

ΑΣΚΗΣΕΙΣ ΚΛΗΡΟΝΟΜΙΚΟΤΗΤΑ (*inheritance*)

ΑΣΚΗΣΗ-1^η

Να γραφεί το πρόγραμμα Java που υλοποιεί την **κληρονομικότητα**, για παράδειγμα η υποκλάση κληρονομεί **μία λειτουργικότητα** της υπερκλάσης. Π.χ. ορίστε την υπερκλάση **Superclass** με χαρακτηριστικά (1) το γένος – gender (String) και (2) την ηλικία – age (int), και μία μέθοδο την **getGenderAndAge()** που εμφανίζει τα ανωτέρω χαρακτηριστικά. Η υποκλάση **Subclass** κληρονομεί από την Superclass την ανωτέρω λειτουργικότητα. Χρησιμοποιείστε επίσης την παραμετρική super() για αρχικοποίηση των χαρακτηριστικών (κλήση του δομητή) της υπερκλάσης.

ΑΣΚΗΣΗ-2^η

Να γραφεί το πρόγραμμα Java που υλοποιεί την **κληρονομικότητα**, για παράδειγμα η υποκλάση κληρονομεί **ένα χαρακτηριστικό** της υπερκλάσης. Π.χ. ορίστε την υπερκλάση **Superclass1** με χαρακτηριστικό το όνομα – name(String) και μία μέθοδο την **display()** που εμφανίζει το όνομα. Η υποκλάση **Subclass1** κληρονομεί από την Superclass1 το ανωτέρω χαρακτηριστικό. Χρησιμοποιείστε επίσης την **super.<πεδίο>** για πρόσβαση στο πεδίο της υπερκλάσης, αλλά και αντικείμενο της υποκλάσης για κλήση της μεθόδου display().

ΑΣΚΗΣΗ-3^η

Να γραφεί το πρόγραμμα Java που υλοποιεί την **κληρονομικότητα**, για παράδειγμα η υποκλάση κληρονομεί **χαρακτηριστικά και μεθόδους** της υπερκλάσης. Π.χ. ορίστε την υπερκλάση **Employee** με private χαρακτηριστικά (1) το όνομα – name(String), (2) την ηλικία – int, καθώς setter() και αντίστοιχες getter(). Η υποκλάση **Manager** κληρονομεί από την Employee και στην main() ορίστε ένα αντικείμενο της Manager και με τις getter() εμφανίστε τα ανωτέρω χαρακτηριστικά.

ΑΣΚΗΣΗ-4^η

Να γραφεί το πρόγραμμα Java που **ελέγχει την κληρονομικότητα** σε μια κληρονομικότητα (Animal, Mammal, Reptile, Cat, Dog) με χρήση της **instanceof**. Η σύνταξη:
if (<objectReference> instanceof <type>)

ΑΣΚΗΣΗ-5^η

Να γραφεί ένα πρόγραμμα Java που δείχνει την υπέρβαση μεθόδου. Το πρόγραμμα ορίζει την υπερκλάση **Human** με δύο μεθόδους (1) **void eat()** και (2) **void drink()** με απλά αντίστοιχα μηνύματα και την υποκλάση **Boy** που κληρονομεί την Human αλλά υλοποιεί τις ανωτέρω μεθόδους με τα αντίστοιχα μηνύματα. Με αντικείμενο της υποκλάσης δείξτε την υπέρβαση των μεθόδων.

ΑΣΚΗΣΗ-6^η (Κληρονομικότητα και υπέρβαση μεθόδων)

Η μισθοδοσία των υπαλλήλων μιας εταιρείας περιλαμβάνει 2 κατηγορίες υπαλλήλων (1) τους Διοικητικούς και (2) τους Τεχνικούς. Για τις 2 κατηγορίες υπάρχουν δύο τύποι μισθοδοσίας (α) **0**=Μισθωτός και (β) **1**=Ωρομίσθιος. Ο βασικός μισθός για τους Διοικητικούς είναι 1200 Ευρώ και για τους Τεχνικούς 800 Ευρώ, ενώ και οι 2 κατηγορίες μπορούν να λάβουν και ένα bonus (X)-Ευρώ. Ο μισθός υπολογίζεται με την σχέση: **Μισθός = Βασικός + Bonus**, για τους μισθωτούς και για τους ωρομίσθιους με τη σχέση: **Μισθός = Ώρες Εργασίας * Τιμή Ώρας** (όπου για τον Διοικητικό είναι 12 Ευρώ την ώρα, και για τον Τεχνικό 10 Ευρώ/ώρα).

