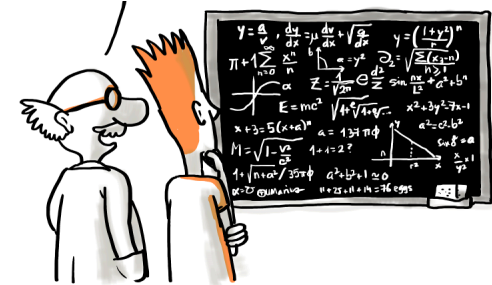


Module 3

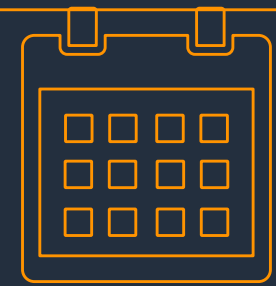
Κληρονομικότητα



Σύνοψη



- **Τι έχουμε μάθει μέχρι τώρα;**
 - Βασικά στοιχεία Α/Σ Π και προγραμματισμού Java.
 - Βασικά στοιχεία διαχείρισης μνήμης.
 - Αρχές Αφαιρετικότητας στον Α/Σ Προγραμματισμό και υλοποίηση τους στη Java.
 - Αναλλοίωτοι περιορισμοί (invariants).
- **Τι θα δούμε στη συνέχεια;**
 - Επαναχρησιμοποίηση κώδικα
 - Σύνθεση κλάσεων
 - Κληρονομικότητα
 - Βιβλιοθήκες
 - UML



Ενότητα 2: **Διαχείριση Μνήμης** και Σχεδιασμός Κλάσεων

Μάθημα 13β: 5/3/2024

Επαναχρησιμοποίηση κώδικα
Κληρονομικότητα



Σύνθεση και Κληρονομικότητα



Περίγραμμα



- Σύνθεση Κλάσεων
- Εισαγωγή στην Κληρονομικότητα
- Παράδειγμα κληρονομικότητας
- Αναίρεση και επισκίαση
- Κατασκευή αντικειμένων και κληρονομικότητα
- Σύνθεση και κληρονομικότητα
- Αντιπροσώπευση
- Ο τροποποιητής final
- Ενθυλάκωση (encapsulation) και Κληρονομικότητα





