



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Ανώτατο Εκπαιδευτικό Ίδρυμα Πειραιά
Τεχνολογικού Τομέα



Εισαγωγή στην Πληροφορική & τον Προγραμματισμό

Ενότητα 5^η: Εισαγωγή στη Γλώσσα Προγραμματισμού

Ι. Ψαρομήλιγκος – Χ. Κυτάγιας
Τμήμα Διοίκησης Επιχειρήσεων



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Ανώτατο Εκπαιδευτικό Ίδρυμα Πειραιά Τεχνολογικού Τομέα**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Σκοποί ενότητας

Στην συγκεκριμένη ενότητα γίνεται μια εισαγωγή στη γλώσσα προγραμματισμού Visual Basic.NET . Πιο συγκεκριμένα παρουσιάζονται τα παρακάτω θέματα: η δήλωση μεταβλητών και σταθερών, οι αριθμητικοί και οι λογικοί τελεστές της γλώσσας, οι αριθμητικές και οι λογικές παραστάσεις, η προτεραιότητα των πράξεων καθώς, η χρήση σχολίων και συναρτήσεις μετατροπής τύπου.

Περιεχόμενα

- Μεταβλητές & Σταθερές
- Αριθμητικοί Τελεστές
- Λογικοί Τελεστές
- Αριθμητικές Πράξεις
- Λογικές Πράξεις
- Συναρτήσεις Μετατροπής Τύπου
- Προγραμματιστικά Σχόλια
- Παραδείγματα

Μεταβλητές

- Για την αποθήκευση και την επεξεργασία των δεδομένων (**data**) που διαχειρίζεται ένα πρόγραμμα χρησιμοποιούμε τις μεταβλητές.
- Οι μεταβλητές αντιστοιχούν σε **θέσεις μνήμης** του Η/Υ και χαρακτηρίζονται από:
 - ένα **όνομα** (name),
 - μια **τιμή** (value) και
 - έχουν κάποιον **τύπο** (type)

Στοιχεία των Μεταβλητών

- Το **όνομα** μιας μεταβλητής μπορεί να δημιουργηθεί από συνδυασμό γραμμάτων (πεζών ή κεφαλαίων), ψηφίων καθώς και το χαρακτήρα υπογράμμισης (underscore `_`). Πιο συγκεκριμένα το όνομα της μεταβλητής πρέπει να αρχίζει με γράμμα, να είναι μοναδικό στην εμβέλειά της δηλαδή δε μπορούμε να έχουμε πανομοιότυπα ονόματα (στην ίδια περιοχή εμβέλειας)
- Η **τιμή** μιας μεταβλητής μπορεί να εκχωρηθεί στην αρχή του προγράμματος (ως σταθερά) ή να προκύψει μέσα από την εκτέλεση των εντολών του
- Ο **τύπος** μιας μεταβλητής προσδιορίζει το **χώρο μνήμης** που θα δεσμεύσει ο υπολογιστής κατά την εκτέλεση του προγράμματος καθώς και την **οργάνωση** με την οποία θα αποθηκευτεί η τιμή αυτή

Δήλωση Μεταβλητών

Για τη δήλωση των μεταβλητών χρησιμοποιούμε την εντολή **Dim**.

Dim όνομα μεταβλητής [**As** όνομα τύπου]

Μπορεί όμως να δηλωθεί και με τα **Private** ή **Public** (γιαυτό θα μιλήσουμε σε λίγο ...)

Παραδείγματα:

Dim a As integer

Dim x As String

Dim sum As integer, choice As boolean, LastName As string

Dim x, y, z, sum As Single

Τύποι Δεδομένων (1)

| Όνομα Τύπου | Μέγεθος | Εύρος τιμών |
|-----------------------|------------------|--|
| Byte | 8-bit (1 byte) | 0 έως 255 (χωρίς πρόσημο) |
| SByte | 8-bit (1 byte) | -128 έως 127 (με πρόσημο) |
| Short (short integer) | 16-bit (2 bytes) | -32,768 έως 32,767 (με πρόσημο) |
| UShort | 16-bit (2 bytes) | 0 έως 65,535 (χωρίς πρόσημο) |
| Integer | 32-bit (4 bytes) | -2,147,483,648 έως 2,147,483,647 (με πρόσημο) |
| UInteger | 32-bit (4 bytes) | 0 έως 4,294,967,295 (χωρίς πρόσημο) |
| Long (long integer) | 64-bit (8 bytes) | -9,223,372,036,854,775,808 έως 9,223,372,036,854,775,807 (9.2...E+18) (με πρόσημο) |
| ULong | 64-bit (8 bytes) | 0 έως 18,446,744,073,709,551,615 (1.8...E+19) (χωρίς πρόσημο) |

