



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Ανώτατο Εκπαιδευτικό Ίδρυμα Πειραιά
Τεχνολογικού Τομέα



Εισαγωγή στην Πληροφορική & τον Προγραμματισμό

Ενότητα 10^η: Εισαγωγή στον Αντικειμενοστραφή
Προγραμματισμό

Ι. Ψαρομήλιγκος – Χ. Κυτάγιας
Τμήμα Διοίκησης Επιχειρήσεων



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



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 στον αντικειμενοστραφή προγραμματισμό. Παρουσιάζονται βασικές έννοιες, όπως: η έννοια του αντικειμένου (object), ιδιότητες (properties) και μέθοδοι (methods) αντικειμένου, μηνύματα (messages), η έννοια της κλάσης (class), η έννοια της αφάιρεσης (abstraction), γενίκευση-εξειδίκευση (generalization-specialization), κληρονομικότητα (inheritance).

Περιεχόμενα

- Η έννοια του αντικειμένου (Object)
- Ιδιότητες (Properties) και μέθοδοι (Methods) αντικειμένου
- Επικοινωνία αντικειμένων (μηνύματα - messages)
- Η έννοια της κλάσης (Class)
- Η έννοια της αφάιρεσης (Abstraction)
- Γενίκευση (Generalization), Εξειδίκευση (Specialization) & Κληρονομικότητα (Inheritance)
- Συμβάντα και Χειριστές Συμβάντων
- Επαναχρησιμοποίηση αντικειμένων

Η έννοια του αντικειμένου

- Ως αντικείμενα μπορούμε να θεωρήσουμε σχεδόν τα πάντα, δηλαδή, οτιδήποτε υπάρχει:
 - στον πραγματικό κόσμο π.χ. Μαθητής, Ραδιόφωνο, Laptop, Αυτοκίνητο, κ.λπ.
 - στον κόσμο της νόησης και αφορά τον τρόπο που αντιλαμβανόμαστε τον πραγματικό κόσμο π.χ. η θεωρία της σχετικότητας
- Αυτό σημαίνει ότι τα αντικείμενα μπορεί να μην υπάρχουν
- ... και να είναι παράγωγα της αντίληψής μας για τη συμπεριφορά και τη δομή των αντικειμένων του πραγματικού κόσμου



Προγραμματιστικά Αντικείμενα

The diagram illustrates various programming objects in three different contexts:

- GUI Application (Ορθογώνια Τρίγωνα):**
 - Αντικείμενα Forms:** The entire application window.
 - Αντικείμενο Label:** The text "Αποτελέσματα" (Results).
 - Αντικείμενα Text Box:** Input fields for "a=3", "b=4", and "c=5".
 - Αντικείμενο List Box:** A list box containing "a=3 b=4 c=5" and "a=6 b=8 c=10".
 - Other Elements:** Input fields for "P= 1" and "Q= 10", and buttons for "ΒΡΕΣ" (Find) and "ΚΑΘΑΡΙΣΕ" (Clear).
- Calculator:**
 - Αντικείμενα Buttons:** Individual buttons like "AC", "CE", "%", "\", "7", "8", "9", "/", "4", "5", "6", "x", "1", "2", "3", "-", "0", ".", "=", and "+".
- Console Window (C:\WINDOWS\system32\cmd.exe):**
 - Αντικείμενο Console:** The text "Hello world!!!" and "Press any key to continue . . .".

Ιδιότητες αντικειμένου

- Μια ιδιότητα (property) εκφράζει ένα **χαρακτηριστικό** του αντικειμένου που μας **ενδιαφέρει**. Για παράδειγμα, αν το αντικείμενο παριστάνει ένα μαθητή μπορεί να έχει τις ιδιότητες: όνομα μαθητή, φύλο (αγόρι ή κορίτσι), ίδρυμα, κ.λπ.
- Το **είδος** της εφαρμογής που αναπτύσσουμε καθορίζει τα αντικείμενα και τα αντίστοιχα χαρακτηριστικά που ενδιαφέρουν



Ιδιότητες προγραμματιστικού αντικειμένου

```
Module Module1
```

```
Sub Main()
```

```
    Console.
```

```
End Sub
```

```
End Module
```

Title
TreatControlCAsInput
WindowHeight
WindowLeft
WindowTop
WindowWidth
Write
WriteLine

Common

All

Ιδιότητες
αντικειμένου
Console

Μέθοδοι αντικειμένου

- Μια μέθοδος (method) εκφράζει μια **λειτουργία** που μπορεί να φέρει σε πέρας ένα αντικείμενο. Γιαυτό το λόγο λέμε ότι οι μέθοδοι καθορίζουν τη **συμπεριφορά** του αντικειμένου.
- Για παράδειγμα, σε ένα αντικείμενο «μαθητής» μπορούμε να έχουμε μεθόδους όπως: εγγραφή σε εξάμηνο, δήλωση μαθήματος, ανάθεση πρακτικής άσκησης, κ.λπ.



