κ. μέχρις ότου υπολογισθεί το παραγοντικό του αρχικού αριθμού και τελειώσει η εκτέλεση όλων των συναρτήσεων που χρησιμοποιήθηκαν.



Εικόνα 7.1 Διαδοχικές κλήσεις αναδρομικών συναρτήσεων

Στατικές μεταβλητές υποπρογραμμάτων

Οι μεταβλητές ενός προγράμματος δημιουργούνται κατά την κλήση του υποπρογράμματος και αποδεσμεύονται με το τέλος του υποπρογράμματος (οπότε και δεν διατηρούν τις τιμές τους). Με την χρήση της λέξης Static για την δήλωση μιας μεταβλητής μπορεί να διατηρηθεί η τιμή της μεταβλητής μεταξύ διαδοχικών κλήσεων του υποπρογράμματος.

Στο παράδειγμα 7.7 χρησιμοποιείται η δήλωση μιας στατικής μεταβλητής με το όνομα num η οποία διατηρεί την τιμή της και την αυξάνει κάθε φορά που καλείται η συνάρτηση counter η οποία χρησιμοποιείται ως μετρητής.

```
Module Module1
```

```
Public Sub main()
Console.WriteLine(counter())
Console.WriteLine(counter())
Console.WriteLine(counter())
Console.WriteLine(counter())
Console.WriteLine(counter())
Console.ReadLine()
End Sub
Public Function counter() As Integer
Static num As Integer
num = num + 1
Return num
End Function
```

End Module

Παράδειγμα 7.10 Στατική μεταβλητή

Το αποτέλεσμα της εκτέλεσης του παραπάνω προγράμματος είναι

7.2 Οργάνωση υποπρογραμμάτων και δεδομένων σε λειτουργικές μονάδες

Μια εφαρμογή μπορεί να διαθέτει αρκετές λειτουργικές μονάδες οι οποίες οργανώνουν τα υποπρογράμματα σε λογικές ενότητες.

Η προσθήκη μιας νέας λειτουργικής μονάδας σε μία εφαρμογή γίνεται με την επιλογή «Add New Module» του μενού «Project» και τον ορισμό του ονόματος της λειτουργικής μονάδας στο παράθυρο διαλόγου το οποίο εμφανίζεται στην συνέχεια.

A	dd New Item - lai	b7_ask_1						? X
	Templates:							00 0-0- 0-0- 0-0-
	Visual Studio i	nstalled templa	tes					_
		0.0					Tab	
	Windows Form	Dialog	Explorer Form	MDI Parent Form	About Box	Login Form	Splash Screen	
	V B	1						
	Class	Module	DataSet	SQL Database	User Control	Inherited Form	Inherited User Control	
	Text File							
								_
	A file for storing gr	oups of functions						
ļ	<u>N</u> ame:	Module2.vb						
							Add	Iancel

Εικόνα 7.2 Πρόσθεση λειτουργικής μονάδας

Έστω για παράδειγμα η παρακάτω εφαρμογή η οποία αποτελείται από δύο λειτουργικές μονάδες με ονομασίες Exercises και Trigonemetric Η λειτουργική μονάδα Exercices είναι η αρχική μονάδα εκκίνησης της εφαρμογής και η λειτουργική μονάδα Trigonemetric χρησιμοποιείται ως βοηθητική μονάδα για την ομαδοποίηση τριγωνομετρικών συναρτήσεων και ορισμών.

Η λειτουργική μονάδα Exercises (Παράδειγμα 7.8) αποτελείται από την διαδικασία Main και τις διαδικασίες Askisi1, Askisi2, Askisi3 και Askisi4 οι οποίες αντιστοιχούν στις τέσσερις ασκήσεις ενός εργαστηρίου. Η διαδικασία Main εμφανίζει ένα μενού επιλογών από όπου ο χρήστης μπορεί είτε να επιλέξει την άσκηση την οποία θέλει να εκτελέσει ή να να τελειώσει την εκτέλεση της εφαρμογής.

