

## ΑΣΚΗΣΕΙΣ

### Εξαιρέσεις (*Exceptions*) – Λάθη (*Errors*) Χειρισμός των Εξαιρέσεων (*Exception Handling*)

#### ΑΣΚΗΣΗ-1<sup>η</sup> (Δημιουργία κλάσης εξαιρέσεων – έγερση/ρίξιμο εξαιρέσεων-Checked)

Να γραφεί το πρόγραμμα Java που ελέγχει **μέσω των εξαιρέσεων** την ηλικία ενός υποψηφίου που πρέπει να είναι μεταξύ 18 έως 30 ετών. Το πρόγραμμα δημιουργεί την κλάση **AgeException** που κληρονομεί (επεκτείνει) την κλάση **Exception** (με παραμετρικό δομητή (το μήνυμα)), καθώς και τις υποκλάσεις **TooYoungException** και **TooOldException** που κληρονομούν την **AgeException** με αντίστοιχους παραμετρικούς δομητές. Η κλάση **AgeUtils** περιέχει την μέθοδο **checkAge(int age)** - τύπου void, που ελέγχει την ηλικία του υποψηφίου. Αν η ηλικία είναι μικρότερη των 18 ετών ή μεγαλύτερη των 30 ετών τότε εγείρονται οι αντίστοιχες εξαιρέσεις με τα κατάλληλα μηνύματα, διαφορετικά εμφανίζεται το μήνυμα ότι η ηλικία του υποψηφίου είναι σωστή και ο υποψήφιος επιλέγεται. Στην main() δώστε μία ηλικία και γράψτε την κατάλληλη try..catch που θα χειριστεί τις εξαιρέσεις που θα προκύψουν.

#### ΑΣΚΗΣΗ-2<sup>η</sup> (Δημιουργία κλάσης εξαιρέσεων – έγερση/ρίξιμο εξαιρέσεων-Unchecked)

Να γραφεί το πρόγραμμα Java που ελέγχει **μέσω των εξαιρέσεων** (σε ξεχωριστή κλάση) τις κινήσεις ανάληψη (*withdraw*) και κατάθεση (*deposit*) ενός πελάτη τράπεζας. Η κλάση **Account** περιέχει το πεδίο υπόλοιπο (*balance*) με ποσό 5000 Ευρώ και τις μεθόδους *withdraw()* και *deposit()* τύπου void που ρίχνουν/εγείρουν (*throws*) εξαίρεση: (1) για την ανάληψη, όταν το ποσό της ανάληψης είναι μεγαλύτερο του υπολοίπου και (2) για την κατάθεση, όταν το ποσό δεν είναι σωστό, π.χ. αρνητικό. Στις δύο περιπτώσεις θα εμφανίζεται το κατάλληλο μήνυμα. Το πρόγραμμα θα περιέχει την κλάση **MyException** (*unchecked exception*), με πεδίο το message (*String*) και θα εμφανίζει τα ανωτέρω μηνύματα. Στην main() δημιουργήστε αντικείμενο της Account και εκτελέστε δοσοληψίες (αναλήψεις – καταθέσεις). Χρησιμοποιήστε loop – επαναλήψεων και μενού επιλογών. Η επανάληψη τελειώνει όταν συμβεί λάθος στην δοσοληψία.

#### ΑΣΚΗΣΗ-3<sup>η</sup> Παραλλαγή 1<sup>ης</sup> (Δημιουργία κλάσης εξαιρέσεων – έγερση/ρίξιμο εξαιρέσεων - Unchecked)

Να χρησιμοποιήσετε στην λύση της 2<sup>ης</sup> άσκησης την try..catch ώστε να μην διακόπτεται η ροή του προγράμματος όταν συμβαίνει η εξαίρεση, αλλά να συνεχίζει κανονικά μετά την εμφάνιση του κατάλληλου μηνύματος λάθους.

# ΑΛΥΤΗ ΑΣΚΗΣΗ

## ΑΣΚΗΣΗ 1<sup>η</sup>

Για μια εταιρία ενοικιάσεων εξοχικών, ορίστε την αφηρημένη κλάση "ExohikiKatoikia" η οποία έχει χαρακτηριστικά: (α) **typeName**: τύπου String (π.χ. δωμάτιο, μπανγκαλόου, βίλα), (β) **Periohi\_Poli**: τύπου String (π.χ. Ν. Πλάγια - Χαλκιδική), (γ) **typeID**: τύπου int (1=δωμάτιο, 2=μπανγκαλόου, 3=βίλα), (δ) **NrOfDays**: τύπου int (π.χ. 9 μέρες), (ε) **TimiHmeras**: τύπου int (π.χ. 60 Ευρώ) και τις παρακάτω δύο αφηρημένες μεθόδους:

(α) **getEnoikioType()**: η οποία επιστρέφει τον τύπο ενοικίου (String): ('day' για τα δωμάτια, 'week' για τα μπανγκαλόους και 'month' για τις βίλες)

(β) **calcEnoikio()**: η οποία υπολογίζει και επιστρέφει το πληρωτέο ενοίκιο για κάθε ενοικίαση.

Οι κλάσεις "Domatio", "Bungalow" και "Villa" κληρονομούν/εξειδικεύουν την κλάση "ExohikiKatoikia" και διαθέτουν επιπλέον τα παρακάτω χαρακτηριστικά:

Η κλάση Domatio: **DomatioTypeID**: ((int), 1=μονόκλινο, 2=δίκλινο, 3=τρίκλινο)

Η κλάση Bungalow: **PlithosDomation**: (int)

Η κλάση Villa: **PlithosDomation**: (int)

Κάθε αντικείμενο (από τις 3 κλάσεις) υλοποιεί και τις 2 μεθόδους getEnoikioType() και calcEnoikio(). Για τον υπολογισμό του ενοικίου (calcEnoikio()) θα ληφθούν υπόψη: (α) ότι τα bungalows ενοικιάζονται ανά εβδομάδα (μέρες > 7), ενώ οι βίλες ανά μήνα (μέρες > =30) (δες υποδείξεις). (β) για τα bungalows και τις βίλες υπάρχει έκπτωση 10% στο τελικό ποσό ενοικίου.

Στο κυρίως πρόγραμμα δημιουργήστε ένα πίνακα N κατοικιών. Σε κάθε θέση του πίνακα μπορεί να ανατίθεται αντικείμενο τύπου "Domatio" (40%-στον πίνακα), "Bungalow" (30%-στον πίνακα), ή τύπου "Villa" (30%-στον πίνακα). Το πρόγραμμα εμφανίζει τα χαρακτηριστικά κάθε εξοχ. κατοικίας, καθώς και τα αποτελέσματα των μεθόδων getEnoikioType() και calcEnoikio().

### Υποδείξεις:

Ορίστε δομητές και μεθόδους όπου χρειάζεστε. Όλα τα χαρακτηριστικά να είναι private. Υπερβείτε την μέθοδο toString της Object. Φροντίστε να δημιουργείται η εξαίρεση **IllegalArgumentException** σε κάθε περίπτωση που προσπαθήσουμε να καταχωρήσουμε (α) για τα bungalows μέρες ενοικίασης < 7, και για τις βίλες μέρες ενοικίασης < 30.