

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μ.Π.Σ. Επιστήμης Υπολογιστών

Προγραμματισμός με Java™

Φθινοπωρινό Εξάμηνο 2008-2009

ΑΣΚΗΣΕΙΣ

Άσκηση 1

Υλοποιήστε την κλάση `Vehicle` οι οποία περιλαμβάνει πεδία για την ταυτότητα του οχήματος (αλφαριθμητικό), την τρέχουσα κατεύθυνσή του (πραγματικός αριθμός) εκφρασμένη σε μοίρες, και την τρέχουσα ταχύτητα του (πραγματικός αριθμός) εκφρασμένη σε km/h.

Όλα τα πεδία πρέπει να είναι ιδιωτικά και για καθένα πρέπει να παρέχεται μια δημόσια μέθοδος ανάγνωσης με υπογραφή της μορφής `public type getFieldName ()` όπου `type` είναι ο τύπος του πεδίου και `fieldName` το όνομα του πεδίου.

Η κλάση πρέπει να παρέχει ένα δημόσιο κατασκευαστή ο οποίος αρχικοποιεί όλα τα πεδία ενός αντικειμένου τύπου `Vehicle` καθώς επίσης και ένα εξ' ορισμού κατασκευαστή.

Γράψτε ένα απλό πρόγραμμα το οποίο κατασκευάζει ένα αντικείμενο τύπου `Vehicle` και στη συνέχεια τυπώνει στην κονσόλα τις τιμές των πεδίων του.

Άσκηση 2

Βελτιώστε την κλάση `Vehicle` ώστε να περιλαμβάνει την θέση του οχήματος εκφρασμένη σε γεωγραφικό πλάτος (`latitude`) και μήκος (`longitude`). Το γεωγραφικό πλάτος μετράται σε μοίρες στο διάστημα `[-90.0,90.0]`. Οι αρνητικές τιμές αφορούν το Νότιο ημισφαίριο, η τιμή 0 τον ισημερινό και οι θετικές τιμές το Βόρειο ημισφαίριο. Το γεωγραφικό μήκος μετράται σε μοίρες στο διάστημα `[-180.0,180.0]`. Οι αρνητικές τιμές αφορούν το Ανατολικό ημισφαίριο, η τιμή 0 τον μεσημβρινό του Greenwich και οι θετικές τιμές το Δυτικό ημισφαίριο.

Υλοποιήστε την κλάση `PassengerVehicle` η οποία παριστά επιβατικά οχήματα. Κάθε τέτοιο όχημα, εκτός από τις ιδιότητες ενός τυπικού οχήματος περιλαμβάνει το συνολικό πλήθος θέσεων και το τρέχον πλήθος επιβατών.

Άσκηση 3

Υλοποιήστε μια απλή εφαρμογή κονσόλας η οποία δημιουργεί και τυπώνει τυχαίες διανομές `bridge`. Στο παιχνίδι συμμετέχουν τέσσερις παίκτες που ονομάζονται Βορράς, Δύση, Νότος και Ανατολή. Σε κάθε διανομή κάθε παίκτης παίρνει 13 φύλλα της τράπουλας.

Η κλάση `Bridge` πρέπει να ορίζει μια στατική φωλιασμένη κλάση `Card` η οποία παριστάνει ένα φύλλο. Κάθε φύλλο προσδιορίζεται από την αξία και το χρώμα του. Για να παραστήσετε τις επιτρεπτές τιμές των αντίστοιχων πεδίων της κλάσης `Card` θα πρέπει να δημιουργήσετε τους απεριθμητούς τύπους `Value` με τιμές `TWO(2)`, `THREE(3)`, `FOUR(4)`, `FIVE(5)`, `SIX(6)`, `SEVEN(7)`, `EIGHT(8)`, `NINE(9)`, `TEN(10)`, `JACK(J)`, `QUEEN(Q)`, `KING(K)`, `ACE(A)` και `Suite` με τιμές `CLUBS(♣)`, `SPADES(♠)`, `DIAMONDS(♦)`, `HEARTS(♥)`. Προκειμένου να διευκολυνθεί η εκτύπωση οι τύποι αυτοί πρέπει να παρέχουν μια ετικέτα (ένα επιπλέον πεδίο τύπου `String` με όνομα `label`) το οποίο θα χρησιμοποιείται για την εκτύπωση των τιμών 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A και ♣, ♦, ♥, ♠ αντίστοιχα. Φυσικά απαιτείται και μια δημόσια μέθοδος `String label()` για την ανάγνωση των τιμών των ετικετών.

