



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΠΑΤΡΩΝ  
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά  
μαθήματα ΠΠ

# Οντοκεντρικός Προγραμματισμός

Ενότητα 6: C++ ΚΛΑΣΕΙΣ, ΚΛΗΡΟΝΟΜΙΚΟΤΗΤΑ, ΠΟΛΥΜΟΡΦΙΣΜΟΣ

## Κληρονομικότητα

ΔΙΔΑΣΚΟΝΤΕΣ: Ιωάννης Χατζηλυγερούδης, Χρήστος Μακρής

Πολυτεχνική Σχολή

Τμήμα Μηχανικών Η/Υ & Πληροφορικής

# Κληρονομικότητα

# Κληρονομικότητα

---

- Η κληρονομικότητα αφορά
  - Επαναχρησιμοποίηση λογισμικού
  - Δημιουργία νέας κλάσης (**παραγόμενη κλάση**) από μια αρχική υπάρχουσα κλάση (**κλάση βάσης**)
    - Κληρονομούνται τα χαρακτηριστικά και η συμπεριφορά της αρχικής κλάσης
    - Επέκταση της παραγόμενης κλάσης με νέες δυνατότητες (customization)
      - νέα πεδία
      - επιπρόσθετη συμπεριφορά



# Τύποι Κληρονομικότητας

- Υποστηρίζονται 3 τύποι κληρονομικότητας
  - **public**
    - Κάθε αντικείμενο μιας παραγόμενης κλάσης είναι αντικείμενο και της κλάσης βάσης
      - Αντικείμενα μιας κλάσης βάσης δεν είναι αντικείμενα της παραγόμενης κλάσης
      - Παράδειγμα: Όλα τα αυτοκίνητα είναι οχήματα, αλλά δεν ισχύει το αντίστροφο
    - Επιτρέπεται η προσπέλαση των μη-**private** μελών της κλάσης βάσης
      - Η παραγόμενη κλάση μπορεί να επιφέρει αλλαγές στα **private** μέλη της κλάσης βάσης, μέσω κληρονομούμενων μη-**private** μεθόδων
  - **private**
    - Αντίστοιχη με τη σχέση σύνθεσης
  - **protected**
    - Χρησιμοποιείται σπάνια