Η άσκηση 1 ζητά από τον χρήστη να εισάγει μία γωνία σε μοίρες και υπολογίζει το ημίτονο της γωνίας. Για τον υπολογισμό του ημίτονου της γωνίας χρησιμοποιείται η βοηθητική συνάρτηση «Ημίτονο» η οποία ορίζεται στην λειτουργική μονάδα Trigonemetric (Παράδειγμα 7.9).

```
Module exercises
    Sub main()
        Dim selection As String
        Do
            'Εκτύπωση μενού επιλογών χρήστη
            Console.WriteLine()
            Console.WriteLine("Μενού Επιλογών")
            Console.WriteLine()
            Console.WriteLine("1. 'Aoxnon 1")
            Console.WriteLine("2. 'Aoxnon 2")
            Console.WriteLine("3. 'Aoxnon 3")
            Console.WriteLine("4. 'Aoxnon 4")
            Console.WriteLine("5. Τέλος")
            Console.WriteLine()
            Console.Write("Δώστε επιλογή : ")
            selection = Console.ReadLine()
            Console.WriteLine()
            'Εκτέλεση της ανάλογης άσκησης
            Select Case selection
                Case "1" : askisi1()
                Case "2" : askisi2()
                Case "3" : askisi3()
                Case "4" : askisi4()
            End Select
        Loop Until selection = "5"
    End Sub
    Sub askisi1()
        'Άσκηση 1
        Dim x As Double
        Console.Write("Δώστε γωνία σε μοίρες : ")
        x = Console.ReadLine()
        Console.WriteLine("Το ημίτονο της γωνίας είναι {0}", _
                            Trigonometric.Hµ(τονo(x))
    End Sub
    Sub askisi2()
        'Ορισμός υποπρογράμματος άσκησης 2
    End Sub
    Sub askisi3()
        'Ορισμός υποπρογράμματος άσκησης 3
    End Sub
    Sub askisi4()
        Ορισμός υποπρογράμματος άσκησης 4
    End Sub
End Module
```

```
Παράδειγμα 7.11 Η λειτουργική μονάδα Exercises
```

```
Module trigonometric
```

```
'Ορισμός σταθεράς π

Public Const π As Double = Math.PI

Public Function Ημίτονο (ByVal angle As Double) As Double

'Επιστρέφει το ημίτονο γωνίας η οποία δίνεται σε μοίρες

Return Math.Sin(angle * π / 180)

End Function
```

End Module



Το αποτέλεσμα της εκτέλεσης της εφαρμογής εάν ο χρήστης επιλέξει την πρώτη άσκηση και πληκτρολογήσει γωνία 30 μοιρών είναι το παρακάτω:

```
Μενού Επιλογών
1. Άσκηση 1

    Άσκηση 2

3. Άσκηση 3
4. Άσκηση 4
5. Τέλος
Δώστε επιλογή : 1
Δώστε γωνία σε μοίρες : 30
Το ημίτονο της γωνίας είναι 0,5
Μενού Επιλογών
1. Άσκηση 1
2. Άσκηση 2
3. Άσκηση 3
4. Άσκηση 4
5. Τέλος
Δώστε επιλογή :
```

Μετά το τέλος της εκτέλεσης του υποπρογράμματος της άσκησης 1 το μενού επιλογών επανεμφανίζεται ώστε ο χρήστης να επιλέξει κάποια άλλη άσκηση ή τον τερματισμό της εφαρμογής.

Μία λειτουργική μονάδα μπορεί να ορίζει μεταβλητές ή σταθερές η οποίες είναι κοινές για όλα τα υποπρογράμματα της. Οι μεταβλητές αυτές ή οι σταθερές δημιουργούνται και λαμβάνουν τιμές με την εκκίνηση της εφαρμογής και είναι διαθέσιμες σε όλη την διάρκεια εκτέλεσης της εφαρμογής. Για παράδειγμα η λειτουργική μονάδα Trigonemetric ορίζει την σταθερά π.

