



Πανεπιστήμιο Αιγαίου

Εισαγωγή στον Προγραμματισμό Η/Υ

Ενότητα 7: Αντικειμενοστρεφής προγραμματισμός

Μιχάλης Βαΐτης
Τμήμα Γεωγραφίας



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
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 Πανεπιστήμιο Αιγαίου**» έχει χρηματοδοτήσει μόνο τη 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.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Σκοποί ενότητας

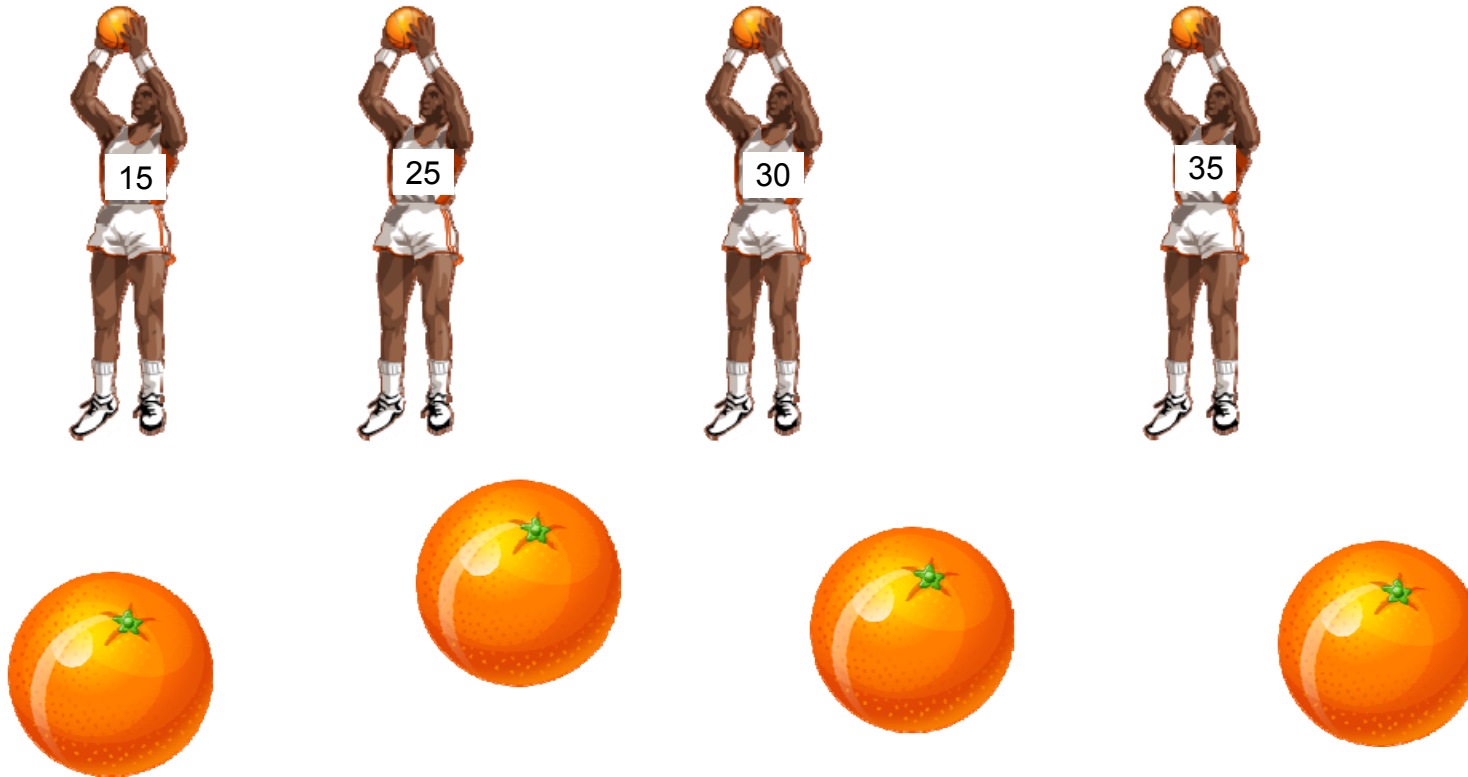
- Να γνωρίζετε τις βασικές έννοιες του αντικειμενοστρεφούς προγραμματισμού (κλάσεις, αντικείμενα, μέθοδοι).
- Να γνωρίζετε την σύνταξη του ορισμού κλάσεων στη γλώσσα R στην έκδοση S4.
- Να μπορείτε να αναζητάτε πληροφορίες για την κατάσταση αντικειμένων και για τις μεθόδους κλάσεων.
- Να μπορείτε να χρησιμοποιείτε αποτελεσματικά αντικείμενα και μεθόδους.