Τύποι Δεδομένων (2)

| Όνομα Τύπου | Μέγεθος | Εύρος τιμών |
|---|-----------------------|--|
| Single (<i>single-precision floating-point</i>) | 32-bit (4 bytes) | -3.4028235E+38 έως -1.401298E-45 για αρνητικές τιμές 1.401298E-45 έως 3.4028235E+38 για θετικές τιμές |
| Double (<i>double-precision floating-point</i>) | 64-bit (8 bytes) | -1.79769313486231570E+308 έως -4.94065645841246544E-324 για αρνητικές τιμές 4.94065645841246544E-324 έως 1.79769313486231570E+308 για θετικές τιμές |
| Decimal | 128-bit (16 bytes) | 0 έως +/-79,228,162,514,264,337,593,543,950,335 (+/-7.9...E+28) χωρίς δεκαδικά 0 έως +/-7.9228162514264337593543950335 με 28 θέσεις δεκαδικών Ο μικρότερος μη-αρνητικός αριθμός είναι +/- 0.00000000000000000000000000000001 (+/-1E-28) |

Τύποι Δεδομένων (3)

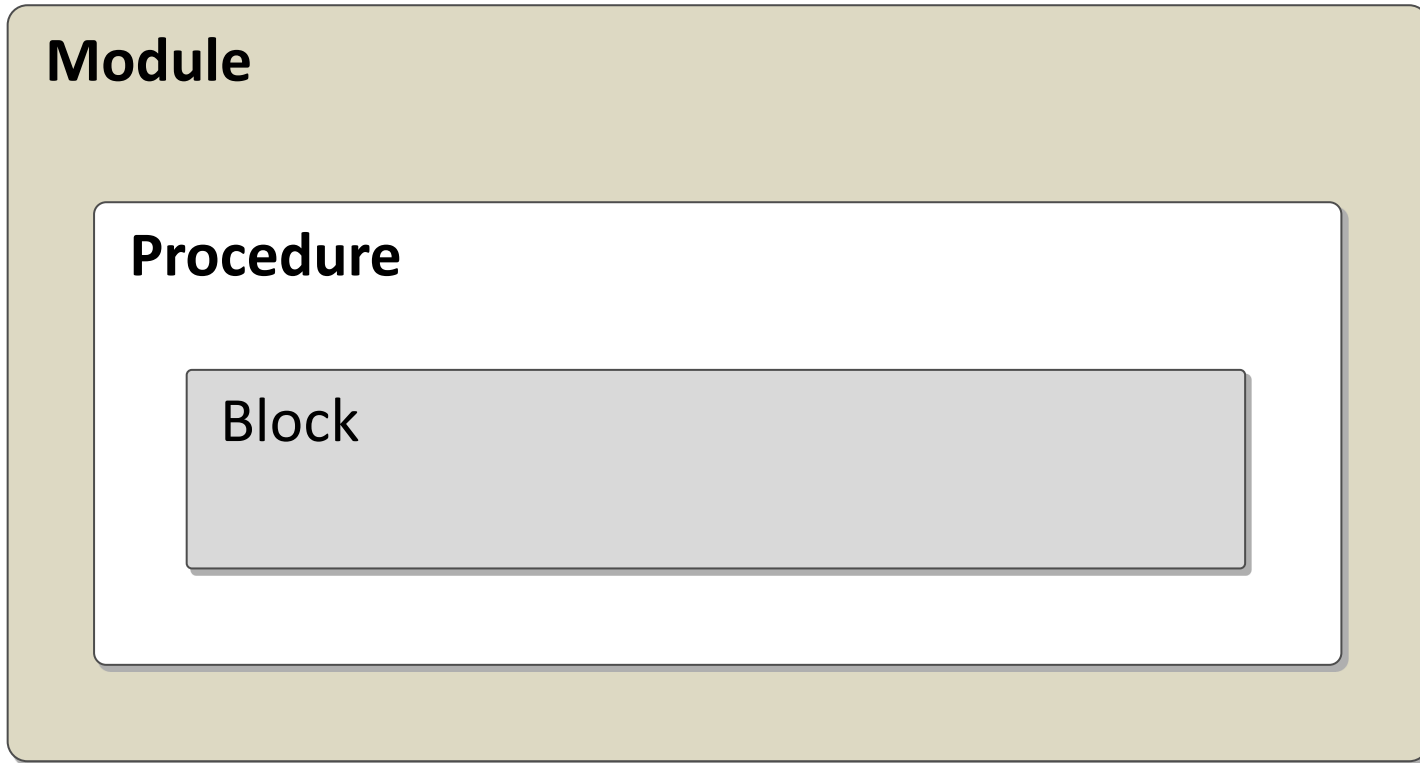
| Όνομα Τύπου | Μέγεθος | Εύρος τιμών |
|-----------------------------|---------------------------------|--|
| Boolean | 8-bit (1 byte) | True ή False |
| Char (single character) | 16-bit (2 bytes) | 0 έως 65535 (χωρίς πρόσημο) |
| String (variable-length) | Συνήθως 16-bit ανά χαρακτήρα | 0 έως περίπου 2 δισεκατομμύρια Unicode χαρακτήρες 16-bit unicode χαρακτήρες |
| Date | 64-bit (8 bytes) | 0:00:00 (μεσάνυχτα) Ιανουάριος 1, 0001 έως 11:59:59 μμ Δεκέμβριος 31, 9999 |
| Object | 32-bit (4 bytes) | Οποιοσδήποτε τύπος μπορεί να αποθηκευτεί σε μια μεταβλητή τύπου Object |