Κατασκευαστικά:

Η υπερκλάση: **Employee**, θα ορίζει 4 χαρακτηριστικά που είναι κοινά και για τις δύο κατηγορίες (1) Name, (2) Bonus, (3) Hours (για τους ωρομίσθιους) και (4) Τύπος Υπάλληλου (0=Διοικητικός, 1=Τεχνικός) και θα υλοποιεί δύο σημαντικές μεθόδους (α) **getPayType()** που επιστρέφει τα μηνύματα 'Μισθός' ή 'Με την Ώρα' ανάλογα του τύπου μισθοδοσίας, (β) **getEmpType()** που επιστρέφει το μήνυμα "Employee", και (γ) **calcSalary()**, που υπολογίζει τον μισθό του Employee. Οι υποκλάσεις: **Administrator** και **Technical** θα κληρονομούν τα 4-χαρακτηριστικά και την μέθοδο getPayType() από την υπερκλάση, ενώ θα υπερβαίνουν (*override*) τις μεθόδους getEmpType() και calcSalary(). Χρησιμοποιήστε μια βοηθητική κλάση την **Company** στην οποία αρχικοποιείται ένας πίνακας 4-αντικειμένων όλων των τύπων και επιπλέον 2 μεθόδους (α) **setEmployee()**, που εισάγει τα αντικείμενα στον πίνακα και (β) **printAll()** που εμφανίζει τα στοιχεία του κάθε υπάλληλου και τον μισθό του.

ΑΣΚΗΣΗ-7^η (Σύνθεση)

Η κλάση **Job** περιέχει τα πεδία: (1) perigrifi (String), (2) misthos(int), (3) kodikos(int) και τις αντίστοιχες setter() και getter(). Η κλάση **Person** δημιουργεί και αρχικοποιεί στον δομητή της ένα σύνθετο αντικείμενο της Job και υλοποιεί τις μεθόδους της (Job). Στην main() να υλοποιηθεί η σύνθεση με αντικείμενο της Person (εμφάνιση όλων των στοιχείων).

ΑΣΚΗΣΗ-8^η (Σύνθεση)

Η κλάση **Employee** θα λάβει στοιχεία από δύο άλλες κλάσεις τις: (1) **Name**: που ορίζει το όνομα του υπάλληλου, και (2) **Address**: που ορίζει την διεύθυνση του υπάλληλου.

Η κλάση Employee: που ορίζει τα στοιχεία του υπάλληλου, χρησιμοποιώντας όμως σαν μεταβλητές **αντικείμενα** του των τύπων Name και Address.

Η **TestEmployee**: η οποία ορίζει αντικείμενα του τύπου Name, Address και Employee και που εμφανίζει όλα τα στοιχεία του Υπάλληλου.

Γράψτε το πρόγραμμα Java που υλοποιεί την σύνθεση χρησιμοποιώντας τις τέσσερις ανωτέρω κλάσεις.

ΑΣΚΗΣΗ-9^η (Σύνθεση)

Η κλάση **Person** έχει τα πεδία: (1) salary (double), (2) name(String) και (3) μια **αναφορά σε αντικείμενο** της κλάσης **Birthday** που έχει και αυτή τρία πεδία: day(int), month(int), year(int). Στην main() να δημιουργήσετε αντικείμενο της Person και να υλοποιήσετε την σύνθεση εμφανίζοντας όλα τα στοιχεία (συμπεριλαμβανομένης της ημερομηνίας των γενεθλίων).

ΑΣΚΗΣΗ-10^η (Σύνθεση και κληρονομικότητα)

Η κλάση **Car** έχει τα πεδία (1) color(String), (2) maxSpeed(int), την μέθοδο carInfo() που εμφανίζει τα ανωτέρω πεδία, και τις αντίστοιχες setter() για τα πεδία. Η κλάση **Ferrari** που κληρονομεί την Car έχει μια μέθοδο την FerrariDemo() που ορίζει ένα αντικείμενο της κλάσης **Engine** και υλοποιεί την μέθοδο της- την start(). Η κλάση Engine περιέχει εκτός από την μέθοδο start() και την μέθοδο stop() (απλά μηνύματα). Στην main() δημιουργήστε αντικείμενο της Ferrari που θα **κληρονομεί** και εμφανίζει από την Car τα πεδία και την carInfo(), αλλά θα υλοποιεί και **σύνθεση** με χρήση της μεθόδου start() της κλάσης Engine.

ΑΛΥΤΕΣ ΑΣΚΗΣΕΙΣ

ΑΣΚΗΣΗ-1^η (κληρονομικότητα)

Δημιουργήστε τις υποκλάσεις **StatheroTelefono** και **KinitoTelefono** και την υπερκλάση **Telefono**. Οι δύο υποκλάσεις θα κληρονομούν από την υπερκλάση χαρακτηριστικά και μεθόδους. Όλες οι κλάσεις που θα δημιουργήσετε να έχουν ένα πλήρη δομητή, τα κατάλληλα πεδία, τις μεθόδους setters() και getters(), toString() και τι μεθόδους: (1) **TyposThlefonou**, (2) **klisiKinitou** και (3) **klisiStatherou** που πραγματοποιούν, ανάλογα με τον τύπο του τηλεφώνου, κλήσεις στο κινητό και σταθερό τηλέφωνο. Η χρέωση είναι 0.10 και 0.20 το λεπτό προς το σταθερό και κινητό αντίστοιχα. Το πρόγραμμα θα υπολογίζει τα επιμέρους κόστη για τις κλήσεις προς κινητό και σταθερό αλλά και το συνολικό κόστος όλων των **N – κλήσεων**.

