

Πρώτη Σειρά ασκήσεων
Ημερομηνία Παράδοσης: 21 Απριλίου 2015, 3 μ.μ.

Οι παρακάτω ασκήσεις πρέπει να παραδοθούν μέχρι τις 21/4/2015 3μ.μ. . Στην υλοποίηση των κλάσεων σας δεν θα πρέπει να έχετε public μεταβλητές. Βαθμοί θα αφαιρεθούν για προγράμματα που δεν είναι καλά γραμμένα, δηλαδή δεν είναι σωστά στοιχισμένα ή δεν έχουν καλά επιλεγμένα ονόματα μεταβλητών ώστε να διαβάζονται εύκολα.

Άσκηση 1

Για την άσκηση αυτή θα υλοποιήσετε σε Java τον Αφηρημένο Τύπο Δεδομένων **Ουρά (Queue)**. Η ουρά αποθηκεύει στοιχεία και μας δίνει πρόσβαση σε αυτά με βάση την σειρά με την οποία ήρθαν, δηλαδή First-In-First-Out (FIFO). Η ουρά που θα υλοποιήσετε θα έχει σταθερή χωρητικότητα (**capacity**), που είναι ο μέγιστος αριθμός στοιχείων που μπορεί να αποθηκεύσει. Για την υλοποίηση σας θα χρησιμοποιήσετε ένα **πίνακα** (όχι `ArrayList`). Θα πρέπει να κρατάτε την θέση του πρώτου (**head**) και του τελευταίου (**tail**) στοιχείου στην ουρά. Οι θέσεις αυτές θα πρέπει να κινούνται *κυκλικά* όπως θα προστίθενται ή θα αφαιρούνται στοιχεία από την ουρά. Δηλαδή, αν φτάσουμε στο τέλος του πίνακα ξεκινάμε ξανά από την αρχή.

Κατασκευάστε την κλάση **Queue** που υλοποιεί μια ουρά που αποθηκεύει ακεραίους. Η κλάση θα πρέπει να έχει τις εξής μεθόδους:

- Ένα **constructor** χωρίς ορίσματα που αρχικοποιεί την ουρά με χωρητικότητα `capacity = 100`.
- Ένα **constructor** που αρχικοποιεί την ουρά με όρισμα την χωρητικότητα του πίνακα.
- Μία μέθοδο **insert** η οποία παίρνει σαν όρισμα ένα ακέραιο και τον προσθέτει στο τέλος της ουράς. Επιστρέφει μία boolean τιμή αν έγινε σωστά η προσθήκη του στοιχείου. Η προσθήκη θα πρέπει να γίνεται σε σταθερό χρόνο, δηλαδή δεν επιτρέπεται να μετακινήσετε τα υπάρχοντα στοιχεία της ουράς.
- Μία μέθοδο **remove** η οποία αφαιρεί και επιστρέφει τον πρώτο ακέραιο στην κορυφή της ουράς. Η αφαίρεση θα πρέπει να γίνεται σε σταθερό χρόνο, δηλαδή δεν επιτρέπεται να μετακινήσετε τα υπάρχοντα στοιχεία της ουράς.
- Την μέθοδο **toString** η οποία επιστρέφει ένα αλφαριθμητικό με τα στοιχεία της ουράς σε FIFO σειρά, χωρισμένα με κενό.
- Την μέθοδο **equals** η οποία ελέγχει αν η ουρά είναι ίδια με κάποια άλλη. Ισότητα σημαίνει ότι οι δύο ουρές περιέχουν ακριβώς τα ίδια στοιχεία με ακριβώς την ίδια σειρά.
- Την μέθοδο **isEmpty** η οποία επιστρέφει true ή false ανάλογα αν η ουρά είναι άδεια ή όχι.
- Μια μέθοδο **getSize** που επιστρέφει τον αριθμό στοιχείων στην ουρά.
- Την **μέθοδο πρόσβασης (accessor method)** για την χωρητικότητα (`capacity`) της ουράς.