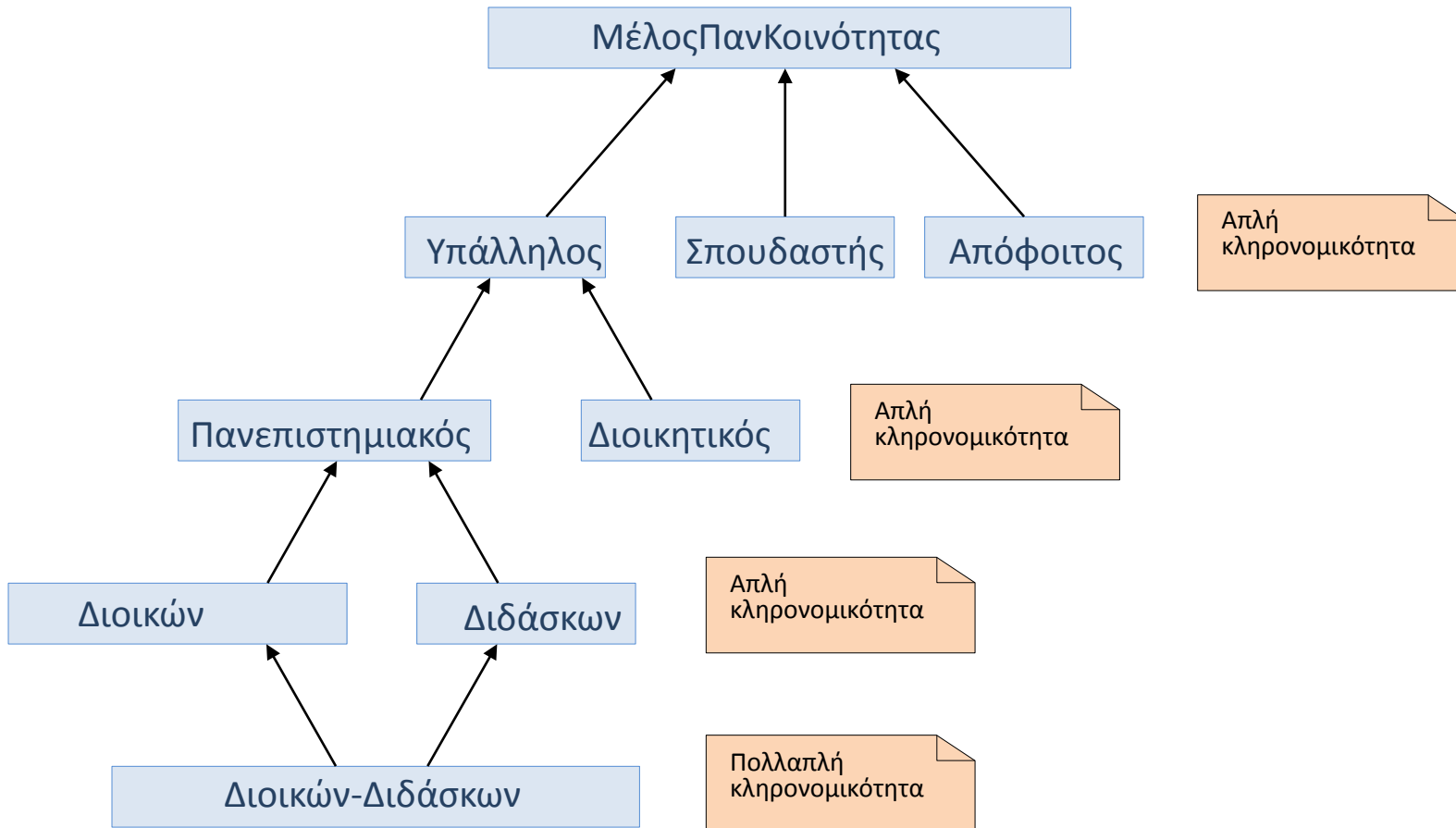
# Ιεραρχία κλάσεων

---

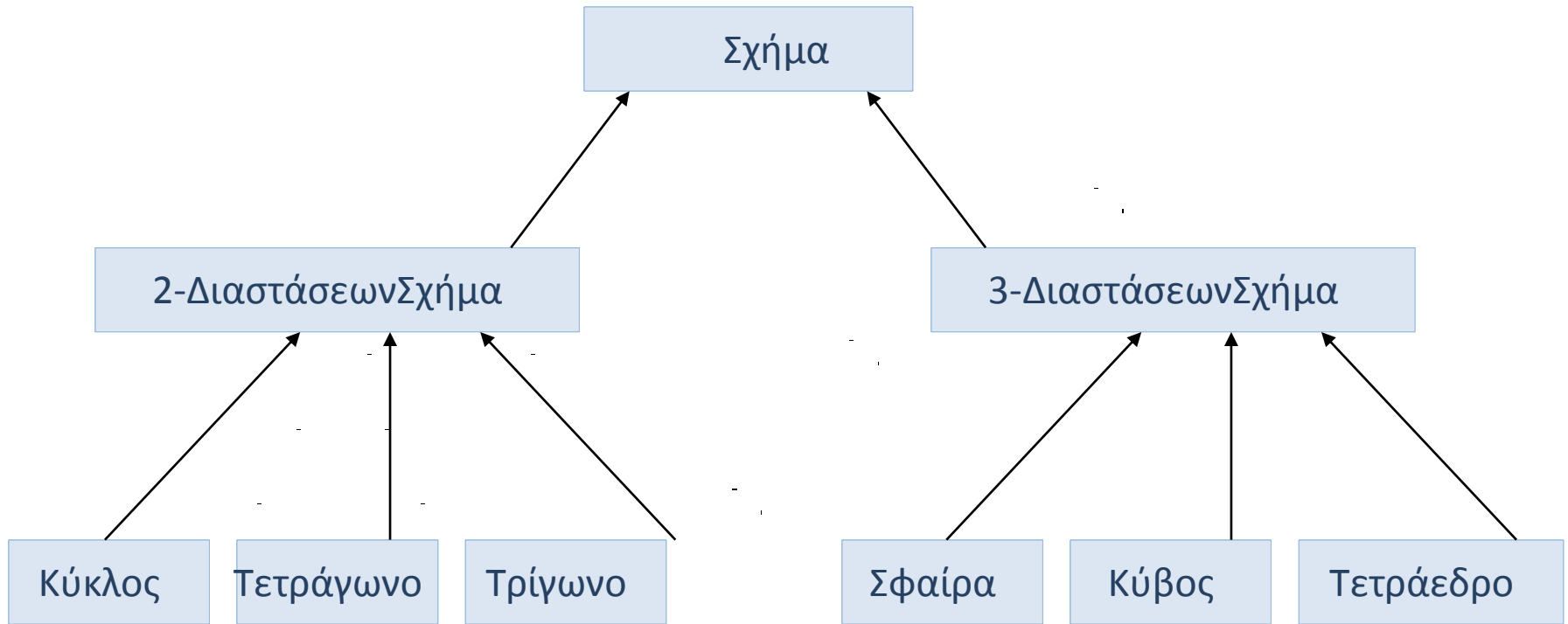
- Άμεση κλάση βάσης
  - Κληρονομείται απ' ευθείας (ιεραρχία ενός επιπέδου)
- Έμμεση κλάση βάσης
  - Κληρονομείται σε ιεραρχία δύο ή περισσότερων επιπέδων
- Απλή κληρονομικότητα
  - Μία παραγόμενη κλάση συνδέεται με μία μόνο κλάση βάσης
- Πολλαπλή κληρονομικότητα
  - Μία παραγόμενη κλάση συνδέεται με πολλές κλάσεις βάσης
  - Να χρησιμοποιείται με προσοχή