Topic 3: Σύνθεση και Κληρονομικότητα

Σύνθεση Κλάσεων



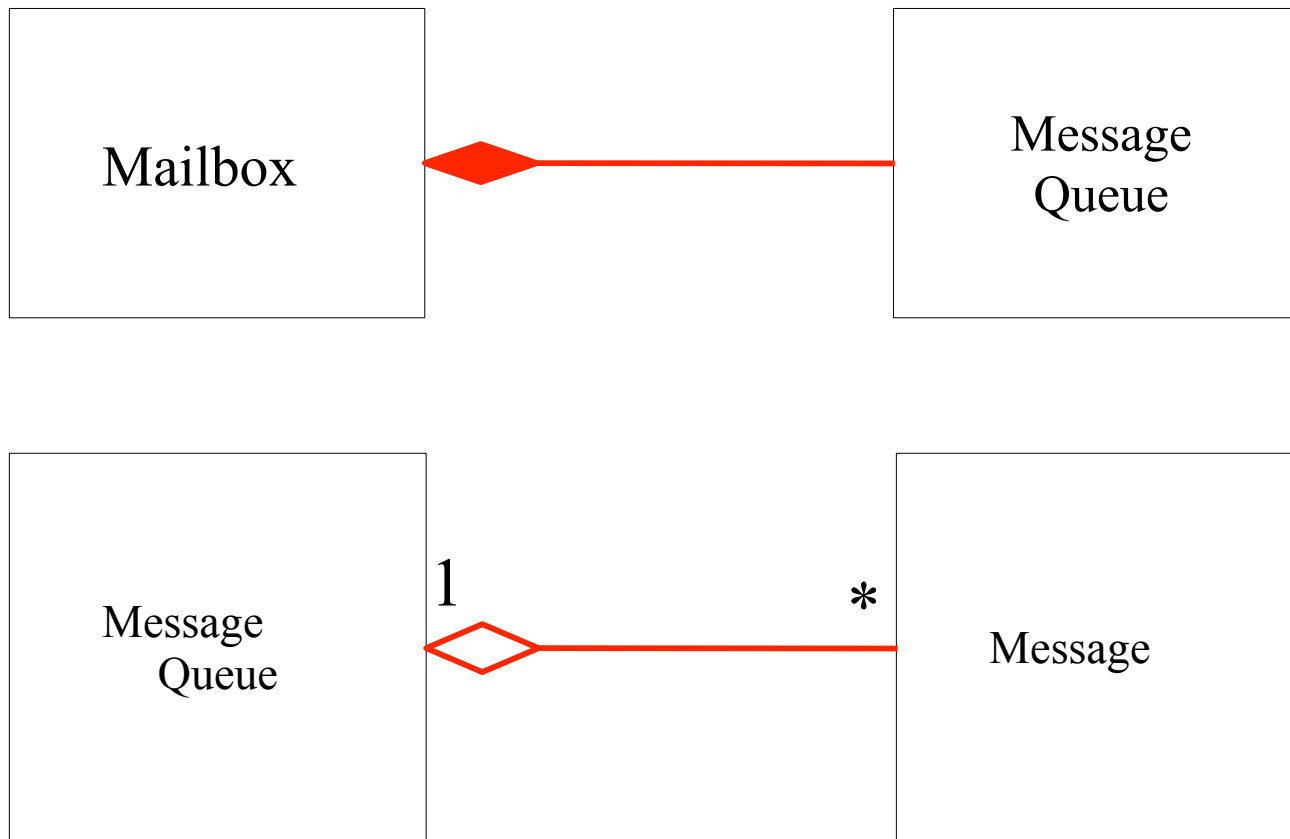
ΠΩΣ ΕΠΑΝΑΧΡΗΣΙΜΟΠΟΙΟΥΜΕ ΚΩΔΙΚΑ JAVA;



University of Cyprus
Department of Computer Science

Επαναχρησιμοποίηση Κλάσεων

- **Σύνθεση** (composition): σε μια κλάση εντάσσουμε ως στοιχεία χειριστήρια άλλων κλάσεων.



Σύνθεση

```
class WaterSource {  
    private String s;  
    WaterSource() {  
        System.out.println("WaterSource()");  
        s = new String("Constructed");  
    }  
    public String toString() {  
        return s;  
    }  
}
```

Η χρήση του **toString()**: όταν θέλουμε να μπορεί να γίνει μετατροπή ενός αντικειμένου μιας κλάσης σε συμβολοσειρά, δεν έχουμε παρά να ορίσουμε στην κλάση μια μέθοδο με όνομα **toString()**.

Σύνθεση (παράδειγμα)

```
public class SprinklerSystem {  
    private String valve1, valve2;  
    WaterSource source;  
    float f;  
    void print() {  
        System.out.println("valve1 =" + valve1);  
        System.out.println("valve2 =" + valve2);  
        System.out.println("f = " + f);  
        System.out.println("source =" + source);  
    }  
    public static void main (String[] args) {  
        SprinklerSystem x = new SprinklerSystem();  
        x.print();  
    }  
}
```

Περίγραμμα



- Σύνθεση Κλάσεων
- **Εισαγωγή στην Κληρονομικότητα**
- Παράδειγμα κληρονομικότητας
- Αναίρεση και επισκίαση
- Κατασκευή αντικειμένων και κληρονομικότητα
- Σύνθεση και κληρονομικότητα
- Αντιπροσώπευση
- Ο τροποποιητής final
- Ενθυλάκωση (encapsulation) και Κληρονομικότητα