Μέθοδοι προγραμματιστικού αντικειμένου

αντικειμένου

```
Module Module1
    Sub Main()
        Console.
    End Sub
End Module
```

Μέθοδοι αντικειμένου Console

- Title
- TreatControlCAsInput
- WindowHeight
- WindowLeft
- WindowTop
- WindowWidth
- Write
- WriteLine

Common All

Μέθοδοι & Μηνύματα (Methods & Messages)

- Μηνύματα (**Messages**)

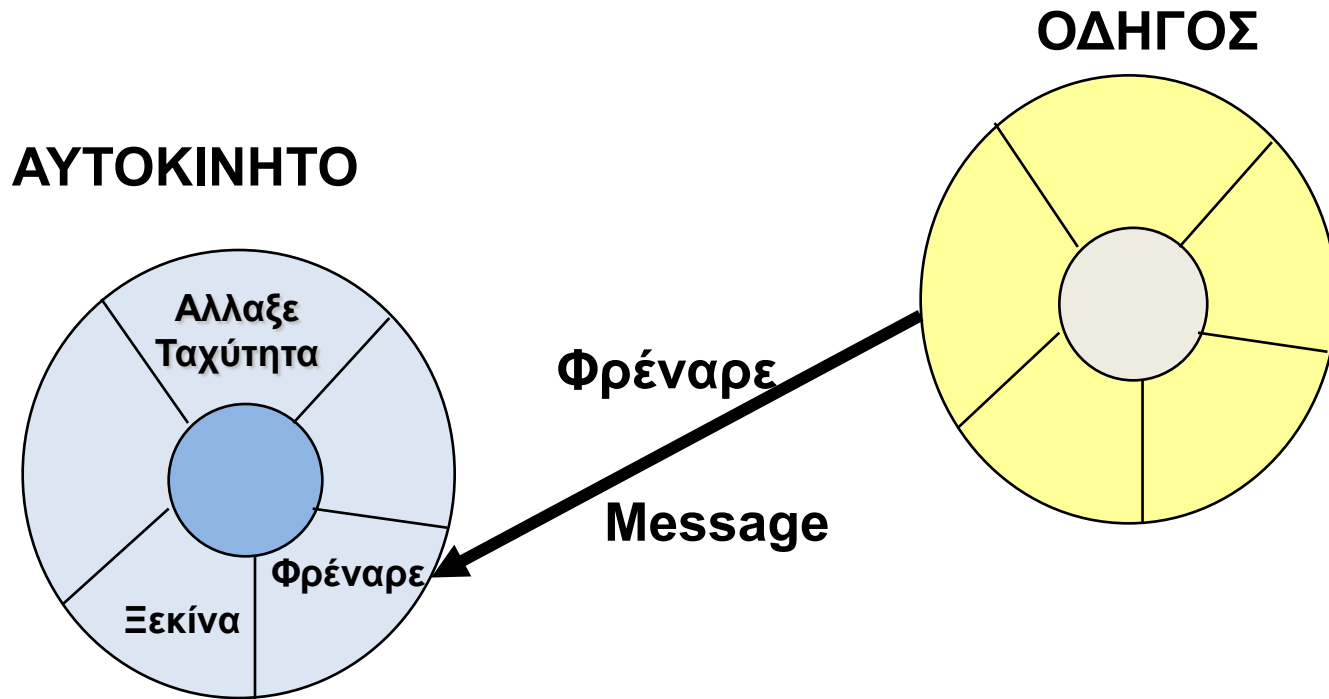
- Τα αντικείμενα επικοινωνούν και αλληλεπιδρούν μεταξύ τους αποστέλλοντας **μηνύματα**.
- Επιπλέον πληροφορία που απαιτείται δίνεται ως **παράμετροι (parameters)** των μηνυμάτων.

`Console.WriteLine("Hello World!!")`

- Πλεονεκτήματα

- Όλη η αλληλεπίδραση υποστηρίζεται με μεταβίβαση μηνυμάτων.
- Τα αντικείμενα δε χρειάζεται να βρίσκονται στην ίδια **διεργασία (process)**, ούτε καν στον ίδιο υπολογιστή.