Οι μεταβλητές, σταθερές και τα υποπρογράμματα μιας λειτουργική μονάδας για να είναι διαθέσιμα στις υπόλοιπες λειτουργικές μονάδες της εφαρμογής θα πρέπει να δηλωθούν ως Public (Δημόσια). Για παράδειγμα η σταθερά π και η συνάρτηση ημίτονο δηλώνονται ως Public. Όταν χρησιμοποιείται η δήλωση Public ή Private για το ορισμό μιας μεταβλητής η λέξη Dim παραλείπεται.

$\Pi_{\cdot}\chi_{\cdot}$ Public myProjectName as String

Δεδομένα και υποπρογράμματα τα οποία δεν χρειάζεται να είναι διαθέσιμα σε άλλες μονάδες ή είναι απαραίτητο να προστατευθούν από την υπόλοιπη εφαρμογή δηλώνονται ως Private (Ιδιωτικά). Εάν κατά τον ορισμό δεδομένων ή υποπρογραμμάτων δεν χρησιμοποιηθεί η δήλωση Public η Private, τότε οι μεν σταθερές και μεταβλητές είναι ιδιωτικές και τα δε υποπρογράμματα δημόσια.

Η αναφορά στα δεδομένα ή σε κάποιο υποπρόγραμμα μιας λειτουργικής μονάδας από κάποια άλλη λειτουργική μονάδα (αν και μπορεί να παραλειφθεί όταν το όνομα της μεταβλητής, σταθεράς ή του υποπρογράμματος είναι μοναδικό) πρέπει να γίνεται με την ταυτόχρονη δήλωση του ονόματος της λειτουργικής μονάδας.

 $\Pi_{.\chi}$. Trigonometric.Hµítovo Trigonometric.π

7.3 Μέθοδοι κλάσεων αντικειμένων

Τα υποπρογράμματα κλάσεων αντικειμένων ονομάζονται μέθοδοι. Για παράδειγμα οι μέθοδοι με τις οποίες διασυνδέονται τα συμβάντα μιας παραθυρικής εφαρμογής είναι υποπρογράμματα. Μία κλάση αντικειμένων μπορεί επίσης να περιλαμβάνει δηλώσεις μεταβλητών και σταθερών.

Στο παρακάτω παράδειγμα δίνεται ένας εναλλακτικός τρόπος υλοποίησης της αριθμομηχανής του υποκεφαλαίου 6.7. Με την χρήση των μεθόδων ReadNumbers και ShowResult και των μεταβλητών num1 και num2 της κλάσης ελαχιστοποιείται και οργανώνεται καλύτερα ο κώδικας της εφαρμογής.

```
Public Class FormCalculator
   Dim num1, num2, result As Double
   Private Sub ReadNumbers()
        'Μετατροπή και αποθήκευση πρώτης τιμής
       If Me.TextBoxNumber1.Text <> "" Then
           num1 = CDbl(Me.TextBoxNumber1.Text)
       Else
            num1 = 0
       End If
        'Μετατροπή και αποθήκευση δεύτερης τιμής
       If Me.TextBoxNumber1.Text <> "" Then
            num2 = CDbl(Me.TextBoxNumber2.Text)
        Else
           num2 = 0
       End If
   End Sub
    Private Sub ShowResult()
       Me.TextBoxResult.Text = result.ToString
   End Sub
    Private Sub ButtonAdd Click (ByVal sender As System.Object,
                         ByVal e As System.EventArgs) Handles ButtonAdd.Click
       ReadNumbers()
        result = num1 + num2
        showResult()
   End Sub
    Private Sub ButtonSubtract_Click (ByVal sender As System.Object,
                        ByVal e As System.EventArgs) Handles ButtonSubtract.Click
       ReadNumbers()
       result = num1 - num2
        ShowResult()
   End Sub
    Private Sub ButtonMultiply_Click(ByVal sender As System.Object,
                        ByVal e As System.EventArgs) Handles ButtonMultiply.Click
       ReadNumbers()
        result = num1 * num2
        ShowResult()
   End Sub
    Private Sub ButtonDivide Click (ByVal sender As System.Object,
                           ByVal e As System.EventArgs) Handles ButtonDivide.Click
```

```
ReadNumbers()
result = num1 * num2
ShowResult()
End Sub
Private Sub ButtonOk_Click(ByVal sender As System.Object, _
ByVal e As System.EventArgs) Handles ButtonOk.Click
End
End Sub
End Class
```

