



Όνοματεπώνυμο: [REDACTED]

Α.Μ. [REDACTED]

Γράψτε τα στοιχεία σας σε όλα τα φύλλα που θα παραλάβετε καθώς και στο έντυπο των θεμάτων.
Στο τέλος παραδώστε όλα τα φύλλα που παραλάβετε, ακόμα και αν είναι μόνο πρόχειρα, καθώς και το έντυπο των θεμάτων.

[42] **Θέμα 1**

Θέλουμε να κρατήσουμε πληροφορίες για τους πόντους που πέτυχε κάθε παίκτης μιας ομάδας μπάσκετ σε έναν αγώνα.

- [11] α) Στον ακόλουθο κώδικα η πληροφορία αυτή αποθηκεύεται σε δύο παράλληλους πίνακες: έναν με τα ονόματα (names) και έναν με τους πόντους (points). Στην αρχή εισάγονται τα στοιχεία από το πληκτρολόγιο και κατόπιν καλείται μια συνάρτηση για την ταξινόμηση των παικτών (και των αντίστοιχων πόντων) κατά αλφαβητική σειρά ονόματος.

Ο κώδικας αυτός έχει σφάλματα και σημεία τα οποία καλείστε να συμπληρώσετε. Ξαναγράψτε τον κώδικα διορθώνοντας όλα τα σφάλματα που θα εντοπίσετε και συμπληρώνοντάς τον ώστε να είναι πλήρης.

```
1 public class BasketStats {
2     public static void main(String[] args) {
3         String[12] names;
4         int[12] points;
5
6         Scanner input;
7
8         // Διάβασμα των ονομάτων και των πόντων από το πληκτρολόγιο.
9         for (int i = 0; i < names.length; i++) {
10             names[i] = ...;
11             points[i] = ...;
12         }
13
14         sortByName();
15
16     }
17
18     sortByName() {
19         // Ταξινόμηση φουσάιδας. Οι επαναλήψεις δεν έχουν σφάλματα!
20         for (int i = 1; i < names.length; i++) {
21             for (int j = names.length-1; j >= i; j--) {
22                 // Σύγκριση τα ονόματα των παικτών j-1 και j
23                 // και, αν χρειάζεται, βάλτε τα στοιχεία τους στη σωστή σειρά
24                 // ...
25             }
26         }
27     }
28 }
```

β) Μετατρέψτε τον παραπάνω κώδικα σε αντικειμενοστρεφή.

- [4] i. Δημιουργήστε μία κλάση Player η οποία θα αντιπροσωπεύει έναν παίκτη, με πεδία το όνομα (name) και τους πόντους του (points). Το όνομα του παίκτη δεν θα πρέπει να αλλάζει μετά τη δημιουργία του.
- [4] ii. Προσθέστε μέθοδο δημιουργίας (constructor) στην κλάση Player η οποία να δίνει τιμές στα δύο πεδία.
- [2] iii. Προσθέστε μέθοδο πρόσβασης (get) για ένα από τα πεδία.
- [11] iv. Προσθέστε μέθοδο toString() στην κλάση η οποία να επιστρέφει τις πληροφορίες του παίκτη στη μορφή (για παράδειγμα): Giannis: 45 points
- [12] v. Ξαναγράψτε τον κώδικα του ερωτήματος (α) χρησιμοποιώντας την κλάση Player και αντικαθιστώντας τους δύο παράλληλους πίνακες με κατάλληλο ArrayList<>. Αν χρειάζεστε περισσότερα στοιχεία στην κλάση Player από αυτά που έχετε γράψει στα προηγούμενα υποερωτήματα, να τα προσθέσετε.

4

- vi. Στο κύριο πρόγραμμα του προηγούμενου υποερωτήματος προσθέστε κώδικα για την εκτύπωση των πληροφοριών των παικτών μετά την ταξινόμηση, αξιοποιώντας τη μέθοδο `toString()` που έχετε αναπτύξει. Για πλήρεις μονάδες, χρησιμοποιήστε τη δομή `for-each` για το σκοπό αυτό.

4. Θέμα 2

Δίνονται οι ακόλουθες κλάσεις που αντιπροσωπεύουν διάφορους τύπους πληρωμών, και τα στοιχεία κάθε μίας:

- **Payment:** Μία οποιαδήποτε πληρωμή. Στοιχεία: το ποσό της πληρωμής, π.χ. 5.99.
- **CashPayment:** Μία πληρωμή με μετρητά. Στοιχεία: το ποσό της πληρωμής.
- **CardPayment:** Μία πληρωμή με κάρτα. Στοιχεία: το ποσό της πληρωμής, ο αριθμός της κάρτας.
- **CreditCardPayment:** Μία πληρωμή με πιστωτική κάρτα. Στοιχεία: το ποσό της πληρωμής, ο αριθμός της κάρτας, το πλήθος των δόσεων.
- **DebitCardPayment:** Μία πληρωμή με χρεωστική κάρτα. Στοιχεία: το ποσό της πληρωμής, ο αριθμός της κάρτας.

