

ΑΣΚΗΣΕΙΣ  
Πολυμορφισμός –  
Αφηρημένες Κλάσεις  
(*Polymorphism – Abstract Classes*)

---

**ΑΣΚΗΣΗ-1<sup>η</sup>**

Να γραφεί το πρόγραμμα Java που υλοποιεί την **απόκρυψη πεδίων(hiding fields)**. Δηλώστε ένα String στην υπερκλάση και με το ίδιο όνομα άλλο String στην υποκλάση. Στην main() δημιουργείτε αντικείμενο της υποκλάσης και υλοποιείτε την απόκρυψη. Επίσης κάντε χρήση της αναφοράς υπερκλάσης σε αντικείμενο της υποκλάσης για να υλοποιήσετε πρόσβαση στο κρυμμένο πεδίο της υπερκλάσης. Τέλος δείξτε πως θα έχουμε πρόσβαση στο κρυμμένο πεδίο της υπερκλάσης με την χρήση casting.

**ΑΣΚΗΣΗ-2<sup>η</sup>**

Να γραφεί το πρόγραμμα που υλοποιεί **Στατική και Δυναμική δέσμευση** στη Java. Γράψτε την υπερκλάση **Employee** που υλοποιεί την μέθοδο Salary (μήνυμα ο «Υπάλληλος πληρώνεται») και την υποκλάση **Misthotos** που κληρονομεί την Employee και υλοποιεί επίσης την μέθοδο Salary (μήνυμα ο «Μισθωτός πληρώνεται»). Στην main() με τον ορισμό των κατάλληλων αντικειμένων και αναφορών σε αντικείμενα υλοποιείτε και τις δύο δεσμεύσεις.

**ΑΣΚΗΣΗ-3<sup>η</sup>**

Να γραφεί το πρόγραμμα Java που υλοποιεί τον **Πολυμορφισμό**. Γράψτε την υπερκλάση **Animal** που περιέχει την μέθοδο sound (μήνυμα «Animal sound») και τις υποκλάσεις **Cat** και **Dog** που κληρονομούν την Animal και υλοποιούν την μέθοδο sound (με αντίστοιχα μηνύματα). Στην main() δημιουργείτε αντικείμενα και αναφορές σε αντικείμενα και δείξτε τον πολυμορφισμό.

**ΑΣΚΗΣΗ-4<sup>η</sup>**

Να γραφεί το πρόγραμμα Java που υλοποιεί **Upcasting** και **Downcasting**. Η κλάση **Camera** έχει την μέθοδο start (μήνυμα η «κάμερα ξεκινά») και η κλάση **Nikon** που κληρονομεί την Camera, έχει και αυτή την μέθοδο start (μήνυμα η «Nikon ξεκινά») και την μέθοδο snap (μήνυμα «Βγαζει φωτό»). Στην main() ορίστε αντικείμενα και αναφορές σε αντικείμενα που θα χρησιμοποιήσετε για να δείξετε το Upcasting και Downcasting. Μη ξεχνάτε ότι στο downcasting απαιτείται casting.

### ΑΣΚΗΣΗ-5<sup>η</sup>

Να γραφεί ένα πρόγραμμα Java που δείχνει την **Υπέρβαση και RunTime Polymorphism**. Η υπερκλάση **Human** έχει δύο μεθόδους (1) void eat() (μήνυμα ο άνθρωπος τρώει πίτσα) και (2) void drink() (μήνυμα ο άνθρωπος πίνει μπίρα) και η υποκλάση **Boy** που κληρονομεί την Human υλοποιεί τις δύο ανωτέρω μεθόδους με τον δικό της τρόπο (αντίστοιχα μηνύματα). Στην main() δημιουργήστε τα κατάλληλα αντικείμενα και δείξτε την υπέρβαση ή RunTime Πολυμορφισμό.

### ΑΣΚΗΣΗ-6<sup>η</sup>

Να γραφεί ένα πρόγραμμα Java που δείχνει την **Υπέρβαση και Πολυμορφισμό Υποκατάστασης**. Η κλάση **Circle** με χαρακτηριστικό την ακτίνα(radius) υλοποιεί την μέθοδο getArea() που υπολογίζει την περιφέρεια του κύκλου ( $radius * radius * Math.PI$ ) και την getRadius(). Η υποκλάση **Cylinder** που κληρονομεί την Circle έχει επιπλέον χαρακτηριστικό το ύψος (height) και υλοποιεί την μέθοδο getVolume() ( $getArea() * height$ ) και την getHeight(). Στην main() δημιουργήστε τα κατάλληλα αντικείμενα και δείξτε την υπέρβαση και τον πολυμορφισμό υποκατάστασης.