Ορίστε μια μέθοδο `main` για να τεστάρετε την κλάση σας. Για να σας βαθμολογήσουμε θα φτιάξουμε μια κλάση **QueueTest** που θα χρησιμοποιεί την `Queue` που θα παραδώσετε. Ένα παράδειγμα δίνεται στην σελίδα του μαθήματος.

Υποδείξεις

- Το υπόλοιπο διαίρεσης δίνεται με τον τελεστή `%`.
- Για να πάρετε όλους τους βαθμούς θα πρέπει να φροντίσετε και τις οριακές καταστάσεις (δηλαδή τις περιπτώσεις που μπορεί να δημιουργηθεί κάποιο λάθος).

Άσκηση 2

Για την άσκηση αυτή θα υλοποιήσετε σε Java ένα πρόγραμμα που θα υλοποιεί μια απλή εκδοχή του επιτραπέζιου παιχνιδιού Κινέζικα Ζευγάρια. Στο παιχνίδι αυτό υπάρχουν $2*N$ κάρτες που έχουν ζευγάρια από εικόνες. Τοποθετούνται στο τραπέζι και σε κάθε γύρο ο κάθε παίχτης ανοίγει δύο από τις κάρτες. Αν οι κάρτες είναι ίδιες ο παίχτης κερδίζει ένα πόντο και οι κάρτες μένουν ανοιχτές. Αλλιώς οι κάρτες κλείνουν και συνεχίζει ο επόμενος παίχτης.

Στην απλοποιημένη εκδοχή που θα υλοποιήσετε αντί για ζευγάρια κάρτες με εικόνες θα έχουμε N ζευγάρια με αριθμούς από το 0 έως το $N-1$. (Παρακάτω όποτε μιλάμε για «κάρτες» θα εννοούμε τους αριθμούς $0...N-1$. Ο κάθε αριθμός εμφανίζεται δύο φορές.) Οι αριθμοί είναι τοποθετημένοι σε ένα μονοδιάστατο πίνακα $2*N$ θέσεων. Το παιχνίδι θα παίζεται μεταξύ ενός παίχτη-ανθρώπου και ενός παίχτη-υπολογιστή. Σε κάθε γύρο ο παίχτης-άνθρωπος ανοίγει δύο θέσεις του πίνακα. Αν είναι οι ίδιοι αριθμοί κερδίζει ένα πόντο, και το ζευγάρι μένει ανοιχτό. Αλλιώς το ζευγάρι εμφανίζεται για λίγο και μετά κλείνει. Ο παίχτης-υπολογιστής επίσης επιλέγει δύο θέσεις με μια τακτική που περιγράφεται παρακάτω. Το παιχνίδι τελειώνει όταν όλα τα ζευγάρια έχουν βρεθεί.

Για την υλοποίησή σας θα πρέπει να δημιουργήσετε τέσσερις κλάσεις. Την κλάση **Board** η οποία κρατάει πληροφορίες για το «τραπέζι» και τις «κάρτες». Την κλάση **ComputerPlayer** που υλοποιεί το παιχνίδι του παίχτη-υπολογιστή. Την κλάση **HumanPlayer** που χειρίζεται το παιχνίδι του παίχτη-ανθρώπου. Την κλάση **ChinesePairs** που υλοποιεί την ροή του παιχνιδιού.

Board: Η πιο σημαντική κλάση που κρατάει πληροφορία για τις «κάρτες» και την εκτύπωση. Οι κάρτες-αριθμοί θα τοποθετηθούν σε ένα μονοδιάστατο πίνακα ακεραίων με $2*N$ θέσεις, όπου N είναι ο αριθμός των ζευγών. Επίσης χρειάζεται να κρατάει πληροφορία για το ποιες και πόσες θέσεις του πίνακα έχουν ανοιχτεί, δηλαδή περιέχουν κάρτες από ζευγάρια που έχουν βρεθεί από τους παίχτες.