Στην επιστημονική σημειογραφία το "E" συμβολίζει τη δύναμη του 10. Έτσι 3.56E+2 σημαίνει 3.56×10^2 ή αλλιώς 356 και 3.56E-2 σημαίνει $3.56 / 10^2$ ή 0.0356.

Εμβέλεια Μεταβλητών (1)

- Κατά τη δήλωση μεταβλητών μπορούμε να χρησιμοποιήσουμε τις εντολές **Private** και **Public** (πιο αναλυτικά θα το συζητήσουμε σε επόμενη ενότητα):
 - Με τη δήλωση **Public** δηλώνουμε μεταβλητές που θέλουμε να έχουν **καθολική εμβέλεια**, δηλαδή να είναι ορατές από όλα τα σημεία-ενότητες κώδικα του προγράμματός μας
 - Με τη δήλωση **Private** δηλώνουμε μεταβλητές που θέλουμε να έχουν **τοπική εμβέλεια** δηλαδή είτε μέσα σε μια διαδικασία ή το πολύ μέσα σε μια ενότητα κώδικα

Εμβέλεια Μεταβλητών (2)



Σταθερές

- Οι σταθερές (**constants**) αποτελούν επώνυμες τιμές που το πρόγραμμα δε έχει το δικαίωμα να αλλάξει κατά τη διάρκεια της εκτέλεσής του.
- Οι **σταθερές** μπορεί να είναι **οποιοδήποτε τύπου** από αυτούς που ήδη αναφέραμε πχ `integer`, `single`, `char`, `string` κ.λπ.
- Διαχειριζόμαστε μια σταθερά όπως και μια κανονική μεταβλητή με τη μόνη διαφορά ότι η τιμή της δε μπορεί να αλλάξει μετά τη δήλωσή της.

Παραδείγματα Δηλώσεων Σταθερών

Const maxScore As Long = 100000

Const message As String = "TEI PIRAEUS"

Const FPA As Single = 0.23

Const is_a_VIP_Customer As Boolean = false

Sub Main()

 Const PI As Double = 3.14149

 Dim radius, area As Single

 radius = 7

 area = PI * radius * radius

 Console.WriteLine("Area = " & area)

 Console.ReadKey()