Αντικειμενοστρεφής προγραμματισμός

Στόχοι

- Να γνωρίζετε τις βασικές έννοιες του αντικειμενοστρεφούς προγραμματισμού (κλάσεις, αντικείμενα, μέθοδοι, μηνύματα).
- Να γνωρίζετε την σύνταξη του ορισμού κλάσεων στη γλώσσα R στην έκδοση S4.
- Να μπορείτε να αναζητάτε πληροφορίες για την κατάσταση αντικειμένων και για τις μεθόδους κλάσεων.
- Να μπορείτε να χρησιμοποιείτε αποτελεσματικά αντικείμενα και μεθόδους.

Αντικείμενα (objects)



Στον πραγματικό κόσμο, όλα τα παραπάνω είναι αντικείμενα, αλλά στον προγραμματισμό δεν μπορούμε να διαχειριστούμε αντικείμενα χωρίς «ταυτότητα» (π.χ. τα πορτοκάλια)

Αντικείμενα και Κλάσεις

Αντικείμενο (object): Μια ανεξάρτητη φυσική ή ιδεατή οντότητα του πραγματικού κόσμου. Κάθε αντικείμενο βρίσκεται σε μια *κατάσταση (state)* κάθε χρονική στιγμή και εμφανίζει μια *συμπεριφορά (behavior)*.

Κλάση (class): σύνολο αντικειμένων που έχουν τον ίδιο τύπο (κατηγορήματα κατάστασης) και την ίδια συμπεριφορά.

Φοιτητής
Όνομα : String AM : Integer Τμήμα : String
Reg_lesson (code:Integer): boolean

