

Πρώτη Σειρά ασκήσεων
Ημερομηνία Παράδοσης: Κυριακή 31 Μαρτίου 11:59 μ.μ.

Οι παρακάτω ασκήσεις πρέπει να παραδοθούν μέχρι τις 31 Μαρτίου τα μεσάνυχτα. 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 ώστε να διαβάζονται εύκολα.

Άσκηση 1

Για την άσκηση αυτή θα υλοποιήσετε σε Java ένα **δυναμικό πίνακα**. Ο δυναμικός πίνακας είναι μια υλοποίηση του Αφηρημένου Τύπου Δεδομένων **πίνακα**, ο οποίος αποθηκεύει στοιχεία σειριακά και προσφέρει τυχαία πρόσβαση στα στοιχεία που αποθηκεύει με βάση τη θέση τους. Επιτρέπει επίσης προσθήκη και αφαίρεση στοιχείων από το τέλος του πίνακα. Η δομή είναι πολύ παρόμοια με το array που ξέρετε από την Python. Η διαφορά με τους πίνακες της Java που έχετε μάθει μέχρι τώρα είναι ότι δεν έχει σταθερό μέγεθος, αλλά το μέγεθος του αλλάζει ανάλογα με το πόσα στοιχεία περιέχει.

Ο δυναμικός πίνακας δουλεύει ως εξής: Αρχικά δεσμεύεται ένας χώρος μνήμης μιας συγκεκριμένης χωρητικότητας (capacity) C , όπου C είναι μια σταθερά. Ο πίνακας αρχικά είναι άδειος. Νέα στοιχεία προστίθενται στην επόμενη διαθέσιμη θέση του πίνακα. Αν η προσθήκη ενός νέου στοιχείου έχει ως αποτέλεσμα να ξεπεραστεί η χωρητικότητα του πίνακα, τότε νέα ποσότητα μνήμης διπλάσιας χωρητικότητας δεσμεύεται και όλα τα στοιχεία του πίνακα αντιγράφονται σε αυτό το νέο χώρο μνήμης και το νέο στοιχείο προστίθεται στην κατάλληλη θέση. Όταν αφαιρούμε στοιχεία από τον πίνακα, αφαιρούμε πάντα το τελευταίο στοιχείο του πίνακα. Αν η αφαίρεση έχει ως αποτέλεσμα το μέγεθος του πίνακα (ο αριθμός των στοιχείων) να γίνει το $\frac{1}{4}$ της παρούσας χωρητικότητας και η παρούσα χωρητικότητα είναι τουλάχιστον δύο φορές το C , τότε δεσμεύεται νέος χώρος μνήμης της μισής χωρητικότητας και τα στοιχεία του πίνακα αντιγράφονται στο νέο χώρο μνήμης.

Κατασκευάσετε μια κλάση **Array** που υλοποιεί ένα δυναμικό πίνακα από ακεραίους. Η κλάση θα πρέπει να έχει τις εξής μεθόδους:

- Ένα Constructor χωρίς ορίσματα που αρχικοποιεί τον πίνακα με χωρητικότητα $C = 4$.
- Ένα Constructor που αρχικοποιεί τον πίνακα με όρισμα την χωρητικότητα του πίνακα.
- Μία μέθοδο **add(int x)** η οποία παίρνει σαν όρισμα ένα ακέραιο και τον προσθέτει στο τέλος του πίνακα, και διπλασιάζει την χωρητικότητα του πίνακα αν χρειάζεται.
- Μία μέθοδο **delete()** η οποία αφαιρεί το τελευταίο στοιχείο του πίνακα, και υποδιπλασιάζει το μέγεθος του πίνακα, αν είναι απαραίτητο
- Μια μέθοδο **get(int i)** η οποία επιστρέφει το στοιχείο στη θέση i .
- Μια μέθοδο **set(int i, int x)** η οποία αναθέτει στην θέση i την τιμή x , εφόσον η θέση i περιέχει ένα από τα στοιχεία του πίνακα.
- Μια μέθοδο **getSize()** που επιστρέφει το μέγεθος (αριθμό στοιχείων) του πίνακα.
- Μια μέθοδο **getCapacity()** που επιστρέφει τη χωρητικότητα του πίνακα.