10. α) Εντάξτε τις κλάσεις αυτές σε μία ιεραρχία. Γράψτε τις δηλώσεις και τα πεδία κάθε κλάσης.
- β) Θα μπορούσαν ή θα έπρεπε κάποιες από αυτές τις κλάσεις να χαρακτηριστούν `abstract`; Αιτιολογήστε.
- γ) Θα μπορούσαν ή θα έπρεπε κάποιες από αυτές τις κλάσεις να είναι `interface`; Αιτιολογήστε.
- δ) Ποιου/ποίων τύπου/τύπων πρόσβαση θα μπορούσε να έχει το πεδίο που αποθηκεύει τον αριθμό της κάρτας στην κλάση `CardPayment` ώστε να είναι άμεσα προσβάσιμο από τις υποκλάσεις της; Αιτιολογήστε.
- ε) Υλοποιήστε όπου και όπως χρειάζεται μεθόδους δημιουργίας (`constructor`) στις παραπάνω κλάσεις έτσι ώστε να μπορεί να εκτελεστεί ο κώδικας

```
CreditCardPayment ccr = new CreditCardPayment("9900123478458956", 12, 579.99);
```

όπου η πρώτη παράμετρος είναι ο αριθμός της πιστωτικής κάρτας, η δεύτερη είναι ο αριθμός των δόσεων και η τρίτη είναι το ποσό της πληρωμής.

- στ) Υλοποιήστε μέθοδο `printPaymentInfo()` στην κλάση `CreditCardPayment` η οποία να τυπώνει της πληροφορίες της πληρωμής ως εξής:

Πληρωμή: 579.99€ με πιστωτική κάρτα *****8956 σε 12 δόσεις.

για την παραπάνω πληρωμή. Στον αριθμό της πιστωτικής κάρτας, όλα τα ψηφία εκτός από τα τέσσερα τελευταία έχουν αντικατασταθεί με `****` (μόνο για την εκτύπωση, ο αριθμός της κάρτας διατηρείται στο πεδίο πλήρης). Επισημαίνεται ότι δεν έχουν όλοι οι αριθμοί πιστωτικής κάρτας το ίδιο πλήθος ψηφίων. Αν χρειάζονται επιπλέον στοιχεία σε άλλες κλάσεις να τα υλοποιήσετε επίσης.

Θέμα 3

Σε ένα αρχείο είναι γραμμένοι με τη βοήθεια της `RandomAccessFile`, πραγματικοί αριθμοί (`double` - 8 bytes). Επιθυμούμε να υλοποιήσουμε πρόγραμμα για το εξής πρόβλημα:

1. Ο χρήστης δίνει την θέση ενός αριθμού, έστω `i`.
2. Εντοπίζουμε στο αρχείο τον `i`-οστό αριθμό (ο πρώτος αριθμός του αρχείου αντιστοιχεί σε θέση 1) και τον διαβάζουμε. Εφόσον η θέση `i` είναι μεγαλύτερη από το πλήθος των στοιχείων του αρχείου, δεν κάνουμε τίποτα.
3. Τα παραπάνω δύο βήματα επαναλαμβάνονται μέχρις ότου ο χρήστης να δώσει θέση 0 (μηδέν).
4. Στο τέλος εκτυπώνουμε τον μέσο όρο των αριθμών που διαβάστηκαν από το αρχείο (χωρίς να λαμβάνονται υπόψη θέσεις που δόθηκαν εκτός ορίων).

- α) Υλοποιήστε το ζητούμενο χρησιμοποιώντας την κλάση `RandomAccessFile`.

- β) Περιγράψτε (χωρίς απαραίτητα να γράψετε πλήρη κώδικα) τις διαφορές στην υλοποίηση αν οι αριθμοί ήταν γραμμένοι σιριακά με την `ObjectOutputStream` και τους διαβάζατε αντίστοιχα με την `ObjectInputStream`. Αναφερθείτε ιδιαίτερα στον εντοπισμό του `i`-οστού αριθμού και στον τρόπο με τον οποίο θα διαπιστώναμε ότι η θέση `i` είναι μεγαλύτερη από το πλήθος των στοιχείων.