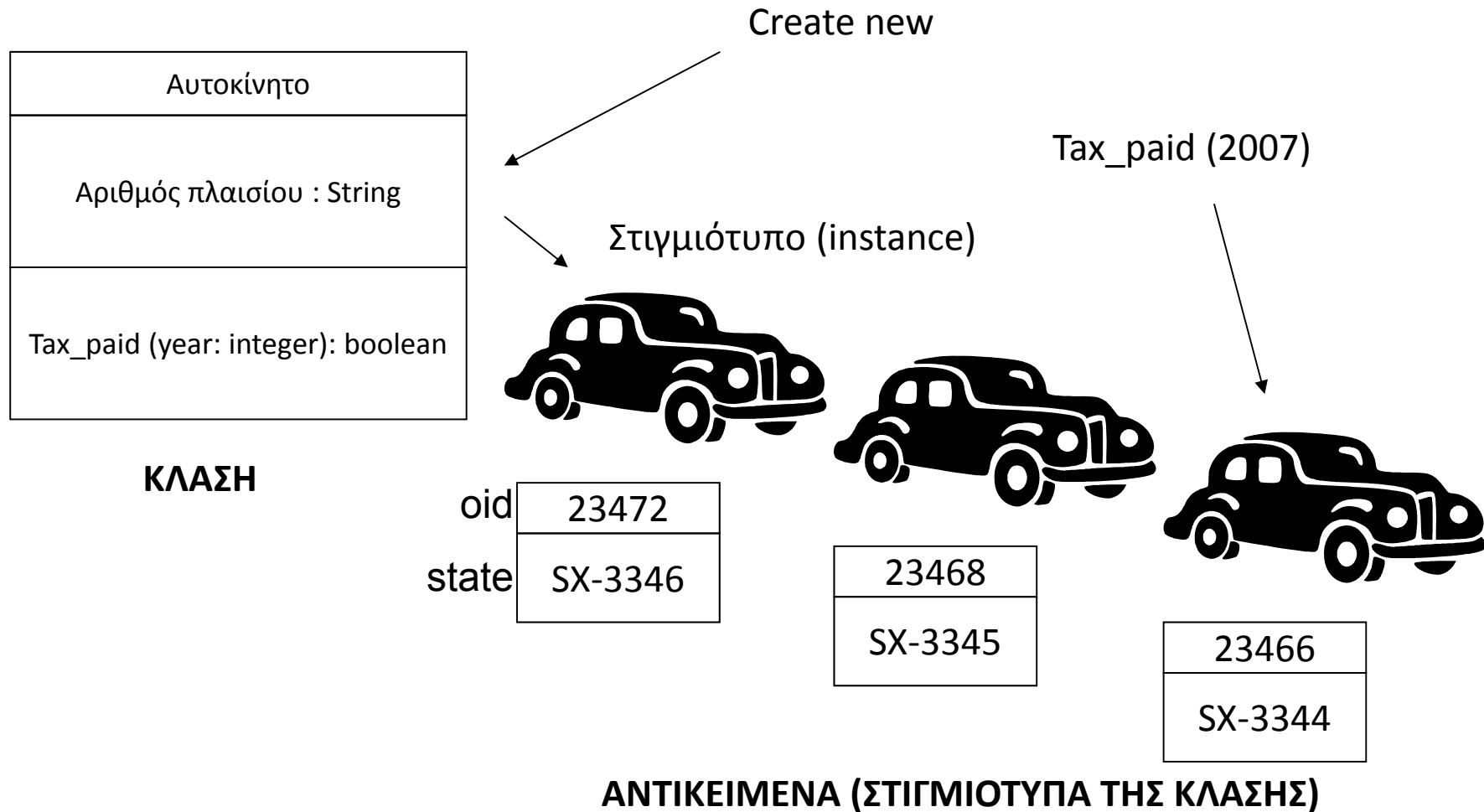
Όνομα κλάσης

Κατηγορήματα
(attributes)
(κατάσταση)

Μέθοδοι (methods)
(συμπεριφορά)

Συμβολισμός κλάσης

Αντικείμενα, Κλάσεις, Μηνύματα



Αντικείμενα, Κλάσεις, Μηνύματα

- Τα αντικείμενα επικοινωνούν μεταξύ τους ανταλλάσσοντας μηνύματά (messages).
- Η αποστολή ενός μηνύματος σε ένα αντικείμενο ισοδυναμεί με την κλήση μιας μεθόδου της κλάσης που ανήκει το αντικείμενο.
- Οι κλάσεις των αντικειμένων είναι και αυτές με τη σειρά τους αντικείμενα (που είναι στιγμιότυπα Μετακλάσης).

Στο παράδειγμα:

- Κάθε φορά που η κλάση «Αυτοκίνητο» δέχεται το μήνυμα «Create new», ενεργοποιείται κατάλληλη μέθοδος της Μετακλάσης, που δημιουργεί ένα νέο αντικείμενο (ένα καινούργιο αυτοκίνητο) και αρχικοποιεί την κατάστασή του (αποδίδεται object identification [oid] και τιμή στο κατηγορημα «Αριθμός_πλαισίου»).
- Κάθε φορά που ένα συγκεκριμένο αυτοκίνητο δέχεται το μήνυμα «Tax_paid», τότε ενεργοποιείται κατάλληλη μέθοδος της κλάσης «Αυτοκίνητο», που απαντά αν έχουν πληρωθεί τα τέλη κυκλοφορίας για το συγκεκριμένο αυτοκίνητο για το έτος που αναφέρεται ως παράμετρος στο μήνυμα.

Ενθυλάκωση (encapsulation)

Η έννοια της ενθυλάκωσης έχει πολλές εκφάνσεις στον αντικειμενοστραφή προγραμματισμό:

- Περιγράφει την από κοινού ομαδοποίηση των κατηγορημάτων και των μεθόδων των αντικειμένων, μέσω των κλάσεων.
- Περιγράφει την απόκρυψη των κατηγορημάτων μιας κλάσης, ή/και την αδυναμία χειρισμού των τιμών τους απευθείας.
- Περιγράφει την απόκρυψη ορισμένων μεθόδων μιας κλάσης, η/και του κώδικα που υλοποιεί τις μεθόδους αυτές.