Παράδειγμα 7.13 Εναλλακτική υλοποίηση της αριθμομηχανής

7.4 Ασκήσεις Κεφαλαίου 7

<u>Άσκηση 7.1</u>

Δημιουργήστε εφαρμογή lab_7_ask_1 με τις λειτουργικές μονάδες Exercises και Trigonemetric του υποκεφαλαίου 7.2 (Παραδείγματα 7.11 και 7.12).

<u>Άσκηση 7.2</u>

Τροποποιήστε την εφαρμογή lab_7_ask_1 ώστε με την επιλογή της άσκησης 2 να υπολογίζεται και να εκτυπώνεται η περίμετρος και το εμβαδόν ενός κύκλου. Η ακτίνα του κύκλου δίνεται από τον χρήστη και ο υπολογισμός γίνεται με την χρήση των συναρτήσεων ΕμβαδόνΚύκλου και ΠερίμετροςΚύκλου της οποίες θα ορίσετε στην λειτουργική μονάδα Trigonemetric.

<u>Άσκηση 7.3</u>

Τροποποιήστε την εφαρμογή lab_7_ask_1 ώστε με την επιλογή της άσκησης 3 να υπολογίζονται και να εκτυπώνονται με εντολή επανάληψης οι αντιστοιχίες σε δραχμές των 10, 20, 30, .. 100 Ευρώ. Η μετατροπή γίνεται με χρήση συνάρτησης ΕυρώΣεΔραχμές την οποία θα ορίσετε στην λειτουργική μονάδα Trigonemetric.

<u>Άσκηση 7.4</u>

Τροποποιήστε την εφαρμογή lab_7_ask_1 ώστε με την επιλογή της άσκησης 4 να υπολογίζεται και να εκτυπώνεται η τιμή της συνάρτησης Fibonacci για έναν ακέραιο αριθμό n ο οποίος δίνεται από τον χρήστη. Ο αναδρομικός ορισμός της συνάρτησης είναι:

 $\label{eq:Fibonacci} \begin{array}{ll} Fibonacci(n) = Fibonacci(n-1) + Fibonacci(n-2) & n \geq 2\\ Fibonacci(1) = 1 & n = 1 \ \mbox{η} n = 2 \end{array}$

<u>Άσκηση 7.5</u>

Δημιουργήστε παραθυρική εφαρμογή lab_7_ask_2 με παράθυρο εκκίνησης FormMain όπως φαίνεται στην παρακάτω εικόνα. Δώστε τις ονομασίες ButtonAskisi1, ButtonAskisi2, ButtonAskisi3, ButtonAskisi4 και ButtonOK στα πλήκτρα του παραθύρου και προσθέστε στην εφαρμογή την λειτουργική μονάδα Trigonemetric του υποκεφαλαίου 7.2. Με το πάτημα το πλήκτρου ButtonAskisi1 εμφανίζεται ένα πλαίσιο διαλόγου (InputBox) για την εισαγωγή μιας γωνίας σε μοίρες και εμφανίζεται ένα μήνυμα στον χρήστη με το αποτέλεσμα του υπολογισμού του ημίτονου της γωνίας.