Επικοινωνία Αντικειμένων

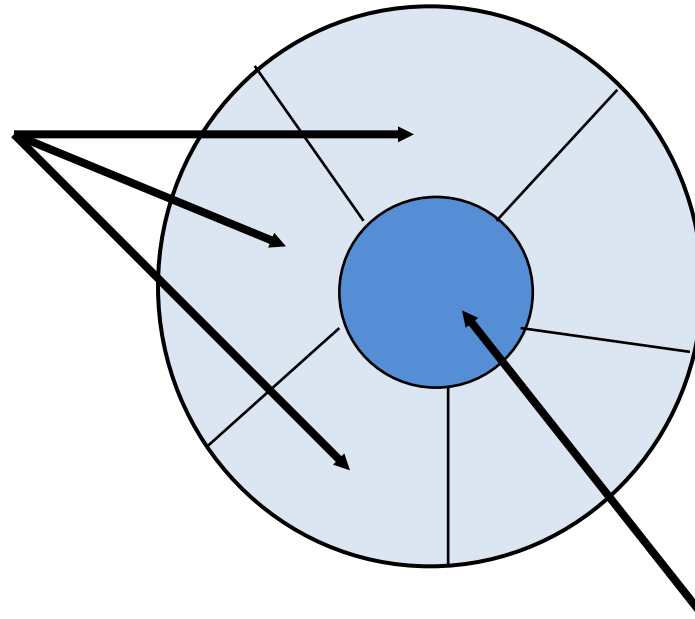


Κελυφοποίηση (Encapsulation) – Απόκρυψη Πληροφορίας (Information Hiding)

- η ελεγχόμενη ορατότητα των χαρακτηριστικών και των μεθόδων ενός αντικειμένου από άλλα αντικείμενα
- η απόκρυψη των λεπτομερειών υλοποίησης ενός αντικειμένου από το περιβάλλον του

Δομή Αντικειμένου

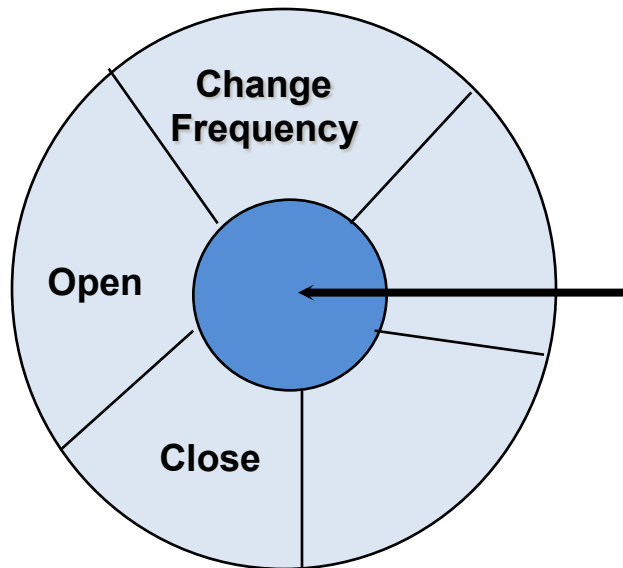
Δημόσιο Μέρος
Public Part



Ιδιωτικό Μέρος - Private Part
(Implementation Details)

Παράδειγμα

Object: ΡΑΔΙΟΦΩΝΟ



*Λεπτομέρειες που αφορούν
την υλοποίηση
Implementation Details*

Διεπαφή (interface) Αντικειμένου (1)

- Οι μέθοδοι και οι ιδιότητες στις οποίες μπορεί να επιδράσει ο χρήστης ενός αντικειμένου χαρακτηρίζονται σαν **δημόσιες (public)** και οι απροσπέλαστες χαρακτηρίζονται σαν **ιδιωτικές (private)**.
- Οι μέθοδοι και οι ιδιότητες που είναι **public** αποτελούν αυτό που λέμε **διεπαφή (interface)** του αντικειμένου με τον έξω κόσμο.

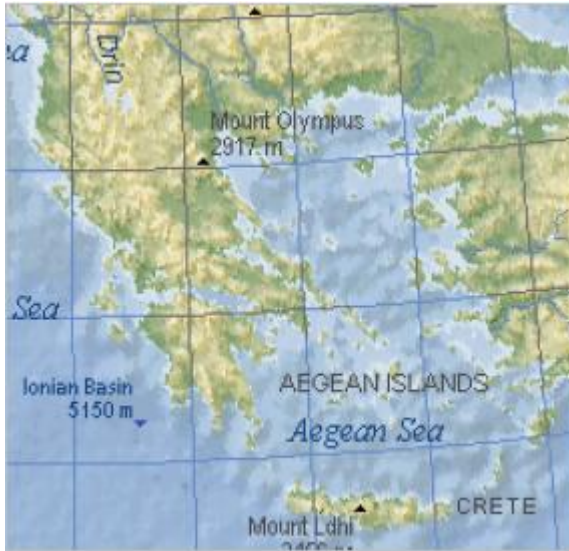
Διεπαφή (interface) Αντικειμένου (2)

- Ο «έξω κόσμος» ενός αντικειμένου επιτρέπεται να προσπελάσει μόνο τις ιδιότητες και μεθόδους που έχουν τον προσδιορισμό **public**.
- Οι μέθοδοι και οι ιδιότητες που είναι **private** δεν είναι προσπελάσιμες στον έξω κόσμο του αντικειμένου.
- Ο «έξω κόσμος» ενός αντικειμένου μπορεί να είναι ο κατασκευαστής μιας εφαρμογής που χρησιμοποιεί το αντικείμενο ή μια άλλη προγραμματιστική οντότητα π.χ. μια υπορουτίνα