Στην main δημιουργήστε ένα πίνακα N - αντικειμένων με κινητά και σταθερά και με τις αντίστοιχες

κλήσεις και υλοποιήστε τις μεθόδους και τους υπολογισμούς που προαναφέρθηκαν. Τα αποτελέσματα θα εμφανίζονται με τα κατάλληλα μηνύματα.

ΑΣΚΗΣΗ-2^η (Σύνθεση και κληρονομικότητα)

Coffee Machine (Κληρονομικότητα και Σύνθεση)

Να γραφεί το πρόγραμμα υλοποίησης του Coffee Machine. Ορίστε την υπερκλάση **Coffee** με πεδία:

- (1) coffeeName (π.χ. "frappe", "capuccino, κ.λ.π...)
- (2) typosCafe (1="kaonikos", 2="decaf")
- (3) sweetness (1="glykos", 2="metrios", 3="sketos")
- (4) dosi (1="monos", 2="diplos")
- (5) price (π.χ. 1 "€")

Επιπλέον θα έχει τις μεθόδους **getCoffeeName()**, **getTypos_cafe()**, **getSweetness()**, **getDosi()** και **getPrice()** οι οποίες θα επιστρέφουν τα ανάλογα μηνύματα.

Στη συνέχεια να δημιουργήσετε τις υποκλάσεις: (1) **Espresso**, (2) **Capuccino**, (3) **Frappe**, (4) **Ellinikos**, (5) **Filtrou** οι οποίες κληρονομούν από την υπερκλάση όλες τις λειτουργίες της. Οι υποκλάσεις διαθέτουν τα παρακάτω επιπλέον πεδία και λειτουργίες:

Η κλάση Espresso:

- (1) gefsi (π.χ. "fountouki", "karamela", κ.λ.π..)
- (2) iced (1="kryos", 2="zestos")

και τις μεθόδους **getIced()** και **getGefsi()** οι οποίες θα επιστρέφουν τα αντίστοιχα μηνύματα.

Η κλάση Capuccino:

- (1) gefsi ("fountouki", "karamela", κ.λ.π..)
- (2) iced (1="kryos", 2="zestos")
- (3) kanella (1="nai", 2="oxi")

και τις μεθόδους **getIced()**, **getGefsi()** και **getKanella()** οι οποίες θα επιστρέφουν τα αντίστοιχα μηνύματα.

Η κλάση Frappe:

- (1) gefsi ("fountouki", "karamela", κ.λ.π..)
- (2) iced (1="kryos", 2="zestos")

και τις μεθόδους **getIced()**, **getGefsi()** οι οποίες θα επιστρέφουν τα αντίστοιχα μηνύματα.

Η κλάση **Filtrou**:

(1) gefsi ("fountouki", "karamela", κ.λ.π..) και τη μέθοδο **getGefsi()** η οποία θα επιστρέφει το ανάλογο μήνυμα.

Η κλάση **Ellinikos** δεν έχει κανένα επιπλέον πεδίο.

Τέλος να δημιουργήσετε τις κλάσεις **Sugar** και **Cream** (για την σύνθεση - composition).

Η **Sugar** έχει ως πεδίο το typosSug το οποίο δέχεται τις τιμές: "1" αν η ζάχαρη είναι "white" και "2" αν η ζάχαρη είναι "brown" και τη μέθοδο **gettyposSug()** η οποία επιστρέφει το ανάλογο μήνυμα.

Η **Cream** έχει ως πεδίο το typosCre το οποίο δέχεται: "0" αν δεν έχει κρέμα, "1" αν η κρέμα είναι "afrogala" , "2" αν η κρέμα είναι "fitiki" και "3" αν έχουμε γάλα και τη μέθοδο **getTyposCre()** η οποία επιστρέφει το ανάλογο μήνυμα.

Στη main() θα δημιουργείτε αντικείμενα όλων των υποκλάσεων και υλοποιείτε όλους τους τύπους των καφέδων.

Παράδειγμα Lay out – εμφάνιση αντικειμένου (καφέ):

Coffee name: Frappe

Typos cafe: kanonikos

Sweetness: glykos (1)

Dosi: diplos (2)

Gefsi: fountouki

Iced: kryos (1)

TyposSug: white -sugar (1)

TyposCre: afrogalo (1)

Price: 2€