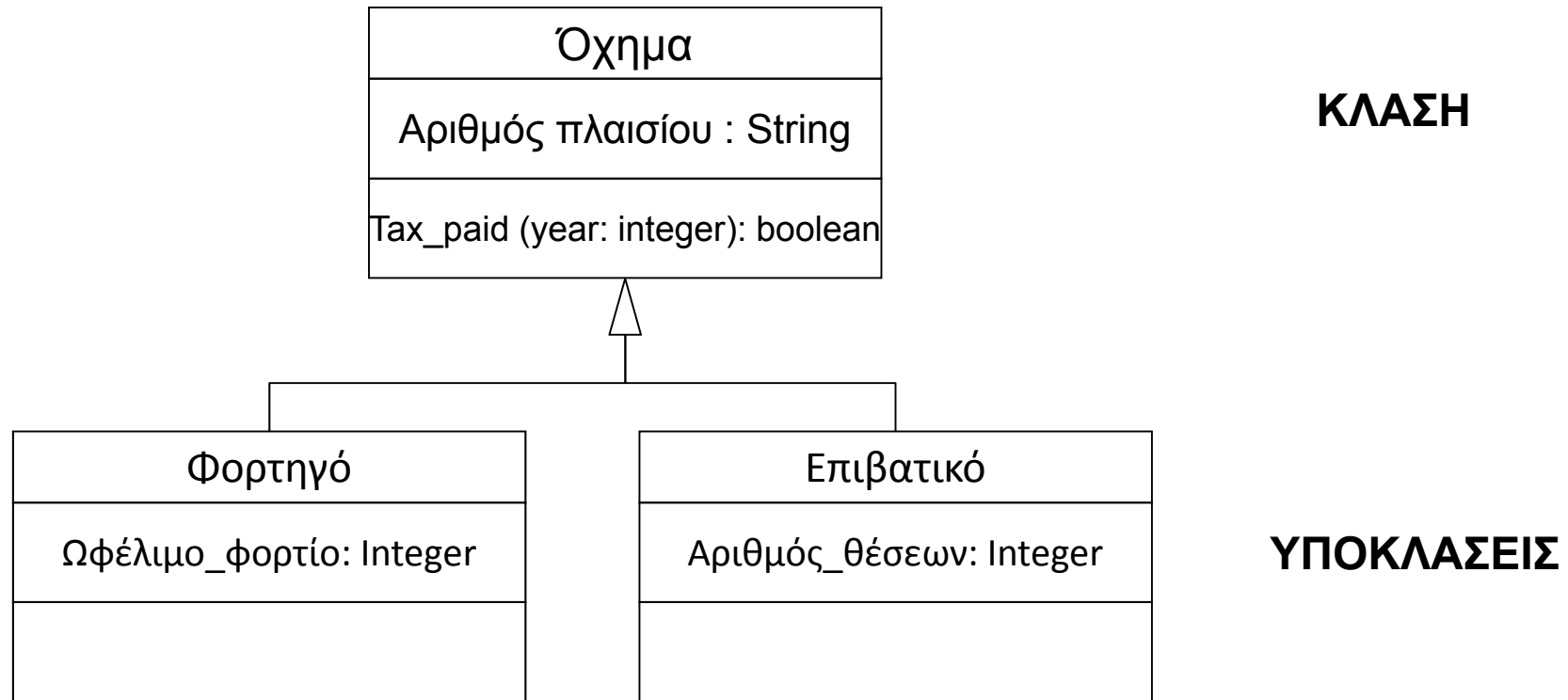
Με την ενθυλάκωση, παρέχεται ελεγχόμενη πρόσβαση στα κατηγορήματα και τις μεθόδους των κλάσεων και με τον τρόπο αυτό προστατεύονται οι προγραμματιστές από λάθη.

Στο παράδειγμα:

Στην κλάση «Φοιτητής», δεν υπάρχει απευθείας πρόσβαση στο κατηγορήμα των μαθημάτων που έχει δηλώσει. Αυτό γίνεται μέσω της μεθόδου «Reg_lesson», η οποία ελέγχει αν ο φοιτητής μπορεί δηλώσει το μάθημα που αντιστοιχεί στην παράμετρο εισόδου (code). Αν ο έλεγχος επιτύχει, το μάθημα δηλώνεται και η μέθοδος επιστρέφει TRUE, αλλιώς το μάθημα δεν δηλώνεται και η μέθοδος επιστρέφει FALSE.