🔛 Εργαστήριο 7	<u>_ </u>
Άσκηση 1	Άσκηση 2
Άσκηση 3	Άσκηση 4
	Τέιλος

<u>Άσκηση 7.6</u>

Τροποποιήστε την παραθυρική εφαρμογή lab_7_ask_2 ώστε με το πάτημα το πλήκτρου ButtonAskisi2 να εμφανίζεται το παρακάτω παράθυρο για τον υπολογισμό της περιμέτρου και του εμβαδού της ακτίνας που εισάγει ο χρήστης σε πλαίσιο κειμένου με όνομα textBoxRadius. Η περίμετρος εμφανίζεται σε πλαίσιο κειμένου με όνομα textBoxPerimeter και το εμβαδόν σε πλαίσιο κειμένου με όνομα textBoxEmbadon. Ο υπολογισμός της περιμέτρου και του εμβαδού γίνεται με το πάτημα του πλήκτρου uε όνομα

🛃 Aகரின 2	
Ακτίνα Περίμετρος Εμβαδό	
	Υπολογισμό

ButtonCalculate. Ο υπολογισμός χρησιμοποιεί τις συναρτήσεις ΕμβαδόνΚύκλου και ΠερίμετροςΚύκλου της οποίες θα ορίσετε στην λειτουργική μονάδα Trigonemetric. Η περίμετρος και το εμβαδόν να εμφανίζονται με ακρίβεια 4 ψηφίων.

<u>Άσκηση 7.7</u>

Τροποποιήστε την παραθυρική εφαρμογή lab 7 ask 2 ώστε με το πάτημα το πλήκτρου ButtonAskisi3 να εμφανίζεται παράθυρο με δύο καταλόγους επιλογής (ListBoxes) με ονόματα ListBoxEuros και ListBoxDrachmas. То περιεχόμενο καταλόγου του επιλογής ListBoxEuros είναι 10, 20, .. 100 (σε Ευρώ) και το επιλογής περιεχόμενο του καταλόγου ListBoxDrachmas είναι οι αντιστοιχίες των Ευρώ σε δραχμές. Τα περιεχόμενα των καταλόγων προστίθενται δυναμικά με εντολή επανάληψης με το πάτημα του πλήκτρου ButtonCalculate. Η μετατροπή γίνεται με χρήση συνάρτησης ΕυρώΣεΔραχμές την οποία θα ορίσετε στην λειτουργική μονάδα Trigonemetric.

🔡 Хокηση 3	_ _
Ευρώ	Δραχμές
10 20 30 40 50 60 70 80 90 100	3407,5 6815 10222,5 13630 17037,5 20445 23852,5 27260 30667,5 34075
	Υπολογισμός

<u>Άσκηση 7.8</u>

Τροποποιήστε την παραθυρική εφαρμογή lab_7_ask_2 ώστε με το πάτημα το πλήκτρου ButtonAskisi4 να εμφανίζεται παράθυρο με κατάλογο επιλογής (ListBox) με όνομα ListBoxProtos. Με το πάτημα του πλήκτρου ButtonCalculate στο περιεχόμενο του καταλόγου επίλογής προστίθονται δυναμικά οι πρώτοι αριθμοί από τον αριθμό 3 έως και τον αριθμό 35. Ο υπολογισμός των πρώτων αριθμών γίνεται με την χρήση εντολής επανάληψης και τη χρήση συνάρτησης isProtos την οποία θα ορίσετε στην λειτουργική μονάδα Trigonemetric. Η συνάρτηση δέχεται ως παράμετρο έναν ακέραιο αριθμό και επιστρέφει την τιμή true εάν ο αριθμός είναι πρώτος και την τιμή false εάν ο αριθμός δεν είναι πρώτος.

🔜 Άσκηση 4	<u> </u>
Πρώτοι αριθμοί	
3 5 7 11 13 17 19 23 29 31	
VotonA	νισμός