End Sub

Πλεονεκτήματα χρήσης σταθερών

- Προσδίδει σαφήνεια στον κώδικα του προγράμματός μας
- Είναι ευκολότερη η συντήρηση του προγράμματός μας σε περίπτωση αλλαγής στο μέλλον της τιμής της σταθεράς
 - για παράδειγμα αν η τιμή του ΦΠΑ αλλάξει στο μέλλον το μόνο που έχουμε να κάνουμε είναι να αλλάξουμε μόνο τη γραμμή δήλωσης της σταθεράς FPA
- Προστατεύει την τιμή της σταθεράς από κατά λάθος αλλαγή της τιμής σε κάποιο άλλο σημείο του κώδικά μας πχ $FPA=0.023$

Αριθμητικοί Τελεστές

| OPERATOR | MEANING | EXAMPLE |
|----------|------------------|------------------------|
| ^ | Power | 2 ^ 5 results in 32 |
| * | Multiplication | 2 * 90 results in 180 |
| / | Real division | 15 / 4 results in 3.75 |
| \ | Integer division | 10 \ 3 results in 3 |
| Mod | Modulus | 25 Mod 8 results in 1 |
| + | Addition | 12 + 6 results in 18 |
| - | Subtraction | 33 - 50 results in -17 |

Αριθμητικές Παραστάσεις

$3 + 4 * 120$

$\text{Data} * 0.06 + 100$

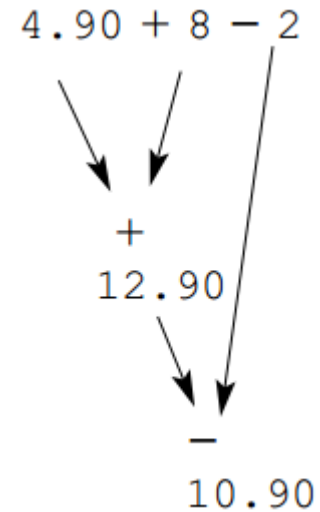
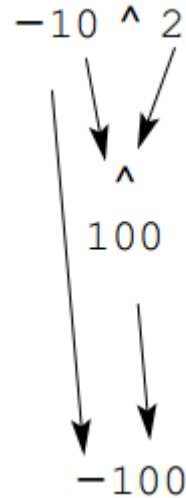
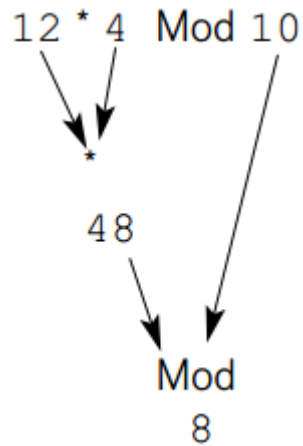
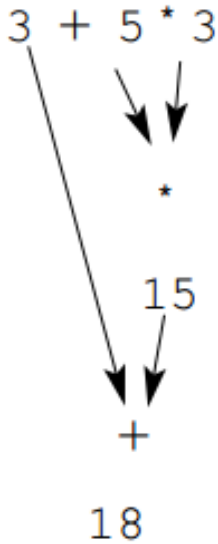
$(12 - \text{Data}) \setminus 8 + 10$

$Z = ((a + b) ^ (-n)) / b$

Προτεραιότητα πράξεων

1. Η παράσταση που βρίσκεται μέσα σε παρένθεση έχει την υψηλότερη προτεραιότητα
2. Ο τελεστής ύψωσης σε δύναμη (Power operator: $^$)
3. Η πράξη που περιέχει μοναδιαίο τελεστή (Unary minus)
4. Πολλαπλασιασμός και διαίρεση ($*$, $/$)
5. Ακεραία διαίρεση (Integer division: \backslash)
6. Υπόλοιπο διαίρεσης (Modulus operator: Mod)
7. Αφαίρεση και Πρόσθεση ($-$, $+$)
8. Ο τελεστής ανάθεσης ($=$) έχει τη χαμηλότερη προτεραιότητα από όλους τους αριθμητικούς τελεστές