# Παράδειγμα Ιεραρχίας



# Παράδειγμα Ιεραρχίας



# Κληρονομικότητα Public

- **public** τύπος κληρονομικότητας

- Ορίζεται με την εντολή:

```
Class TwoDimensionalShape : public Shape
```

- Προσπέλαση των **private** μελών της κλάσης βάσης

- Δεν μπορεί να γίνει απ' ευθείας
- Παρόλα αυτά τα private μέλη κληρονομούνται και μπορούμε να τα χειριστούμε μέσω των μη-private κληρονομούμενων μεθόδων

- Προσπέλαση των **public** και **protected** μελών της κλάσης βάσης

- Κληρονομούνται και είναι δυνατή η απ' ευθείας προσπέλασή τους (με χρήση του ονόματος του μέλους)

- **friend** functions

- Δεν κληρονομούνται





# Protected Μέλη

---

- **protected** προσπέλαση
  - Ενδιάμεσο επίπεδο προστασίας δεδομένων μεταξύ **public** και **private**
  - Η προσπέλαση των **protected** μελών είναι εφικτή σε:
    - μέλη της κλάσης βάσης
    - **friends** της κλάσης βάσης
    - μέλη της παραγόμενης κλάσης
    - **friends** της παραγόμενης κλάσης



# Παράδειγμα

```
class myDate{
    public: myDate(int d, int m, int y) :day(d), month(m), year(y) {}
    void print(){
        cout << year <<"/"<<month<<"/"<<day;
    }
    protected:
        int day, month, year;
};
```

Η myDateTime κληρονομεί την myDate

```
class myDateTime : public myDate{
    public:
        myDateTime(int d, int m, int y, int se, int mi, int ho)
            : myDate(d,m,y), sec(se), minute(mi), hour(ho) {}
        void print(){
            myDate::print();
            cout <<" - " << hour <<":" <<minute<<":" <<sec;
        }
        void incrementDay(){
            day++;
        }
    private:
        int sec, minute, hour;
};
```

Δεν υπάρχει super όπως στην Java

- Κλήση του Δημιουργού της myDate
- Κλήση της print της myDate

Άμεση πρόσβαση στα protected μέλη της κλάσης myDate



# Χρήση Protected Μελών

- Χρήση **protected** μελών

- + **Πλεονεκτήματα**

- Οι παραγόμενες κλάσεις μπορούν να αλλάξουν τις τιμές των πεδίων απ' ευθείας
    - Μικρή βελτίωση της ταχύτητας
      - Αποφεύγουμε την κλήση των μεθόδων set/get

- **Μειονεκτήματα**

- Δεν προσφέρεται για έλεγχο εγκυρότητας τιμών
      - Η παραγόμενη κλάση μπορεί να δώσει μη-επιτρεπτή τιμή
    - Δημιουργία σχέσεων εξάρτησης
      - Οι μέθοδοι της παραγόμενης κλάσης είναι πιο πιθανόν τώρα να εξαρτώνται από την υλοποίηση της κλάσης βάσης
      - Εάν αλλάξει η υλοποίηση της κλάσης βάσης μπορεί να χρειαστεί να τροποποιήσουμε και την παραγόμενη κλάση
        - » Εύθραυστο λογισμικό



# Δημιουργία αντικειμένου

---

- Δημιουργία αντικειμένου παραγόμενης κλάσης
  - Σειρά κλήσης των constructors
    - Ο constructor της παραγόμενης κλάσης καλεί τον constructor της κλάσης βάσης
      - Έμμεσα ή άμεσα
    - Βάση της ιεραρχίας κληρονομικότητας
      - Ο τελευταίος στη σειρά constructor που καλείται και ο πρώτος που ολοκληρώνει την εκτέλεσή του
    - Αρχικοποίηση των πεδίων
      - Ο constructor κάθε κλάσης βάσης αρχικοποιεί τα πεδία του