Ορίστε μια κλάση **ArrayTest** που θα περιέχει τη main και θα τεστάρει την κλάση σας. Ορίστε ένα αντικείμενο της κλάσης Array, με αρχική χωρητικότητα $C = 4$. Κάνετε 2 Add, 1 Delete, 9 Add, 8 Delete, καθώς και μερικές προσβάσεις σε στοιχεία του πίνακα και μία ανάθεση. Σε κάθε κλήση τυπώνετε το μέγεθος και τη χωρητικότητα του πίνακα.

Υποδείξεις

- Για να πάρετε όλους τους βαθμούς θα πρέπει να φροντίσετε και τις οριακές καταστάσεις (δηλαδή τις περιπτώσεις που μπορεί να δημιουργηθεί κάποιο λάθος, π.χ. να προσπαθήσουμε να διαβάσουμε κάποια θέση εκτός ορίων ενός πίνακα).
- Για να δεσμεύσετε χώρο μνήμης χρησιμοποιήστε την new.
- Ορίστε την C ως μεταβλητή της κλάσης με σταθερή τιμή 4. Αυτή είναι η ελάχιστη χωρητικότητα που μπορεί να έχει ο πίνακας.

Άσκηση 2

Για την άσκηση αυτή θα προγραμματίσετε σε Java ένα παιχνίδι blackjack (γνωστό και ως 21), που θα επιτρέπει σε ένα χρήστη να παίζει ενάντια στον υπολογιστή, όπως θα έπαιζε εναντίον ενός dealer στο καζίνο.

Κανόνες του παιχνιδιού

Στο blackjack ο στόχος είναι να πάρεις χαρτιά (χέρι) ώστε το σύνολο των πόντων τους να είναι όσο πιο κοντά στο 21 γίνεται (χωρίς να το ξεπερνάει), και να είναι περισσότεροι από αυτούς του dealer. Τα χαρτιά δίνουν τόσους πόντους όσο το νούμερο τους, οι φιγούρες δίνουν 10, και ο άσσος δίνει 1 ή 11 πόντους, ότι είναι καλύτερο για το σκορ του παίχτη. Το παιχνίδι παίζεται ως εξής. Ο παίχτης ποντάρει ένα ποσό (bet). Από ένα ρεύμα χαρτιών (river) που αποτελείται από μία ή παραπάνω τράπουλες, δίνονται δύο χαρτιά στον παίχτη. Σε αυτό το σημείο ο παίχτης έχει τρεις επιλογές. Μπορεί να διπλασιάσει το στοίχημα (double), να τραβήξει (hit) ή να σταματήσει (stand). Αν διπλασιάσει, τότε διπλασιάζεται το στοίχημα, και ο παίχτης και τραβάει **μόνο ένα** χαρτί που προστίθεται στο χέρι του. Αν συνεχίσει, τραβάει χαρτιά μέχρι να αποφασίσει να σταματήσει. Το τελικό σκορ του παίχτη είναι το άθροισμα των πόντων των χαρτιών του. Αν ξεπεράσει τους 21 πόντους χάνει αυτόματα, ενώ αν πιάσει ακριβώς 21 πόντους κερδίζει αυτόματα. Στην περίπτωση που πιάσει 21 με ένα άσσο και ένα χαρτί αξίας 10 πόντων κερδίζει 1.5 φορά την αξία του στοιχήματος του (bet).