Η έννοια της ενθυλάκωσης (Encapsulation)

- Το αντικείμενο συμπεριφέρεται προς τους έξω σαν να έχει μια κάψα (θύλακα) όπου κρύβει και προστατεύει τα μυστικά του. Η συμπεριφορά αυτή των αντικειμένων ονομάζεται **ενθυλάκωση (Encapsulation)** και αποτελεί ένα χαρακτηριστικό γνώρισμα του αντικειμενοστρεφούς προγραμματισμού.
- Ο χαρακτηρισμός των μεθόδων και των ιδιοτήτων σαν public ή private γίνεται από τον προγραμματιστή που δημιουργεί το αντικείμενο.

Η έννοια της αφαίρεσης (abstraction) (1)



- Από το σύνολο των λεπτομερειών μιας οντότητας ή ενός γεγονότος επιλέγουμε να επικεντρώσουμε την προσοχή μας μόνο σε κάποιες «αφαιρώντας» τις υπόλοιπες

Η έννοια της αφαίρεσης (abstraction) (2)

- Γενικά στην καθημερινή μας ζωή χρησιμοποιούμε τα αντικείμενα ενδιαφερόμενοι μόνο για ορισμένες ιδιότητές τους. Λέμε ότι κάνουμε μια **λογική αφαίρεση (abstraction)** των χαρακτηριστικών και λειτουργιών του όταν απασχολούμαστε με ένα αντικείμενο.
- κατά συνέπεια και στον προγραμματισμό τα αντικείμενα που χρησιμοποιούμε θα έχουν χαρακτηριστικά και λειτουργίες που προέρχονται από **λογική αφαίρεση (abstraction)**
 - γίνεται στη φάση της ανάλυσης απαιτήσεων

Κατάσταση & Συμπεριφορά Αντικειμένου

- Ένα αντικείμενο λέμε ότι έχει:
 - **Κατάσταση (Status)**
 - περιγράφει όλες τις στατικές του ιδιότητες, π.χ. Χρώμα=«Μπλε», Υψος=10cm, κ.λπ.
 - **Συμπεριφορά (Behavior)**
 - είναι ο τρόπος με τον οποίο αυτό ανταποκρίνεται σε κλήσεις από το περιβάλλον του – ένα σύνολο από λειτουργίες (μέθοδοι)
 - **Ταυτότητα (Identity)**
 - είναι η μοναδική διάκριση του αντικειμένου από τα ομοειδή του

Η έννοια της Κλάσης (1)

Η κλάση αποτελεί το **πρότυπο** (template) – **μήτρα** για τη δημιουργία αντικειμένων.

Είναι όπως ένα **εργοστάσιο** που παράγει ένα συγκεκριμένο είδος (τύπο) προϊόντων.

Ένα αντικείμενο αποτελεί **στιγμιότυπο** μιας κλάσης.

Η κλάση εκφράζει μια ομάδα αντικειμένων με κοινές ιδιότητες (**χαρακτηριστικά**) και μεθόδους (**συμπεριφορά**)



Η έννοια της Κλάσης (2)

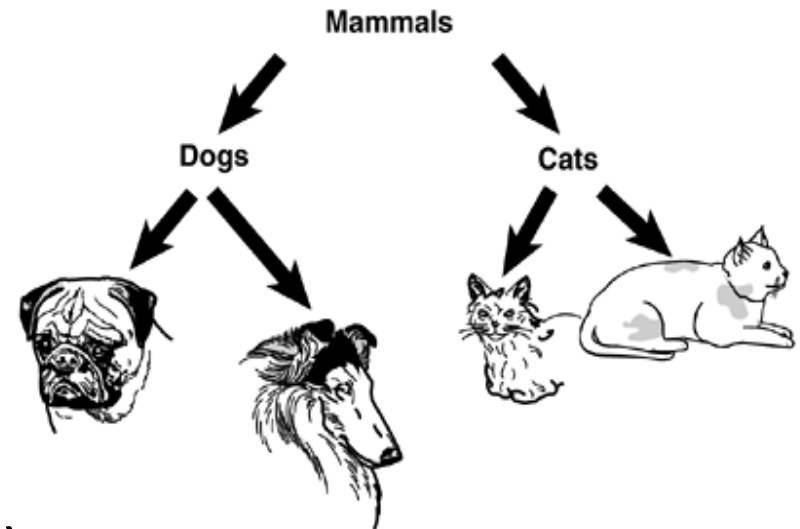
- Έτσι, από μια συγκεκριμένη κλάση ο κατασκευαστής δημιουργεί πανομοιότυπα αντικείμενα (π.χ. αυτοκίνητα, ραδιόφωνα, κλπ).
- **Η κλάση δεν είναι αντικείμενο** αλλά οι προδιαγραφές αντικειμένου.
- **Ποιος είναι ο δημιουργός ενός αντικειμένου (πχ ενός ραδιοφώνου); Αυτός που το σχεδίασε ή αυτός που το κατασκεύασε;**
- Ευτυχώς στον αντικειμενοστραφή προγραμματισμό ο κατασκευαστής δεν είναι πρόσωπο αλλά κώδικας και επομένως δεν χρειάζεται αναγνώριση του έργου του!