# Καταστροφή αντικειμένου

---

- Καταστροφή αντικειμένου παραγόμενης κλάσης
  - Σειρά κλήσης των destructors
    - Αντίστροφη σειρά από αυτή της κλήσης των constructors
    - Ο destructor της παραγόμενης κλάσης καλείται πρώτος
    - Ο destructor της επόμενης κλάσης βάσης στην ιεραρχία καλείται στη συνέχεια
    - Συνεχίζουμε προς τα πάνω μέχρι να φθάσουμε στην τελευταία κλάση βάσης της ιεραρχίας
      - Μετά την εκτέλεση του τελευταίου destructor, το αντικείμενο αφαιρείται από τη μνήμη



# Κληρονομικότητα

---

- Constructors, destructors, τελεστές ανάθεσης της κλάσης βάσης
  - Δεν κληρονομούνται από τις παραγόμενες κλάσεις
  - Οι constructors, τελεστές ανάθεσης της παραγόμενης κλάσης μπορούν να καλέσουν
    - Constructors
    - Τελεστές ανάθεσης



# Κληρονομικότητα

## public, protected και private

Προσδιοριστής προσπέλασης μέλους κλάσης βάσης	Τύπος κληρονομικότητας		
	<b>public</b> κληρονομικότητα	<b>protected</b> κληρονομικότητα	<b>private</b> κληρονομικότητα
<b>Public</b>	<b>public</b> στην παραγόμενη κλάση. Μπορεί να προσπελασθεί απ' ευθείας από όλες τις μη- <b>static</b> μεθόδους, <b>friend</b> και εξωτερικές συναρτήσεις.	<b>protected</b> στην παραγόμενη κλάση. Μπορεί να προσπελασθεί απ' ευθείας από όλες τις μη- <b>static</b> μεθόδους και <b>friend</b> συναρτήσεις.	<b>private</b> στην παραγόμενη κλάση. Μπορεί να προσπελασθεί απ' ευθείας από όλες τις μη- <b>static</b> μεθόδους και <b>friend</b> συναρτήσεις.
<b>Protected</b>	<b>protected</b> στην παραγόμενη κλάση. Μπορεί να προσπελασθεί απ' ευθείας από όλες τις μη- <b>static</b> μεθόδους και <b>friend</b> συναρτήσεις.	<b>protected</b> στην παραγόμενη κλάση. Μπορεί να προσπελασθεί απ' ευθείας από όλες τις μη- <b>static</b> μεθόδους και <b>friend</b> συναρτήσεις.	<b>private</b> στην παραγόμενη κλάση. Μπορεί να προσπελασθεί απ' ευθείας από όλες τις μη- <b>static</b> μεθόδους και <b>friend</b> συναρτήσεις.
<b>Private</b>	Κρυφή στην παραγόμενη κλάση. Μπορεί να προσπελασθεί από μη- <b>static</b> μεθόδους και <b>friend</b> συναρτήσεις μέσω <b>public</b> ή <b>protected</b> μεθόδους της κλάσης βάσης.	Κρυφή στην παραγόμενη κλάση. Μπορεί να προσπελασθεί από μη- <b>static</b> μεθόδους και <b>friend</b> συναρτήσεις μέσω <b>public</b> ή <b>protected</b> μεθόδους της κλάσης βάσης.	Κρυφή στην παραγόμενη κλάση. Μπορεί να προσπελασθεί από μη- <b>static</b> μεθόδους και <b>friend</b> συναρτήσεις μέσω <b>public</b> ή <b>protected</b> μεθόδους της κλάσης βάσης.

# Πρόσθετο Υλικό

---

- Μελετήστε και τα παραδείγματα από τα **Κεφάλαιο 11** του βιβλίου:  
«C++ How to Program, 9/e Paul & Harvey Deitel»  
[http://media.pearsoncmg.com/ph/esm/deitel/cpp\\_hpt\\_9/code\\_examples/Code\\_Examples.zip](http://media.pearsoncmg.com/ph/esm/deitel/cpp_hpt_9/code_examples/Code_Examples.zip)





# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» 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.



# Σημείωμα Ιστορικού Εκδόσεων Έργου

---

Το παρόν έργο αποτελεί την έκδοση 1.0.1



# Σημείωμα Αναφοράς

---

Copyright: Πανεπιστήμιον Πατρών, Ιωάννης Χατζηλυγερούδης, 2015.  
«Οντοκεντρικός Προγραμματισμός». Έκδοση: 1.0.1 Πάτρα 2015. Διαθέσιμο  
από τη δικτυακή διεύθυνση:

<https://eclass.upatras.gr/courses/CEID1105/>



# Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

# Διατήρηση Σημειωμάτων

---

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.



# Σημείωμα Χρήσης Έργων Τρίτων

---

- Οι διαφάνειες βασίζονται στο βιβλίο «C++ How to Program, 8th Edition, Harvey M. Deitel, Paul J. Deitel, Prentice Hall.»