Κληρονομικότητα (inheritance)

Η κληρονομικότητα επιτρέπει τη δημιουργία μιας νέας κλάσης ως «υποκλάση» (εξειδίκευση) μιας ήδη υπάρχουσας κλάσης. Η υποκλάση «κληρονομεί» όλα τα κατηγορήματα και τις μεθόδους της υπάρχουσας κλάσης, ενώ μπορούν να προστεθούν επιπλέον κατηγορήματα ή μέθοδοι.



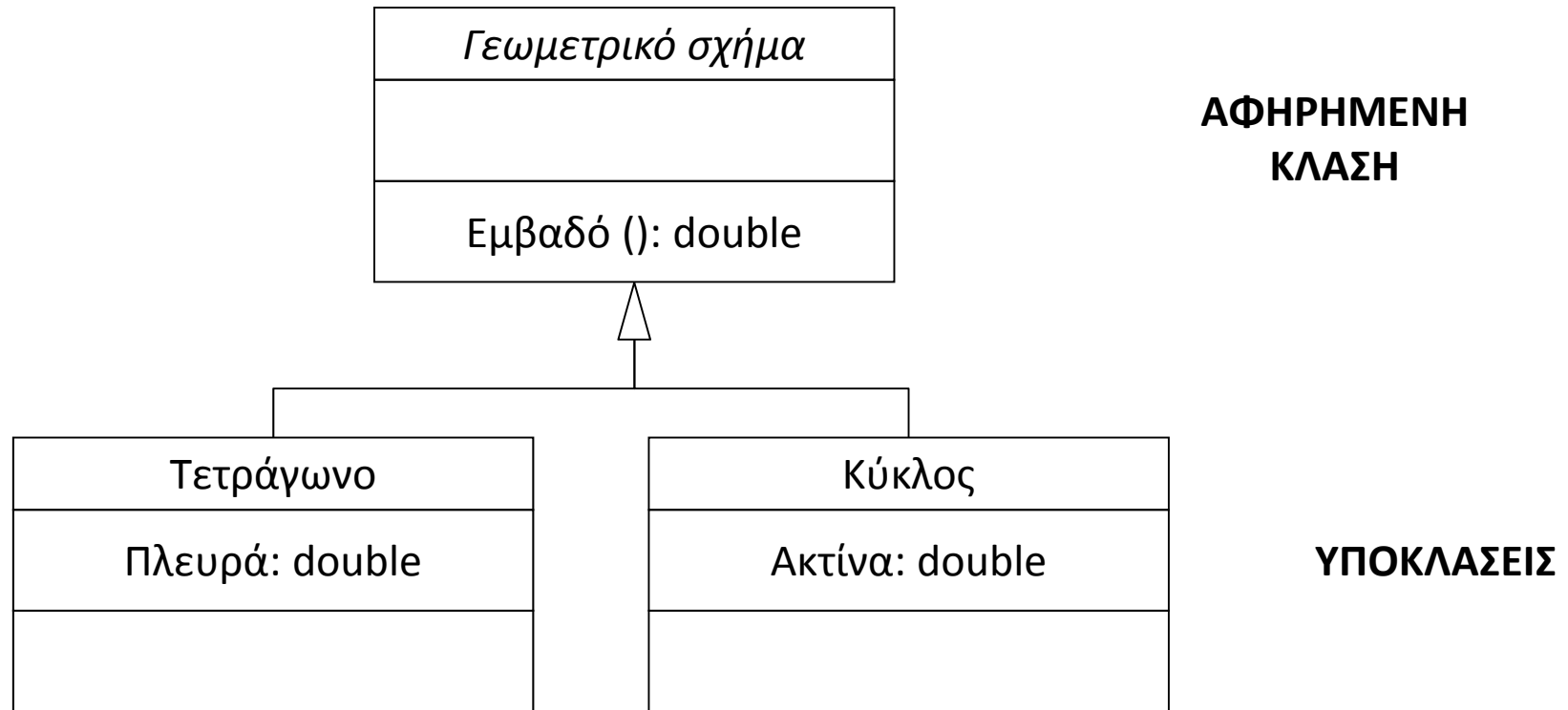
Πολυμορφισμός (polymorphism)

Πολυμορφισμός καλείται η ιδιότητα μιας γλώσσας προγραμματισμού να επιτρέπει τη διαφορετική συμπεριφορά αντικειμένων (ανάλογα σε ποια κλάση ανήκουν) με την ενεργοποίηση της ίδιας κατ' όνομα μεθόδου.