Topic 3: Σύνθεση και Κληρονομικότητα

Βασικά στοιχεία κληρονομικότητας

Κληρονομικότητα: Εισαγωγή

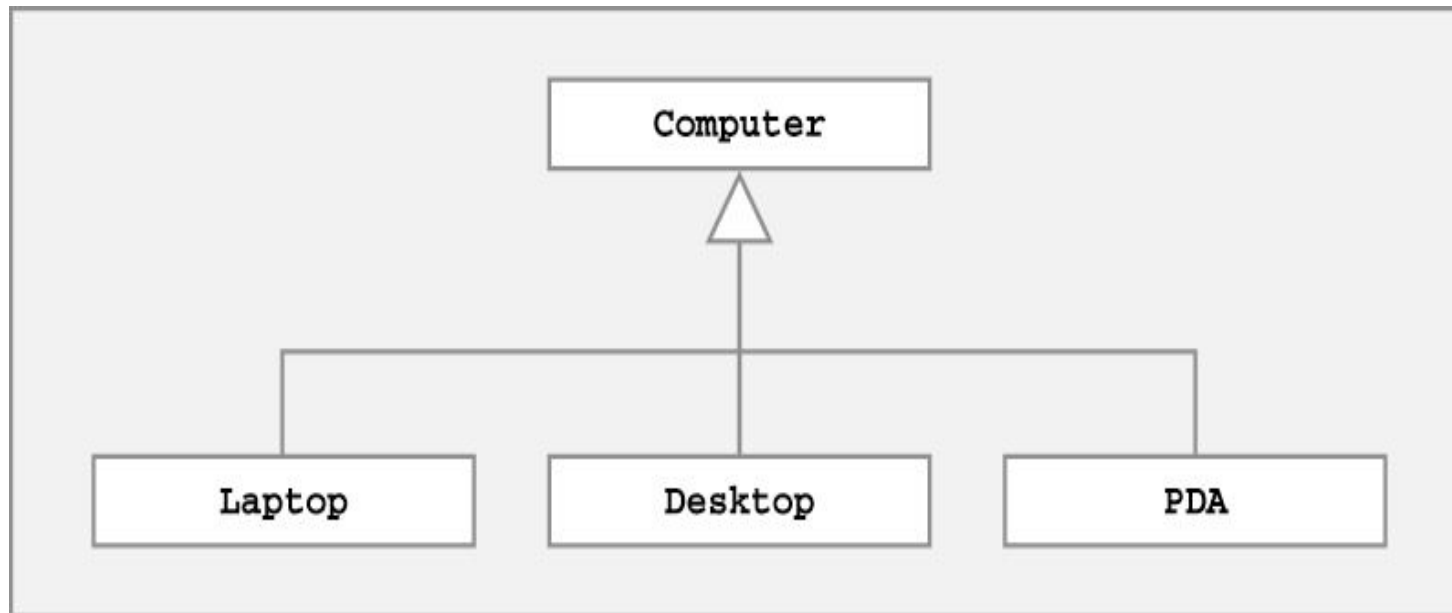
- Η **κληρονομικότητα** (**inheritance**) είναι μια από τις κύριες τεχνικές του αντικειμενοστρεφούς προγραμματισμού (OOP)
- Για να αξιοποιήσουμε την κληρονομικότητα:
- Δηλώνουμε και μεταγλωττίζουμε μια «**γενική**» **κλάση** και στη συνέχεια:
- επαναχρησιμοποιούμε την κλάση αυτή στη δήλωση άλλων, πιο **εξειδικευμένων κλάσεων**, οι οποίες:
 - **διατηρούν** («κληρονομούν») τις **μεθόδους** και τις **μεταβλητές** της αρχικής κλάσης
 - **τροποποιώντας** κατά περίπτωση μεθόδους της αρχικής κλάσης
 - Και/ή προσθέτοντας **επιπλέον μεταβλητές** και **μεθόδους**.

Κληρονομικότητα: Εισαγωγή κ Ορολογία

- Κληρονομικότητα (inheritance) είναι μια διαδικασία δημιουργίας νέας κλάσης (εξειδικευμένης) από άλλη, υφιστάμενη κλάση (γενική)
 - Η νέα κλάση ονομάζεται **κλάση-κληρονόμος** ή **υποκλάση** ή **κλάση-παιδί** ή **απόγονος** ή **παράγωγη κλάση** (*derived class* ή *subclass*)
 - Η αρχική κλάση ονομάζεται **βάση**, **κληροδότης**, ή **υπερκλάση**, **κλάση-γονέας** / **πρόγονος** (*base class* ή *superclass*)
- Η κλάση-κληρονόμος αυτόματα περιλαμβάνει **τις μεταβλητές στιγμιοτύπου και τις μεθόδους** της κλάσης βάσης, και μπορεί να επεκταθεί με **επιπλέον μεθόδους και πεδία δεδομένων**.
- Το πλεονέκτημα της κληρονομικότητας είναι ότι **επιτρέπει την επαναχρησιμοποίηση κώδικα** χωρίς να χρειάζεται η αντιγραφή του μέσα στους ορισμούς των παράγωγων κλάσεων.