Άσκηση 4

Υλοποιήστε μια παραμετρική συνδεδεμένη λίστα μονής σύνδεσης ορίζοντας τον τύπο `LinkedList<T>`.

Η υλοποίηση πρέπει να υποστηρίζει, εισαγωγή, αναζήτηση και διαγραφή. Για τη σύγκριση αντικειμένων χρησιμοποιήστε την μέθοδο `equals` της κλάσης `java.lang.Object`.

Άσκηση 5

Ορίστε την επισημείωση `@MaxLength` η οποία υποστηρίζει μία και μόνο ιδιότητα `value`. Η επισημείωση πρέπει να εφαρμόζεται μόνο σε πεδία και να διατηρείται στον μεταγλωττισμένο κώδικα.

Όταν η επισημείωση εφαρμόζεται σε ένα πεδίο τύπου `java.lang.String` καθορίζει το μέγιστο αριθμό των χαρακτήρων που πρέπει να έχει το πεδίο ώστε η τιμή του να θεωρείται έγκυρη. Αντίστοιχα, όταν η επισημείωση εφαρμόζεται σε ένα πεδίο τύπου συστοιχίας καθορίζει το μέγιστο επιτρεπτό πλήθος των στοιχείων. Σε κάθε άλλη περίπτωση η επισημείωση δεν έχει νόημα.

Στη συνέχεια ολοκληρώστε την υλοποίηση της μεθόδου `validateLength` της παρακάτω κλάσης έτσι ώστε να ελέγχει αν οι τιμές των επισημειωμένων πεδίων του ορίσμάτος της ικανοποιούνται.

```
public Class TestClass {
    @MaxLength(12)
    String string;

    @MaxLength(4)
    int[] array;

    public static boolean validateLength (Object instance) {
        if (instance == null) {
            return true;
        }

        Class<?> type = instance.getClass();
        Field[] fields = type.getDeclaredFields();
        for (Field field : fields) {
            if (field.getType().isArray() || field.getType() == String.class) {
                MaxLength annotation = field.getAnnotation(MaxLength.class);
                if (annotation != null) {
                    // ...
                }
            }
        }
        return true;
    }

    public static void main (String[] args) {
        TestClass instance = new TestClass();
        instance.string = "1234";
        instance.array = new int[4];

        boolean valid = validate(instance);
        System.out.println("Object is " + valid ? "valid" : "not valid");
    }
}
```

Άσκηση 6

Υλοποιείστε μια εφαρμογή η οποία δέχεται ως όρισμα το όνομα μιας κλάσης, φορτώνει αυτή την κλάση και κατασκευάζει ένα αντικείμενό της χρησιμοποιώντας τον εξ' ορισμού κατασκευαστή. Στην συνέχεια η εφαρμογή διαβάζει από την κονσόλα εντολές της μορφής `READ property` ή `SET property value, property` είναι το όνομα μιας ιδιότητας και `value` είναι ένα αλφαριθμητικό που παριστά την τιμή της ιδιότητας. Πρέπει να υποστηρίζονται όλοι οι απλοί τύποι δεδομένων καθώς επίσης ο τύπος δεδομένων `java.lang.String`.

Άσκηση 7

Υλοποιείστε την εφαρμογή `LineCount` η οποία τυπώνει το μονοπάτι και το πλήθος γραμμών όλων των αρχείων τα οποία βρίσκονται σε ένα καθορισμένο υπόδεντρο του συστήματος αρχείων και το όνομα των οποίων ταιριάζει με μια δοσμένη κανονική έκφραση, π.χ.

```
java LineCount /home/me/java/src .*\.java.
```

Η εφαρμογή θα πρέπει επίσης να τυπώνει το πλήθος των αρχείων και το συνολικό αριθμό γραμμών σε όλα τα αρχεία.

Άσκηση 8

Υλοποιείστε την εφαρμογή `Concord` η οποία δέχεται πανομοιότυπα με την εφαρμογή `LineCount` της Άσκησης 7 ορίσματα. Η εφαρμογή πρέπει να κατασκευάζει ένα ευρετήριο λέξεων. Για κάθε λέξη στο ευρετήριο η εφαρμογή διατηρεί μια λίστα εμφανίσεων. Κάθε στοιχείο της λίστας εμφανίσεων αποθηκεύει το μονοπάτι του αρχείου και τον αριθμό γραμμής όπου εντοπίστηκε η λέξη.

Μετά την κατασκευή του ευρετηρίου, και ως ότου τερματιστεί, η εφαρμογή θα πρέπει να διαβάζει από την κονσόλα μία λέξη και να τυπώνει τον κατάλογο των εμφανίσεών της.