Για παράδειγμα, μπορούμε να δηλώσουμε τη μέθοδο *Εμβαδό* για τη γενική κλάση *Γεωμετρικό_σχήμα*. Στη συνέχεια, μπορούμε να ορίσουμε τις υποκλάσεις *Τετράγωνο* και *Κύκλος*. Οι υποκλάσεις κληρονομούν τη μέθοδο *Εμβαδό*, αλλά η λειτουργία της μεθόδου διαφοροποιείται ανάλογα με την υποκλάση (αλλιώς υπολογίζεται το εμβαδό στα τετράγωνα και αλλιώς στους κύκλους). Έτσι, η κλήση της μεθόδου *Εμβαδό* (μήνυμα) θα εκτελέσει τον κατάλληλο υπολογισμό, ανάλογα αν το αντικείμενο που θα λάβει το μήνυμα ανήκει στην υποκλάση των τετραγώνων ή των κύκλων.

Αφηρημένες κλάσεις (abstract classes)

Αφηρημένες καλούνται οι κλάσεις που δεν δημιουργούν αντικείμενα, αλλά χρησιμοποιούνται για τη δημιουργία ιεραρχιών από υποκλάσεις. Στις αφηρημένες κλάσεις είναι δυνατή η δήλωση κατηγορημάτων και μεθόδων, με τη διαφορά ότι οι μέθοδοι δεν περιέχουν υλοποίηση (λειτουργία) αλλά μόνο ονομασία. Η λειτουργία κάθε μεθόδου πρέπει να εξειδικευτεί από κάθε υποκλάση ξεχωριστά.



Αντικειμενοστρεφής προγραμματισμός στην R

- Στην R, ο αντικειμενοστρεφής προγραμματισμός υποστηρίζεται από το λεγόμενο «**S4 class system**».
- Η υλοποίησή του γίνεται στο πακέτο «**methods**».
- Συνυπάρχει ακόμα και το παλαιότερο «**S3 class system**»
- Από πολλούς προγραμματιστές, η R δεν θεωρείται γνήσια αντικειμενοστρεφής γλώσσα προγραμματισμού, αλλά συμβατική (διαδικαστική), που έχει ενσωματώσει αντικειμενοστρεφή χαρακτηριστικά.

Αντικειμενοστρεφής προγραμματισμός στην R

Τα στοιχεία που συγκροτούν τα αντικειμενοστρεφή χαρακτηριστικά στην R είναι:

- Οι κλάσεις (classes) [και οι αφηρημένες κλάσεις (VIRTUAL)]
- Οι γενικές συναρτήσεις (generic functions)
- Οι μέθοδοι (methods)

Η αποστολή μηνυμάτων στα αντικείμενα (δηλαδή η κλήση μεθόδων) καλείται *method dispatch*. Σύμφωνα με τη διαδικασία αυτή, για κάθε μέθοδο πρέπει να υπάρχει αντίστοιχη γενική συνάρτηση με το ίδιο όνομα. Όταν καλείται η γενική συνάρτηση, ανάλογα με την κλάση του αντικειμένου που δέχεται το μήνυμα, γίνεται η κλήση της κατάλληλης μεθόδου (πολυμορφισμός).

Κλάσεις και αντικείμενα στην R

- Η δημιουργία μιας νέας κλάσης γίνεται με τη συνάρτηση `setClass()`.
- Βασικοί παράμετροι στη συνάρτηση `setClass()` αποτελούν το όνομα της κλάσης και οι μεταβλητές που αποθηκεύουν την κατάσταση των αντικειμένων της κλάσης.
- Οι μεταβλητές αυτές στην R καλούνται *slots*.
- Η δημιουργία μιας νέας μεθόδου γίνεται με τη συνάρτηση `setMethod()`, αφού πρώτα δηλωθεί αντίστοιχη *generic function* (αν δεν υπάρχει ήδη).
- Παράμετροι στη συνάρτηση `setMethod()` αποτελούν το όνομα της μεθόδου, το όνομα της κλάσης και η δήλωση της συνάρτησης (function) που υλοποιεί τις λειτουργίες της μεθόδου.
- Η δημιουργία ενός νέου αντικειμένου γίνεται με τη συνάρτηση `new()`.
- Παράμετροι στη συνάρτηση αποτελούν το όνομα της κλάσης και η απόδοση τιμών στα *slots* της κλάσης.