Η έννοια της Κλάσης (3)

- Στον αντικειμενοστραφή προγραμματισμό για να πάρουμε ένα νέο αντικείμενο δε δημιουργούμε το αντικείμενο αυτό καθαυτό, αλλά προδιαγράφουμε σε ειδική ενότητα κώδικα μια **κλάση** ...
- ... στη συνέχεια ο προγραμματιστής μιας εφαρμογής που χρειάζεται αντικείμενα από την κλάση δημιουργεί **στιγμιότυπα της κλάσης** (δηλαδή αντικείμενα) με τους τρόπους που του παρέχει το προγραμματιστικό περιβάλλον

Γενίκευση (Generalization) – Εξειδίκευση (Specialization) – Κληρονομικότητα (Inheritance)

- Generalization: έμφαση στα κοινά σημεία (κοινά χαρακτηριστικά και λειτουργίες)
- Specialization: έμφαση στις διαφορές (επιπλέον χαρακτηριστικά ή λειτουργίες)



Γενίκευση (Generalization) – Εξειδίκευση (Specialization) – Κληρονομικότητα (Inheritance)

- Inheritance : τα κοινά σημεία κληρονομούνται



Γενίκευση (Generalization) – Εξειδίκευση (Specialization) – Κληρονομικότητα (Inheritance)

- **Στον αντικειμενοστραφή προγραμματισμό, ο προγραμματιστής δημιουργός μιας νέας κλάσης μπορεί να στηριχθεί σε κλάση που ήδη υπάρχει και να κτίσει νέα κλάση η οποία κληρονομεί τις ιδιότητες και μεθόδους της παλαιάς.**
- Κάθε αντικείμενο που θα δημιουργηθεί από τη νέα κλάση θα κληρονομήσει όλες τις ιδιότητες και τις μεθόδους της κλάσης αυτής καθώς και **όλες τις ιδιότητες και τις μεθόδους των κλάσεων που κληρονόμησε**

Επαναχρησιμοποίηση αντικειμένων (1)