Επαναχρησιμοποίηση κλάσεων

- **Κληρονομικότητα** (inheritance): επεκτείνουμε υπάρχουσες κλάσεις, είτε προσθέτοντάς τους καινούρια στοιχεία, είτε αλλάζοντας την λειτουργικότητα μεθόδων τους.





Μια εταιρεία X απασχολεί
ωρομίσθιους και εργαζόμενους
με μισθό. Ορισμένοι ωρομίσθιοι
εργάζονται σε καθεστώς
πλήρους απασχόλησης και
ορισμένοι σε καθεστώς μερικής.
Οι εργαζόμενοι με μισθό
διαχωρίζονται σε τεχνικό
προσωπικό και σε προσωπικό
διοίκησης. Στο τεχνικό
προσωπικό εντάσσονται
μηχανικοί, τεχνίτες και γραφείς.

Μοντελοποιείστε σε UML τους
εργαζόμενους της εταιρείας X

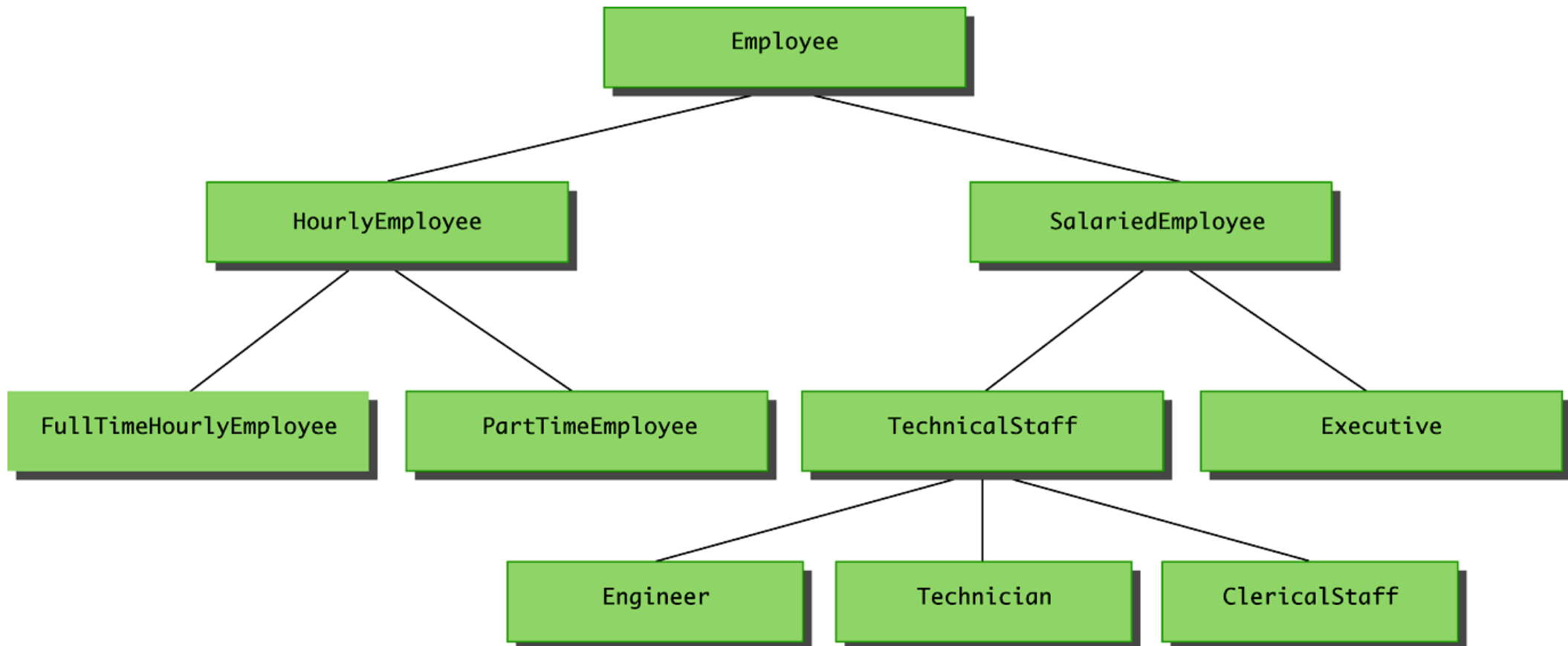
Ιεραρχία κλάσεων



- Κατά το σχεδιασμό ορισμένων κλάσεων, από τη λογική ομαδοποίηση τους, προκύπτει συχνά μια φυσική ιεραρχία (*hierarchy*).
- Π.χ.: σε πρόγραμμα τήρησης αρχείων για τους υπαλλήλους μιας επιχείρησης υπάρχουν **ωρομίσθιοι** (*hourly employees*) και **μισθωτοί** (*salaried employees*).
 - Οι **ωρομίσθιοι** μπορούν να χωριστούν σε **πλήρους και μερικής απασχόλησης**.
 - Οι **μισθωτοί** μπορούν να διαιρεθούν σε εκείνους που εργάζονται ως **τεχνικό** και ως **διοικητικό** προσωπικό.

Ιεραρχία κλάσεων

Display 7.1 A Class Hierarchy



is-a relationship

Ιεραρχία κλάσεων

- Όλοι οι εργαζόμενοι μοιράζονται **ορισμένα κοινά χαρακτηριστικά**:
 - Όλοι οι εργαζόμενοι έχουν όνομα (**name**) και **hire date** (ημερομηνία πρόσληψης)
 - Οι μέθοδοι καθορισμού και αλλαγής ονομάτων και ημερομηνιών πρόσληψης **θα είναι οι ίδιες για όλους τους εργαζομένους**.
- Κάποιες κατηγορίες εργαζόμενων έχουν **εξειδικευμένα χαρακτηριστικά**:
 - Οι ωρομίσθιοι αμείβονται με την ώρα, ενώ οι ημερομίσθιοι έχουν σταθερό μισθό.