Η κλάση θα πρέπει να έχει τις εξής μεθόδους:

- Τον **constructor** ο οποίος παίρνει σαν όρισμα τον αριθμό των ζευγών N . Ο constructor είναι υπεύθυνος για την αρχικοποίηση όλων των μεταβλητών, καθώς και για την τοποθέτηση των $2*N$ καρτών σε τυχαίες θέσεις μέσα στον πίνακα. Ένας τρόπος για την τυχαία τοποθέτηση είναι να βάλετε αρχικά όλες τις κάρτες-αριθμούς με την σειρά και να κάνετε μια τυχαία αναδιάταξη των καρτών. Ένας άλλος είναι για κάθε κάρτα να διαλέξετε τυχαία μια από τις εναπομείναντες θέσεις.
- Την μέθοδο **print** η οποία θα εκτυπώνει το «τραπέζι». Για κάθε μη-ανοιγμένη θέση θα τυπώνει '*' ενώ για κάθε ανοιγμένη θέση θα τυπώνει τον αριθμό σε εκείνη την θέση. Θα είναι χρήσιμο να τυπώνετε και μία επικεφαλίδα με το νούμερο της θέσης.
- Υπερφόρτωση της **print** η οποία παίρνει σαν όρισμα δύο θέσεις του πίνακα (τις οποίες έχει διαλέξει κάποιος παίχτης). Τυπώνει όπως και πριν αλλά επιπλέον τυπώνει και τους αριθμούς στις δύο θέσεις τις οποίες διάλεξε ο χρήστης. Αυτή η εκτύπωση θα εμφανίζεται μόνο για μερικά δευτερόλεπτα και μετά θα τυπώνεται από πάνω το ίδιο όπως και με την **print** (δηλαδή ουσιαστικά θα αντικαθιστά τους αριθμούς στις θέσεις που διάλεξε ο παίχτης με '*'), εκτός κι αν ο χρήστης βρήκε κάποιο από τα ζευγάρια. Βοηθητικό κώδικα για το πώς θα το κάνετε την εκτύπωση και την καθυστέρηση θα βρείτε στις υποδείξεις.
- Μια μέθοδο **checkPositions** η οποία θα παίρνει σαν όρισμα δύο θέσεις του πίνακα και θα ελέγχει αν οι αριθμοί σε αυτές τις θέσεις είναι ίδιοι. Αν ναι, θα «ανοίγει» τις θέσεις αυτές και θα επιστρέφει true. Αλλιώς θα επιστρέφει false. Αν κάποια από τις θέσεις είναι ήδη ανοιγμένη, θα επιστρέφει false.
- Μία μέθοδο **getRandomPosition** η οποία επιστρέφει μια τυχαία θέση στον πίνακα από αυτές που δεν έχουν ανοιχτεί ακόμη.
- Μία μέθοδο **getCard** η οποία παίρνει μία θέση του πίνακα και επιστρέφει τον αριθμό-κάρτα σε εκείνη την θέση.
- Μία μέθοδο **allOpen** η οποία επιστρέφει true αν όλες οι κάρτες έχουν ανοιχτεί.

Μπορείτε να ορίσετε και άλλες public ή private κλάσεις εφόσον το θεωρείτε απαραίτητο.

ComputerPlayer: Η κλάση αυτή θα υλοποιεί το παιχνίδι του παίχτη-υπολογιστή. Ο υπολογιστής θα διατηρεί ένα πίνακα με N θέσεις όπου για κάθε αριθμό-κάρτα που θα κρατάει την θέση στην οποία τον είδε τις προηγούμενες

φορές που έπαιξε ο παίχτης-υπολογιστής, ή μία default τιμή αν δεν τον έχει ξαναδεί. Επίσης θα κρατάει πληροφορία για τους πόντους που έχει μαζέψει ο παίχτης.