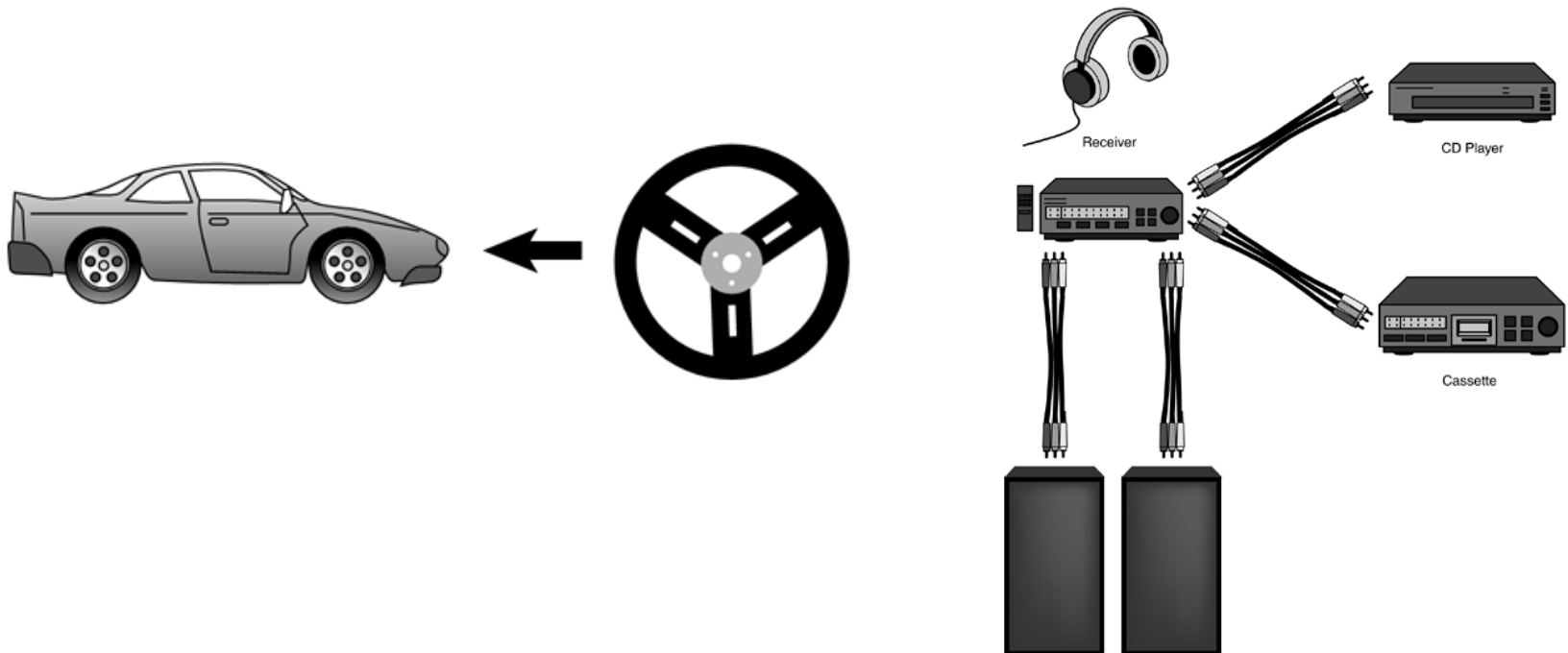
- Σε προγραμματιστικά περιβάλλοντα τα αντικείμενα παράγονται από κλάση. Δημιουργώντας μια εφαρμογή δεν θα έχουμε διαθέσιμα αντικείμενα, αλλά κλάση (ή κλάσεις) από την οποία (-ες) παράγουμε οσαδήποτε επιθυμητά αντικείμενα για την εφαρμογή μας. Η κλάση προδιαγράφει τις ιδιότητες, μεθόδους και συμβάντα των αντικειμένων που θα παραχθούν και οι προδιαγραφές τοποθετούνται σε αρχεία. Τα αρχεία που περιέχουν κλάση (ή κλάσεις) μπορούμε να τα χρησιμοποιήσουμε και σε άλλες εφαρμογές στις οποίες φυσικά θα δημιουργήσουμε νέα αντικείμενα της κλάσης (ή των κλάσεων).
- Η διαδικασία αυτή αναφέρεται με τον όρο "**Επαναχρησιμοποίηση (reusing) αντικειμένων**".

Επαναχρησιμοποίηση αντικειμένων (2)

- Ο όρος επαναχρησιμοποίηση αντικειμένων περιλαμβάνει και την περίπτωση δημιουργίας νέας κλάσης με **συνιστώσες αντικείμενα άλλων κλάσεων**.
- Έχοντας διαθέσιμη μια κλάση (οπότε μπορούμε να παράγουμε αντικείμενα της) μπορούμε να δημιουργήσουμε μια νέα κλάση στην οποία να περιλάβουμε σαν **ιδιότητα** ένα (ή περισσότερα) αντικείμενα της παλαιάς κλάσης

Επαναχρησιμοποίηση αντικειμένων (3)

- Ένα αντικείμενο μπορεί να αποτελείται από άλλα αντικείμενα
- ... τα αντικείμενα στην πραγματικότητα “κτίζονται” από άλλα αντικείμενα



Συμβάντα (Events) ₁

Το αντικείμενο μπορεί να αναγνωρίσει (ή να επηρεασθεί από) κάποιες **δράσεις** (actions) που προκαλούνται από το χρήστη, από άλλους εξωτερικούς παράγοντες ή και από το ίδιο το αντικείμενο.

Η αναγνώριση των δράσεων αυτών μεταφράζεται σαν δυνατότητα δημιουργίας συμβάντων (events) από το αντικείμενο.

Ας δούμε τη συσχέτιση μερικών δράσεων με συμβάντα.



<http://csis.pace.edu/~bergin/Java/eventfundamentals.html>

Συμβάντα (Events) ₂

- Δράση από το χρήστη πχ κάνει κλικ με το ποντίκι στο Button μιας φόρμας καταχώρησης στοιχείων
- Δράση από εξωτερικό παράγοντα πχ ενεργοποίηση μιας εφαρμογής ύστερα από συμβάν που προκλήθηκε από συσκευή συναγερμού
- Δράση από το ίδιο το αντικείμενο πχ σε ένα παιχνίδι ο χρήστης πετυχαίνει high score οπότε ενεργοποιείται η διαδικασία καταχώρησης των στοιχείων του



<http://csis.pace.edu/~bergin/Java/eventfundamentals.html>

Το αντικείμενο στη Visual Basic.NET

- Είναι μια προγραμματιστική οντότητα που έχει δημιουργηθεί με κάποιο τύπο ("καλούπι") που ονομάζουμε **κλάση (class)**. Κάθε αντικείμενο είναι η υλοποίηση των προδιαγεγραμμένων σε μια κλάση. Λέμε ότι το αντικείμενο είναι μια "**περίπτωση**" (**instance**) της κλάσης από την οποία προκύπτει. Από μια κλάση μπορεί να δημιουργηθούν οσαδήποτε αντικείμενα (instances) επιθυμούμε