 - Οι μέθοδοι υπολογισμού των μισθών για αυτές τις δύο διαφορετικές ομάδες πρέπει να είναι διαφορετικές.

Παράγωγες κλάσεις - Κληρονόμοι

- Στη Java ορίζουμε μια κλάση **Employee** που περιγράφει/περιλαμβάνει όλους τους υπαλλήλους
- Αυτή η κλάση μπορεί στη συνέχεια να χρησιμοποιηθεί για να ορίσουμε κλάσεις «κληρονόμους» όπως η **HourlyEmployee** για ωρομίσθιους και η **SalariedEmployee** για μισθωτούς εργαζόμενους
- Με τη σειρά της, η κλάση **HourlyEmployee** μπορεί να χρησιμοποιηθεί για να ορισθεί μια κλάση **PartTimeHourlyEmployee** ΚΟΚ

Παράγωγες κλάσεις - Κληρονόμοι

- Μια **κλάση κληρονόμος** ή **παράγωγος κλάση** (**derived class**) ορίζεται **επεκτείνοντας** μια υπάρχουσα κλάση και προσθέτοντας στη νέα κλάση, επιπλέον μεταβλητές και μεθόδους.
- Η υπάρχουσα κλάση πάνω στην οποία βασίζεται η παράγωγη κλάση ονομάζεται **κλάση βάσης** ή **κλάση κληροδότης** (**base class**).
- Η «επέκταση» υπάρχουσας κλάσης γίνεται με τη χρήση της φράσης **extends BaseClass** στον ορισμό της παράγωγης κλάσης:

```
public class HourlyEmployee extends Employee
```

Μέλη υποκλάσεων: πεδία δεδομένων

- Η παράγωγη κλάση **κληρονομεί** (**inherits**) τις **μεταβλητές στιγμιοτύπου** (πεδία δεδομένων/**instance variables**) της κλάσης βάσης την οποία επεκτείνει:
 - Η κλάση **Employee** ορίζει τις μεταβλητές **name** και **hireDate** ως πεδία δεδομένων της
 - Η κλάση **HourlyEmployee** διαθέτει τις μεταβλητές αυτές, **χωρίς να τις έχει συμπεριλάβει στη δήλωση της**.
 - Η κλάση **HourlyEmployee** διαθέτει επιπλέον τις μεταβλητές **wageRate** και **hours** που περιλαμβάνονται στη δήλωση της, αλλά δεν περιλαμβάνονται στην **Employee**.