Θα έχει τις εξής μεθόδους:

- Τον **constructor** ο οποίος παίρνει σαν όρισμα τον αριθμό των ζευγαριών και αρχικοποιεί τις μεταβλητές.
- Την μέθοδο **play** η οποία θα παίρνει σαν όρισμα ένα αντικείμενο Board και θα υλοποιεί την βασική λογική του παίχτη. Την πρώτη θέση ο παίχτης την διαλέγει τυχαία, από τις θέσεις που δεν έχουν ανοιχτεί ακόμη. Αν τον αριθμό σε αυτή την θέση τον έχει ξαναδεί σε κάποια άλλη θέση στο παρελθόν τότε ανοίγει και την άλλη θέση. Αν όχι, επιλέγει και την δεύτερη θέση τυχαία, από τις θέσεις που δεν έχουν ανοιχτεί. Οι θέσεις που θα επιλέξει τελικά θα πρέπει να είναι διαφορετικές μεταξύ τους. Θα εκτυπώνονται οι θέσεις που διάλεξε ο υπολογιστής, και θα εκτυπώνεται το «τραπέζι» με ανοιχτές τις επιλογές του χρήστη. Για κάθε αριθμό-κάρτα που βλέπει θα κρατάει την θέση στην οποία εμφανίστηκε. Η μέθοδος θα ενημερώνει και τους πόντους του παίχτη αν βρήκε ένα ζευγάρι.
- Μία **μέθοδο προσπέλασης (accessor method)** για τον αριθμό των πόντων που έχει μαζέψει ο παίχτης.

HumanPlayer: Η κλάση αυτή θα υλοποιεί το παιχνίδι του παίχτη-ανθρώπου. Η κλάση θα πρέπει να κρατάει πληροφορία για τους πόντους που έχει μαζέψει ο παίχτης.

Θα έχει τις εξής μεθόδους:

- Την μέθοδο **play** η οποία θα παίρνει σαν όρισμα ένα αντικείμενο Board και θα υλοποιεί το παιχνίδι του χρήστη. Θα του ζητάει να δώσει δύο θέσεις. Θα πρέπει οι δύο θέσεις να είναι διαφορετικές. Θα εκτυπώνει το «τραπέζι» με ανοιχτές τις επιλογές του χρήστη. Αν βρέθηκε ένα ζευγάρι θα ενημερώνει τους πόντους του παίχτη.
- Μία **μέθοδο προσπέλασης (accessor method)** για τον αριθμό των πόντων που έχει μαζέψει ο παίχτης.

ChinesePairs: Η κλάση που περιέχει την μέθοδο **main**, η οποία υλοποιεί τη βασική ροή του παιχνιδιού. Δημιουργεί τα αντικείμενα για το Board και τους παίχτες. Οι παίχτες παίζουν εναλλάξ. Στο τέλος του γύρου τυπώνονται οι πόντοι και των δύο παιχτών. Το παιχνίδι σταματάει όταν βρεθούν όλα τα ζευγάρια και ανακηρύσσεται ο τελικός νικητής (αν δεν είναι ισοπαλία).

Υποδείξεις:

- Για την δημιουργία τυχαίων αριθμών χρησιμοποιήστε την κλάση Random.
- Για την εκτύπωση μπορεί να σας βολεύει να χρησιμοποιήσετε την printf η οποία σας επιτρέπει να δώσετε συγκεκριμένο χώρο για ένα ακέραιο. Π.χ., printf("%3d ", i) θα δώσει 3 θέσεις για την εκτύπωση του ακεραίου i ανεξάρτητα από τον αριθμό ψηφίων του i.
- Για να τυπώσετε πάνω από μία γραμμή την οποία έχετε ήδη τυπώσει και να την σβήσετε, μπορείτε να χρησιμοποιήσετε τον ειδικό χαρακτήρα '\r' ο οποίος σας πάει στην αρχή της γραμμής στην οποία είσαστε και ξεκινάει την εκτύπωση από εκεί. Για παράδειγμα ο κώδικας

```
System.out.print("a ");  
System.out.print("b ");  
System.out.print("\r");  
System.out.print("c ");  
System.out.print("d ");
```

θα τυπώσει "c d " γιατί όταν κληθεί η εντολή System.out.print("\r"); θα ξεκινήσει η εκτύπωση ξανά από την αρχή της γραμμής

- Για να σταματήσετε την εκτέλεση του προγράμματος για κάποια δευτερόλεπτα σας δίνεται η μέθοδος delay η οποία παίρνει σαν όρισμα τον αριθμό από δευτερόλεπτα. Ο κώδικας για την μέθοδο είναι στο αρχείο delay_method.txt στη σελίδα του μαθήματος.