### ΑΣΚΗΣΗ-7<sup>η</sup> (κληρονομικότητα με αφηρημένη υπερκλάση)

Η **αφηρημένη** υπερκλάση **Shape** έχει χαρακτηριστικά (1) color, String (2) filled, boolean, και δύο αφηρημένες μεθόδους τις getArea() και getPerimeter(). Οι υποκλάσεις **Circle** και **Rectangle** κληρονομούν την Shape και περιέχουν:

- (1) Η Circle, το χαρακτηριστικό radius (double) και υλοποιεί τις getArea()( $radius * radius * Math.PI$ ) και getPerimeter() ( $2 * radius * Math.PI$ ) και αντίστοιχες setter() και getter().
- (2) Η Rectangle, τα χαρακτηριστικά width(double) και height(double) και υλοποιεί τις getArea()( $radius * radius * Math.PI$ ) και getPerimeter() ( $2 * radius * Math.PI$ ) και αντίστοιχες setter() και getter().

Στην main() ορίστε αντικείμενα του τύπου Circle και Rectangle και υλοποιήστε τις ανωτέρω μεθόδους.

### ΑΣΚΗΣΗ-8<sup>η</sup> (κληρονομικότητα με αφηρημένη υπερκλάση)

Να γραφεί πρόγραμμα παραλλαγή της λυμένης άσκησης στην κληρονομικότητα με υπερκλάση την **Employee**. Η κλάση τώρα θα είναι αφηρημένη και θα ορίζει εκτός από τα προηγούμενα χαρακτηριστικά και τις αφηρημένες μεθόδους **abstract String getEmpType()** και **abstract void calcSalary()**. Οι υποκλάσεις **Administrator** και **Technical** που κληρονομούν την Employee υλοποιούν τις δύο ανωτέρω μεθόδους. Στην main() ορίστε ένα πίνακα αντικειμένων και εκτελέστε τις ανωτέρω λειτουργίες.

### ΑΣΚΗΣΗ-9<sup>η</sup> (κληρονομικότητα με αφηρημένη υπερκλάση)

Η αφηρημένη υπερκλάση **MousikaOrgana** περιέχει μια abstract μέθοδο την `play()` και μια μέθοδο που δεν είναι αφηρημένη την `BrandName()`, που θα υλοποιηθούν στις υποκλάσεις που την κληρονομούν. Οι υποκλάσεις είναι:

- (1) **Kithara** (κληρονομεί την `MousikaOrgana`)
- (2) **HlektrikiKithara** (κληρονομεί την `Kithara`)
- (3) **KlassikiKithara** (κληρονομεί την `Kithara`)
- (4) **Krousta** (κληρονομεί την `MousikaOrgana`)

Στην `main()` ορίστε ένα πίνακα αντικειμένων των υποκλάσεων και εκτελέστε τις ανωτέρω λειτουργίες.

## **ΑΛΥΤΕΣ ΑΣΚΗΣΕΙΣ**

### **ΑΣΚΗΣΗ-1<sup>η</sup>** (Πολυμορφισμός)

Να γίνει το πρόγραμμα που χειρίζεται την πώληση και ενοικίαση κατοικιών. Το πρόγραμμα θα ορίζει την κλάση **Katoikia**, με private μέλη: (1) **Κωδικός**, `int` ( $1=πώληση$ ,  $2=ενοικίαση$ ), (2) **Πλήθος Δωματίων**, `int`, (3) **Τετραγωνικά μέτρα**, `double`, (4) **Διεύθυνση**, `String`, και τις μεθόδους: (1) **TyposSynalagis()**, που θα επιστρέφει με μήνυμα, τον τύπο της συναλλαγής, δηλ. "Polisi", "Epoikio", ανάλογα με τον κωδικό, (2) **Poso()**, που θα υπολογίζει το τελικό ποσό συναλλαγής, λαμβάνοντας υπόψιν τα παρακάτω: Αν είναι για ενοίκιο, τότε η τιμή προσαυξάνεται επί δύο (προκαταβολή ενός ενοικίου) συν (+) μισό ενοίκιο (μισή **τιμή ενοικίου**) σαν προμήθεια. Αν είναι πώληση τότε η **τιμή πώλησης** επιβαρύνεται με 1200 Ευρώ (προμήθεια). Οι κλάσεις "**Diamerisma**", "**ExohikiKatoikia**" και "**Mesonette**" κληρονομούν/εξειδικεύουν την κλάση "House" και διαθέτουν επιπλέον τα παρακάτω χαρακτηριστικά:

**Diamerisma:** `TimiD` (*double*), **ExohikiKatoikia:** `TimiEx` (*double*), **Mesonette:** `TimiM` (*double*) και υλοποιούν με υπέρβαση τις μεθόδους: **TyposSynalagis()** και **Poso()**, Στο κυρίως πρόγραμμα δημιουργήστε ένα πίνακα **N - κατοικιών**. Το N εισάγεται από το Πληκτρολόγιο. Σε κάθε θέση του πίνακα μπορεί να ανατίθεται αντικείμενο τύπου "Diamerisma" (60%-στον πίνακα), "ExohikiKatoikia" (20%-στον πίνακα), ή τύπου "Mesonette" (20%-στον πίνακα). Το πρόγραμμα εμφανίζει τα χαρακτηριστικά κάθε κατοικίας, καθώς και τα αποτελέσματα των μεθόδων και `TyposSynalagis()` και `Poso()`. Ορίστε δομητές και μεθόδους `getter()`, όπου χρειάζεστε. Όλα τα χαρακτηριστικά να είναι `private`. Υπερβείτε την μέθοδο **toString()** της `Object`.

### **ΑΣΚΗΣΗ-2<sup>η</sup>** (Αφηρημένες κλάσεις)

Να γραφεί πρόγραμμα που δημιουργεί την **αφηρημένη κλάση Shape**, η οποία θα περιέχει τα private μέλη - σημεία **X** και **Y** (τύπου `double`), ένα πλήρη δομητή, αντίστοιχες μεθόδους **getter()**, την αφηρημένη μέθοδο **area()** που θα υλοποιηθεί στις υποκλάσεις και την αντίστοιχη μέθοδο **toString()**. Επίσης το πρόγραμμα θα δημιουργεί δύο υποκλάσεις που θα κληρονομούν (`extends`) την `Shape`: τις **Rectangle** και **Circle**.

### H Rectangle:

- θα περιέχει τα private μέλη `height` και `width`, ένα πλήρη δομητή, την υλοποίηση της `area()` ( $height * width$ ) και την αντίστοιχη μέθοδο `toString()`.

### H Circle:

- θα περιέχει το private μέλος ακτίνα (`radius`), ένα πλήρη δομητή, την υλοποίηση της `area()` ( $Math.PI * radius * radius$ ) και την αντίστοιχη μέθοδο `toString()`.

Στην κλάση **AbstractShape** (με την `main()`), θα δημιουργηθούν δύο αντικείμενα των τύπων `Rectangle` και `Circle` σε ένα πίνακα αντικειμένων και θα εκτελείται ο υπολογισμός και η εμφάνιση των αποτελεσμάτων (`area()`) με τα κατάλληλα μηνύματα

### **ΑΣΚΗΣΗ-3<sup>η</sup>** (Αφηρημένες κλάσεις)

Ορίστε την αφηρημένη κλάση **"GeometricObject"** η οποία έχει τα χαρακτηριστικά: (1) **color**, String, (2) **filled**, boolean, την μέθοδο **isFilled()** που επιστρέφει true ή false ανάλογα, και τις αφηρημένες μεθόδους **getArea()** και **getPerimeter()**.

Οι κλάσεις **"Rectangle"**, **"Circle"** και **"Cylinder"** κληρονομούν/εξειδικεύουν την κλάση **"GeometricObject"** και διαθέτουν επιπλέον τα παρακάτω χαρακτηριστικά και μεθόδους:

**Rectangle :** (1) **width**, int, (2) **height**, int, και (α) **getArea()** ( $width * height$ ) και (β) **getPerimeter()** ( $2 * (width * height)$ ).

**Circle :** (1) **radius**, double και (α) **getArea()** ( $radius * radius * Math.PI$ ;) και (β) **getPerimeter()** ( $2 * (radius * Math.PI)$ ).

**Cylinder :** (1) **height**, int, (2) **radius**, double και (α) **getArea()** ( $2 * getArea() * getPerimeter() * height$ ) και (β) **Volume()** ( $getArea() * height$ ).

Στο κυρίως πρόγραμμα δημιουργήστε ένα πίνακα N σχημάτων. Το N εισάγεται από το πληκτρολόγιο. Σε κάθε θέση του πίνακα μπορεί να ανατίθεται αντικείμενο τύπου **"Rectangle"** (40%-στον πίνακα), **"Circle"** (30%-στον πίνακα), ή τύπου **"Cylinder"** (30%-στον πίνακα).

Το πρόγραμμα εμφανίζει τα χαρακτηριστικά κάθε σχήματος, καθώς και τα αποτελέσματα των μεθόδων `isFilled()`, `getArea()` και `getPerimeter()`.