Μέλη υποκλάσεων: μέθοδοι

- Η παράγωγη κλάση **HourlyEmployee** κληρονομεί και τις **μεθόδους** (**methods**) της κλάσης **Employee**
 - Η κλάση **HourlyEmployee** κληρονομεί τις **getName**, **getHireDate**, **setName**, και **setHireDate** από την **Employee**
 - Οποιοδήποτε αντικείμενο της κλάσης **HourlyEmployee** μπορεί να καλέσει μία από αυτές τις κληρονομημένες μεθόδους, όπως και κάθε άλλη (δημόσια) μέθοδο δηλωμένη στο εσωτερικό της **Employee**.

Παράγωγες κλάσεις: Σύνοψη

- Η δήλωση μιας **παράγωγης κλάσης** (κληρονόμος/υποκλάση/subclass), ξεκινάει με μια υφιστάμενη κλάση: την κλάση βάσης (base class/superclass/υπερκλάση).
- Η παράγωγη κλάση κληρονομεί:
 - Όλες τις **δημόσιες μεθόδους**
 - Όλες τις **μεταβλητές** στιγμιοτύπου
 - Όλες τις **στατικές μεταβλητές** της κλάσης βάσης
- Στην παράγωγη κλάση μπορεί να τροποποιηθούν **κληρονομημένες μέθοδοι**.
- Η παράγωγη κλάση μπορεί να εμπλουτισθεί με πρόσθεση στη δήλωση της περισσότερων μεταβλητών στιγμιοτύπου, στατικών μεταβλητών ή/και μεθόδων.

Κληρονομούμενα μέλη

- Μια παράγωγος κλάση έχει αυτόματα όλες τις μεταβλητές στιγμιότυπου, όλες τις στατικές μεταβλητές και όλες τις δημόσιες μεθόδους της κλάσης βάσης
 - Τα μέλη της κλάσης βάσης λέμε ότι έχουν κληρονομηθεί (inherited)
- Οι ορισμοί για τις κληρονομούμενες μεταβλητές και τις μεθόδους **δεν εμφανίζονται** στην παράγωγο κλάση:
 - Ο κώδικας του κληροδότη **επαναχρησιμοποιείται** χωρίς να χρειάζεται ρητή αντιγραφή του, **εκτός εάν** ο δημιουργός της παράγωγης κλάσης **επαναπροσδιορίσει** μία ή περισσότερες από τις μεθόδους βασικής κλάσης

Κατασκευαστές και ένθετες κλάσεις

- Οι κατασκευαστές δεν θεωρούνται μέλη της κλάσης και συνεπώς δεν κληρονομούνται από υποκλάσεις, αλλά ο κατασκευαστής της υπερκλάσης μπορεί να κληθεί από την υποκλάση.
- Μια υποκλάση (subclass) κληρονομεί όλα τα μέλη της υπερκλάσης της (πεδία, μεθόδους όπως και ένθετες/φωλιασμένες κλάσεις - (nested classes)).

Σύνοψη: ορολογία κληρονομικότητας

- **Base class**: βάση, κληροδότης, ή superclass, υπερκλάση, κλάση-γονέας, μητρική κλάση.
- **Derived class**: Κλάση-κληρονόμος ή subclass, υποκλάση, κλάση-παιδί, απόγονος, παράγωγη κλάση.
- Αυτές οι σχέσεις επεκτείνονται συχνά έτσι ώστε μια κλάση που είναι γονέας ενός γονέα. . . μιας άλλης κλάσης ονομάζεται **πρόγονη κλάση** (*ancestor class*).
- Αν η κλάση **A** είναι **πρόγονος** της κλάσης **B**, τότε η κλάση **B** μπορεί να ονομαστεί **απόγονος** (*descendent*) της κλάσης **A**.