Παραδείγματα Προτεραιότητας Πράξεων



Αριθμητικές παραστάσεις με δεδομένα διαφορετικού τύπου

- Το αποτέλεσμα μιας αριθμητικής παράστασης με τελούμενα διαφορετικών τύπων θα είναι του τύπου δεδομένων που είναι περισσότερο ακριβής ή καταλαμβάνει μεγαλύτερο χώρο στη μνήμη
- Για παράδειγμα το άθροισμα ενός Integer με έναν Long θα δώσει αποτέλεσμα τύπου Long, ο πολλαπλασιασμός ενός Single με έναν Integer θα δώσει αποτέλεσμα τύπου Single
- Η VB.NET κάνει σιωπηρή μετατροπή τύπου όταν κάνει ανάθεση τιμής ενός μικρότερου τύπου δεδομένων σε έναν μεγαλύτερο και αντιστρόφως. Παραδείγματα:

Dim **Result** As Double

Result = 500 'Το 500 μετατρέπεται σε Double αποθηκεύεται στη Result

Dim **Count** As Integer

Count = 10 / 3 'Το 3.33333 γίνεται 3 και αποθηκεύεται στην Count

Dim **N** As Integer = 12.78 'Το 12.78 γίνεται 13 & αποθηκεύεται στο N

Τελεστές Συνδυασμένης Ανάθεσης

Μπορούμε να τους χρησιμοποιήσουμε όταν εμφανίζεται η ίδια μεταβλητή και στα 2 μέρη μιας ανάθεσης:

```
Dim Number As Integer
```

```
Number = 5
```

```
Number = Number + 2  => Number += 2
```

```
Number = Number - 6  => Number -= 6
```

```
Number = Number ^ 3  => Number ^= 3
```

```
Number = Number * (12\5) => Number *= 12\5
```

Τελεστές Σύγκρισης

| Τελεστής | Περιγραφή | Παράδειγμα |
|----------|--|-----------------------|
| = | αν οι τιμές των δύο τελούμενων είναι ίσες τότε η συνθήκη είναι αληθής | (A = B) is not true. |
| <> | αν οι τιμές των δύο τελούμενων είναι άνισες τότε η συνθήκη είναι αληθής | (A <> B) is true. |
| > | αν η τιμή του αριστερού τελούμενου είναι μεγαλύτερη από την τιμή του δεξιού τελούμενου τότε η συνθήκη είναι αληθής | (A > B) is not true. |
| < | αν η τιμή του αριστερού τελούμενου είναι μικρότερη από την τιμή του δεξιού τελούμενου τότε η συνθήκη είναι αληθής | (A < B) is true. |
| >= | αν η τιμή του αριστερού τελούμενου είναι μεγαλύτερη ή ίση από την τιμή του δεξιού τελούμενου τότε η συνθήκη είναι αληθής | (A >= B) is not true. |
| <= | αν η τιμή του αριστερού τελούμενου είναι μικρότερη ή ίση από την τιμή του δεξιού τελούμενου τότε η συνθήκη είναι αληθής | (A <= B) is true. |

Όπου A=10 και B=20

Λογικοί Τελεστές

| Τελεστής | Περιγραφή | Παράδειγμα |
|----------|---|-------------------------|
| And | Αν είναι true και τα 2 τελούμενα η συνθήκη είναι true | (A And B) is False. |
| Or | Αν ένα από τα 2 τελούμενα είναι true ή και τα 2 είναι true τότε η συνθήκη είναι true | (A Or B) is True. |
| Not | Αν η συνθήκη είναι true τότε με τον τελεστή NOT θα γίνει false | Not(A And B) is True. |
| Xor | Η συνθήκη είναι true αν μόνο ένα από τα 2 τελούμενα είναι true | A Xor B is True. |
| AndAlso | Αν το 1 ^ο τελούμενο είναι true τότε προχωρά στην αποτίμηση και του 2 ^{ου} τελούμενου και αν είναι επίσης true το αποτέλεσμα είναι true | (A AndAlso B) is False. |
| OrElse | Αν το 1 ^ο τελούμενο είναι true δεν προχωρά στην αποτίμηση και του 2 ^{ου} τελούμενου και το αποτέλεσμα είναι true. Αν το 1 ^ο τελούμενο είναι false τότε προχωρά και στην αποτίμηση του 2 ^{ου} τελούμενου και αν είναι true το αποτέλεσμα είναι true αλλιώς είναι false | (A OrElse B) is True. |
| IsFalse | Προσδιορίζει αν μια παράσταση είναι False. | |
| IsTrue | Προσδιορίζει αν μια παράσταση είναι True. | |