Το αντικείμενο στη Visual Basic.NET (μεταβλητές με αναφορά)

- Κάθε αντικείμενο χαρακτηρίζεται από ένα όνομα. Το όνομα του αντικειμένου είναι μια μεταβλητή που εκφράζει **αναφορά (reference)** προς το αντικείμενο, δηλαδή μια διεύθυνση μνήμης, όπου φορτώνεται το αντικείμενο. Η Visual Basic προσπελαύνει το αντικείμενο με το όνομά του.
- Μπορούμε να δημιουργούμε και άλλες αναφορές προς το αντικείμενο (με άλλα λόγια να του προσφέρουμε επιπλέον ονόματα).

Αντικείμενα με γραφικά

- Μπορεί να παρουσιάζεται στο χρήστη της εφαρμογής με κάποιο παράθυρο των windows στην οθόνη. Αντικείμενα που παρουσιάζονται με ορατό μέρος στον χρήστη της εφαρμογής και με τα οποία ο προγραμματιστής μπορεί να προσφέρει δεδομένα στην εφαρμογή ή να βλέπει διάφορες τιμές και γενικά να ελέγχει τον τρόπο λειτουργίας της εφαρμογής ονομάζονται **χειριστήρια (controls)**.

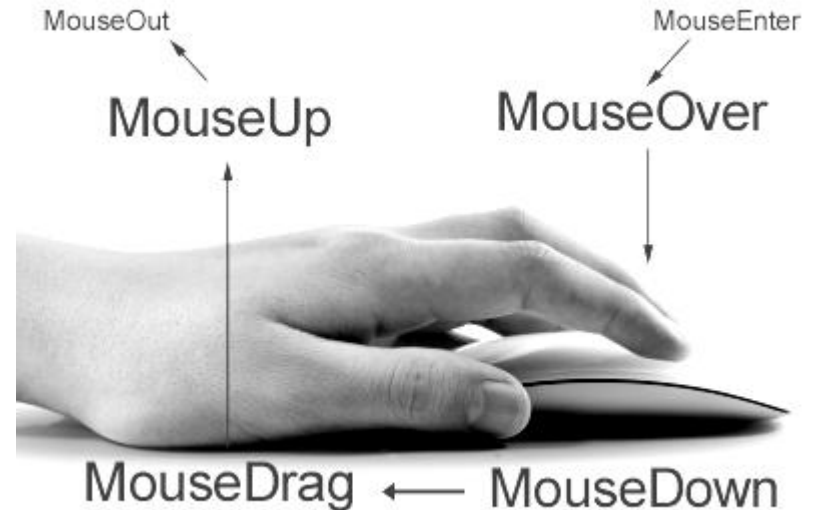
Το αντικείμενο στη Visual Basic.NET

Προσφέρει στον προγραμματιστή της εφαρμογής :

- **Ιδιότητες (properties)** είναι μεταβλητές όπου φυλάσσονται δεδομένα (τιμές) τα οποία εκφράζουν χαρακτηριστικά του αντικειμένου. Ο προγραμματιστής έχει τη δυνατότητα να θέτει-διαβάζει τιμές ιδιοτήτων.
- **Μεθόδους (methods)**. Οι μέθοδοι είναι τμήματα κώδικα διαδικασιών (procedures) που καλούνται από τον προγραμματιστή για εκτέλεση κάποιων λειτου-ργιών. Διακρίνονται σε:
 - **Συναρτήσεις (Functions)** Εκτελούν διάφορους υπολογισμούς και τελικά επιστρέφουν κάποια τιμή .
 - **Υπορουτίνες (Subroutines)**. Εκτελούν διάφορους υπολογισμούς, αλλά δεν επιστρέφουν τιμή.

Συμβάντα & Χειριστές συμβάντων

- Τα συμβάντα χρησιμοποιούνται για την επικοινωνία των αντικειμένων μεταξύ τους. Το συμβάν είναι μια εξειδικευμένη μεταβλητή της οποίας η ύπαρξη μετά από κάποια εξωτερική δράση (action), διεγείρει ένα σινιάλο (signal) που αποστέλλει το αντικείμενο στο Λειτουργικό Σύστημα. Το Λειτουργικό Σύστημα συλλαμβάνει το σινιάλο, το αποκωδικοποιεί και το αποστέλλει σε υπάρχοντα αντικείμενα, σαν ένα **μήνυμα (message)**. Μερικά από τα υπάρχοντα αντικείμενα είναι σε θέση να "συλλάβουν" το μήνυμα και να αντιδράσουν.
- Κάθε συμβάν που διατίθεται από ένα αντικείμενο συνοδεύεται (αυτόματα) από μια **υπορουτίνα χειρισμού του συμβάντος (event handler)**. Το μήνυμα συλλαμβάνεται από τα αντικείμενα που διαθέτουν τούς χειριστές συμβάντων.



