ΑΣΚΗΣΕΙΣ

Διεπαφές / Διασυνδέσεις *(Interfaces)*

ΑΣΚΗΣΗ-1η (Ορισμός σταθερών σε Διεπαφή)

Να γραφεί το πρόγραμμα Java που υλοποιεί την 'τεχνική' του να **ορίζουμε σταθερές σε μια διεπαφή** (π.χ. syntelestis_FPA, mathimatiko_pi, κλπ.) τις οποίες χρησιμοποιούμε απ' ευθείας με το όνομά τους σε κλάση που **υλοποιεί** την διεπαφή.

ΑΣΚΗΣΗ-2η (Υλοποίηση μεθόδων της διεπαφής (κλασική χρήση), σε κλάση που υλοποιεί την διεπαφή)

Να γραφεί το πρόγραμμα Java που **υλοποιεί μεθόδους της διεπαφής σε κλάση** (που την υλοποιεί). Οι μέθοδοι method1() και method2() θα υλοποιούνται στην κλάση εμφανίζοντας απλά μηνύματα.

ΑΣΚΗΣΗ-3η (Κληρονομικότητα διεπαφής από άλλη διεπαφή)

Να γραφεί το πρόγραμμα Java **που υλοποιεί την κληρονομικότητα διεπαφής** *(από άλλη διεπαφή)* όπως μια κλάση κληρονομεί μια άλλη κλάση. Χρησιμοποιείστε μεθόδους που εμφανίζουν απλά μηνύματα. Π.χ. η διεπαφή ***Diepafi1*** έχει την μέθοδο method1() και η διεπαφή ***Diepafi2*** που κληρονομεί την *Diepafi1* έχει την μέθοδο method2() (απλά μηνύματα). Η κλάση **Demo** που υλοποιεί την *Diepafi2* υλοποιεί και τις δύο μεθόδους. Στην main() δημιουργείστε αντικείμενο και υλοποιείστε την κληρονομικότητα (*Diepafi2* από την *Diepafi1*) εκτελώντας και τις 2 μεθόδους.

ΑΣΚΗΣΗ-4η (Υλοποίηση μεθόδων διεπαφής)

Να γραφεί το πρόγραμμα Java **που δείχνει την υλοποίηση μεθόδων διεπαφής.** Η διεπαφή ***Shape*** ορίζει την μέθοδο draw() που θα υλοποιηθεί στις δύο κλάσεις **Circle** και **Rectangle** που υλοποιούν την διεπαφή (κατάλληλο μήνυμα). Στη main() δημιουργείστε αντικείμενα των δύο κλάσεων και δείξτε την υλοποίηση της draw() για τις δύο κλάσεις.

Παραλλαγή άσκησης-4: Χρησιμοποιείστε μια επιπλέον κλάση κατασκευής αντικειμένων, όπου σε μια μέθοδο π.χ. την getShape() η οποία δέχεται σαν παράμετρο ένα String (π.χ. "Circle" ή "Rectangle") δημιουργεί και επιστρέφει τα αντίστοιχα αντικείμενα. Ο τρόπος αυτός υλοποιεί το πρότυπο *(pattern)* ***Factory*** (δες λύσεις ασκήσεων).

ΑΣΚΗΣΗ-5η (Υλοποίηση μεθόδων διεπαφής)

Να γραφεί το πρόγραμμα Java **που δείχνει την υλοποίηση μεθόδων διεπαφής.** Η διεπαφή ***Emvadon*** ορίζει την μέθοδο computeEmvadon()(με δύο παραμέτρους x και y) που θα υλοποιηθεί στις δύο κλάσεις **Rectangle** και **Triangle** που υλοποιούν την διεπαφή. Το εμβαδόν του Rectangle υπολογίζεται ως (x*y) και του Triangle (x*y/2). Στη main() δημιουργείστε αντικείμενα των δύο κλάσεων και δείξτε την υλοποίηση της computeEmvadon() για τις δύο κλάσεις.

ΑΛΥΤΗ ΑΣΚΗΣΗ

ΑΣΚΗΣΗ-1η ("Έμμεση" πολλαπλή κληρονομικότητα)

Να γραφεί πρόγραμμα που χειρίζεται τις δοσοληψίες ενός λογαριασμού. Το πρόγραμμα ορίζει την κλάση **Account** με πεδία (1) ***eponymia***, *String,* (2) ***balance***, *double* και (3) ***poso_kinisis,*** *double* και επιπλέον πλήρη δομητή, getters() και την toString(). Για την εκτέλεση των δοσοληψιών/κινήσεων το πρόγραμμα ορίζει τις διεπαφές:

(1) ***iDebit*** : με την μέθοδο **void debit(String name, double balance, double poso);**

(2) ***iCredit*** : με την μέθοδο **void credit(String name, double balance, double poso);**

(1) ***iMetafora*** : με την μέθοδο **void metafora(String name, double balance, double poso);**

και τις κλάσεις:

(1) **Debit,** που κληρονομεί *(extends)* την Account και υλοποιεί *(implements)* την *iDebit*. Η κλάση περιέχει ένα πλήρη δομητή, και υλοποιεί την μέθοδο debit().

(2) **Credit**, που κληρονομεί *(extends)* την Account και υλοποιεί *(implements)* την *iCredit*. Η κλάση περιέχει ένα πλήρη δομητή, και υλοποιεί την μέθοδο credit().

(3) **Metafora**, που κληρονομεί *(extends)* την Account και υλοποιεί *(implements)* την *iMetafora*. Η κλάση περιέχει πλήρη δομητή, και υλοποιεί την μέθοδο metafora(), που εμφανίζει εκτός από το ποσό μεταφοράς και το νέο υπόλοιπο (balance).

Στην κλάση **AccountTest** *(με την main),* θα ορίσετε ένα πίνακα N – αντικειμένων Account, τύπου Debit, Credit και Metafora, και θα εκτελέσετε τις δοσοληψίες. Χρησιμοποιήστε την **instanceof** (π.χ. **If (pinakas[i] instanceof Debit)** {….}).