Όπου A=true και B=false

Λογικοί Τελεστές (AND)

| If <i>expression1</i> is | And <i>expression2</i> is | The value of <i>result</i> is |
|--------------------------|---------------------------|-------------------------------|
| True | True | True |
| True | False | False |
| False | True | False |
| False | False | False |

`result = expression1 And expression2`

Λογικοί Τελεστές (OR)

| If <i>expression1</i> is | And <i>expression2</i> is | The value of <i>result</i> is |
|--------------------------|---------------------------|-------------------------------|
| True | True | True |
| True | False | True |
| False | True | True |
| False | False | False |

`result = expression1 Or expression2`

Λογικοί Τελεστές (NOT)

| If <i>expression</i> is | The value of <i>result</i> is |
|-------------------------|-------------------------------|
| True | False |
| False | True |

`result = NOT expression`

Λογικοί Τελεστές (XOR)

| If <i>expression1</i> is | And <i>expression2</i> is | The value of <i>result</i> is |
|--------------------------|---------------------------|-------------------------------|
| True | True | False |
| True | False | True |
| False | True | True |
| False | False | False |

`result = expression1 XOR expression2`

Λογικοί Τελεστές (AndAlso)

| If <i>expression1</i> is | And <i>expression2</i> is | The value of <i>result</i> is |
|--------------------------|---------------------------|-------------------------------|
| True | True | True |
| True | False | False |
| False | (not evaluated) | False |

```
result = expression1 AndAlso expression2
```

Λογικοί Τελεστές (OrElse)

| If <i>expression1</i> is | And <i>expression2</i> is | The value of <i>result</i> is |
|--------------------------|---------------------------|-------------------------------|
| True | (not evaluated) | True |
| False | True | True |
| False | False | False |

```
result = expression1 OrElse expression2
```

Συναρτήσεις μετατροπής τύπου

| Όνομα | Τύπος | Εύρος <i>expression</i> argument |
|----------------|--|--|
| CBool | Boolean Data Type (Visual Basic) | Any valid Char or String or numeric expression. |
| CByte | Byte Data Type (Visual Basic) | 0 through 255 (unsigned); fractional parts are rounded. ¹ |
| CChar | Char Data Type (Visual Basic) | Any valid Char or String expression; only first character of a String is converted; value can be 0 through 65535 (unsigned). |
| CDate | Date Data Type (Visual Basic) | Any valid representation of a date and time. |
| CDbl | Double Data Type (Visual Basic) | -1.79769313486231570E+308 through -4.94065645841246544E-324 for negative values; 4.94065645841246544E-324 through 1.79769313486231570E+308 for positive values. |
| CDec | Decimal Data Type (Visual Basic) | +/-79,228,162,514,264,337,593,543,950,335 for zero-scaled numbers, that is, numbers with no decimal places. For numbers with 28 decimal places, the range is +/- 7.9228162514264337593543950335. The smallest possible non-zero number is 0.00000000000000000000000001 (+/-1E-28). |
| CInt | Integer Data Type (Visual Basic) | -2,147,483,648 through 2,147,483,647; fractional parts are rounded. ¹ |
| CLng | Long Data Type (Visual Basic) | -9,223,372,036,854,775,808 through 9,223,372,036,854,775,807; fractional parts are rounded. ¹ |
| CObj | Object Data Type | Any valid expression. |
| CSByte | SByte Data Type (Visual Basic) | -128 through 127; fractional parts are rounded. ¹ |
| CShort | Short Data Type (Visual Basic) | -32,768 through 32,767; fractional parts are rounded. ¹ |
| CSng | Single Data Type (Visual Basic) | -3.402823E+38 through -1.401298E-45 for negative values; 1.401298E-45 through 3.402823E+38 for positive values. |
| CStr | String Data Type (Visual Basic) | Returns for CStr depend on the <i>expression</i> argument. See Return Values for the CStr Function (Visual Basic) . |
| CUInt | UInteger Data Type | 0 through 4,294,967,295 (unsigned); fractional parts are rounded. ¹ |
| CULng | ULong Data Type (Visual Basic) | 0 through 18,446,744,073,709,551,615 (unsigned); fractional parts are rounded. ¹ |
| CUShort | UShort Data Type (Visual Basic) | 0 through 65,535 (unsigned); fractional parts are rounded. ¹ |

Τέλος Ενότητας



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