<https://www.packtpub.com/books/content/tabula-rasa-nurturing-your-site-tablets>

Συμβάντα & Χειριστές συμβάντων

- Ο Προγραμματιστής της εφαρμογής **δεν ασχολείται με τη μεταβλητή συμβάντος, αλλά με την υπορουτίνα χειρισμού του μηνύματος που συνοδεύει το συμβάν**. Εάν ένα αντικείμενο **B** διαθέτει ρουτίνα χειρισμού συμβάντος που προκαλείται από κάποιο άλλο αντικείμενο **A** (ή και από τον εαυτό του), τότε η συμπλήρωση κώδικα στην υπορουτίνα χειρισμού του συμβάντος, προκαθορίζει το πώς θα συμπεριφερθεί το αντικείμενο **B** στην περίπτωση που του "γνωστοποιηθεί" το μήνυμα από το συμβάν του αντικειμένου **A**.
- Οι δράσεις που διεγείρουν συμβάντα αντικειμένων μπορεί να προκληθούν **από τον χρήστη** της εφαρμογής (π.χ. πιέζοντας ένα πλήκτρο), **από το σύστημα της Visual Basic** (π.χ. όταν εμφανίζει κάποιο παράθυρο στην οθόνη) ή και **από κώδικα** που θέτει ο προγραμματιστής της εφαρμογής για πρόκληση συμβάντος.

Η κλάση στη Visual Basic (1)

- Η Visual Basic προσφέρει στον προγραμματιστή έτοιμες κλάσεις και όχι αντικείμενα. Ο προγραμματιστής μιας εφαρμογής δεν ασχολείται με την κλάση, αλλά με τα αντικείμενα που δημιουργούνται από την κλάση και ειδικότερα με τα μέλη των αντικειμένων που μπορεί να προσπελάσει.
- Βασικά τον προγραμματιστή μιας εφαρμογής ο οποίος θα δημιουργεί ή θα ασχολείται με κλάσεις για να παράγει αντικείμενα, τον ενδιαφέρουν τα προσπελάσιμα (public) μέλη αντικειμένων δηλαδή οι ιδιότητες και μέθοδοι (συναρτήσεις ή υπορουτίνες) καθώς και οι χειριστές συμβάντων που πρέπει να ενημερωθούν στην κλάση.
- Φυσικά για να ασχοληθεί με χειριστές συμβάντων θα πρέπει να έχει υπόψη του τα συμβάντα που προκαλούν τα διάφορα αντικείμενα ή και πως θα δημιουργήσει δικά του συμβάντα.

Η κλάση στη Visual Basic (2)

- Η κλάση είναι μια ενότητα κώδικα με την οποία προσδιορίζεται ένας τύπος αντικειμένων. Τα αντικείμενα είναι περιπτώσεις/στιγμιότυπα (instances) της κλάσης.
- Η κλάση έχει όνομα με το οποίο αναγνωρίζεται από το περιβάλλον της Visual Basic.
- Ο προγραμματιστής δημιουργός μιας κλάσης καθορίζει τα μέλη (members) των αντικειμένων που θα παράγονται από την κλάση.
- Τα μέλη των αντικειμένων είναι μεταβλητές (variables) ή και διαδικασίες (Sub ή Function).
- Τα μέλη της κλάσης χαρακτηρίζονται (από το δημιουργό της κλάσης) σαν **public (δημόσια)** ή **private (ιδιωτικά)**. Τα δημόσια (public) μέλη είναι τα προσπελάσιμα μέλη των αντικειμένων που θα δημιουργηθούν από την κλάση. Τα ιδιωτικά μέλη έχουν υποβοηθητικό ρόλο για την υλοποίηση (implementation) των δημοσίων μελών.

Δημιουργία αντικειμένου

- Η δημιουργία ενός αντικειμένου από μια κλάση προϋποθέτει ότι έχει οριστεί μια **μεταβλητή (variable)** του τύπου της κλάσης και στη μεταβλητή αυτή έχει εκχωρηθεί μια **αναφορά (reference)**, δηλαδή μια διεύθυνση προς χώρο μνήμης που διατίθεται για το αντικείμενο.
- Από μια κλάση μπορούμε να δημιουργούμε οσαδήποτε αντικείμενα του τύπου της κλάσης. Όλα τα αντικείμενα μιας συγκεκριμένης κλάσης έχουν τις ίδιες προγραμματιστικές δυνατότητες, δηλαδή ίδιες ιδιότητες, ίδιες μεθόδους και ίδιες δυνατότητες δημιουργίας συμβάντων (events) και αντίδρασης σε μηνύματα από συμβάντα (event handlers).

Τέλος Ενότητας



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