Δημιουργία νέας κλάσης

```
setClass("όνομα κλάσης",  
        representation(όνομα_κατηγορήματος="data_type",  
                        όνομα_κατηγορήματος="data_type",  
                        ...,  
                        "VIRTUAL"),  
        prototype(όνομα_κατηγορήματος=τιμή,  
                 όνομα_κατηγορήματος=τιμή,  
                 ...))  
contains=("υπερκλάση", ...)  
)
```

Δημιουργία νέας κλάσης

<code>"όνομα κλάσης"</code> :	Συμβολοσειρά για το όνομα της κλάσης
<code>όνομα_κατηγορήματος</code> :	Όνομα μεταβλητής που αναπαριστά ένα κατηγορημα (slot) της κλάσης
<code>"data_type"</code> :	Ο τύπος δεδομένων του κατηγορήματος <code>"VIRTUAL"</code> : Δηλώνει ότι η κλάση είναι αφηρημένη
<code>prototype</code> :	Αρχικές (default) τιμές για τα κατηγορήματα της κλάσης, αν δεν δοθούν κατά τη δημιουργία των αντικειμένων
<code>contains</code> :	Συμβολοσειρές με τα ονόματα των κλάσεων που η συγκεκριμένη κλάση εξειδικεύει [δηλ. είναι υποκλάση] (multiple inheritance)

Δημιουργία νέου αντικειμένου

```
Όνομα_μεταβλητής = new("όνομα κλάσης",  
                        όνομα_κατηγορήματος=τιμή,  
                        όνομα_κατηγορήματος=τιμή,  
                        ...)
```

"όνομα κλάσης": Συμβολοσειρά με το όνομα της κλάσης για την οποία δημιουργείται το νέο αντικείμενο

όνομα_κατηγορήματος: Το όνομα του κατηγορήματος για το οποίο θέτουμε κατά τη δημιουργία του αντικειμένου. Για όσα κατηγορήματα δεν δοθεί τιμή, θα τεθεί αυτή που έχει δηλωθεί στο τμήμα prototype της κλάσης.

Δημιουργία νέας μεθόδου

Αρχικά δημιουργείται η γενική συνάρτηση (αν δεν υπάρχει)

```
setGeneric (name="όνομα μεθόδου",  
            def=function(object) {standardGeneric("όνομα μεθόδου") }  
            )
```

"όνομα μεθόδου":

Συμβολοσειρά με το όνομα της μεθόδου

def:

Δήλωση της μεθόδου που δεν περιλαμβάνει υλοποίηση (λειτουργία), αλλά μόνο την κλήση της συνάρτησης `standardGeneric()`. Η συνάρτηση αυτή υλοποιεί τη διαδικασία `method dispatch`.

object

Υποδηλώνει το αντικείμενο για το οποίο θα γίνει η κλήση της γενικής συνάρτησης. Ανάλογα με την

κλάση του αντικειμένου, θα γίνει κλήση της κατάλληλης μεθόδου.

Δημιουργία νέας μεθόδου

Στη συνέχεια δημιουργείται μια μέθοδος για κάθε κλάση που θέλουμε να υλοποιεί τη γενική συνάρτηση

```
setMethod(  
    f= "όνομα_μεθόδου",  
    signature= "όνομα_κλάσης",  
    definition= function(object) {  
        κώδικας που υλοποιεί τη λειτουργία της μεθόδου  
        return()  
    }  
)
```

Δημιουργία νέας μεθόδου

“όνομα μεθόδου”:

Συμβολοσειρά με το όνομα της μεθόδου

“όνομα κλάσης”:

Συμβολοσειρά με το όνομα της κλάσης για την οποία υλοποιούμε τη μέθοδο

`return`

Είναι απαραίτητη η επιστροφή τιμής.

Τέλος Διάλεξης



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