Αν ο παίχτης δεν κερδίσει ή δεν χάσει αυτόματα τότε παίζει ο dealer. Ο dealer τραβάει χαρτιά όσο το άθροισμα των πόντων του είναι λιγότερο από 17. Αν ξεπεράσει τα 21 καίγεται και πληρώνει το στοίχημα. Αλλιώς κερδίζει αυτός που έχει το μεγαλύτερο άθροισμα πόντων, και ο νικητής πληρώνεται όσο είναι το στοίχημα. Αν έχουμε ισοπαλία στους πόντους δεν κερδίζει κανένας.

Υλοποίηση

Για την υλοποίηση σας θα ακολουθήσετε τα εξής βήματα.

1. Πρώτα θα προγραμματίσετε μία κλάση **River** η οποία θα υλοποιεί το ρεύμα των χαρτιών. Η κλάση αυτή θα αρχικοποιείται με τον αριθμό από τράπουλες που έχει το ρεύμα. Αν δεν προσδιορίζεται (constructor χωρίς ορίσματα) τότε θα είναι μία μόνο τράπουλα. Η βασική λειτουργία της κλάσης θα είναι να μας δίνει το επόμενο τυχαίο χαρτί από τα εναπομείναντα χαρτιά. Αυτό θα γίνεται με τη μέθοδο **nextCard**. Μας ενδιαφέρει μόνο η αξία του χαρτιού, άρα η μέθοδος θα επιστρέφει ένα αριθμό από το 1 έως το 10.

Για την τυχαία επιλογή θα κάνετε την εξής υλοποίηση. Η κλάση σας θα έχει ένα πίνακα με 10 θέσεις, ο οποίος στη θέση j κρατάει τον αριθμό των εναπομεινάντων χαρτιών που έχουν τιμή **μικρότερη ή ίση** του j+1. Ο πίνακας αυτός λέγεται πίνακας μερικών αθροισμάτων (partial sums array). Αν έχουμε N εναπομείναντα χαρτιά, δημιουργούμε ένα τυχαίο ακέραιο x από το 1 έως το N και βρίσκουμε τη μεγαλύτερη θέση στον πίνακα με τιμή (περιεχόμενο) μικρότερη ή ίση του x. Αν αυτή είναι η θέση j (όπου η πρώτη θέση είναι η θέση μηδέν) επιστρέφουμε την τιμή j+1. Μπορείτε να δείτε ότι ο αριθμός των τιμών του x για τις οποίες θα επιστρέψουμε την τιμή j+1 είναι ίσος με τον αριθμό των χαρτιών με τιμή j+1, αρά διαλέγουμε με πιθανότητα ανάλογη της συχνότητας εμφάνισης κάθε τιμής.

Εκτός από τους constructors και την nextCard, θα έχουμε και μια συνάρτηση **start**, η οποία θα αρχικοποιεί τον αριθμό των εναπομεινάντων χαρτιών και τον πίνακα με τα μερικά αθροίσματα (ουσιαστικά θα είναι σαν να ξανά-ανακατεύουμε τα χαρτιά).

2. Θα προγραμματίσετε μια κλάση **Hand** η οποία θα κρατάει πληροφορία για ένα χέρι. Συγκεκριμένα θα κρατάει και θα ενημερώνει τους πόντους που έχει το χέρι, και θα γνωρίζει αν στο χέρι υπάρχει άσσος. Θα πρέπει να υπάρχει μια μέθοδος **getPoints**, η οποία θα επιστρέφει τους πόντους του χεριού. Στην μέθοδο αυτή θα χειριστείτε την περίπτωση που το χέρι έχει ένα άσσο. Στην περίπτωση αυτή αν το άθροισμα των τιμών των χαρτιών είναι P, και $P+10 \leq 21$, τότε οι πραγματικοί πόντοι του χεριού είναι P+10, γιατί ο άσσος αποτιμάται για 11 πόντους. Η getPoints μπορεί επίσης να τυπώνει τους πόντους του παίχτη, δίνοντας και τις δύο εναλλακτικές αποτιμήσεις στην περίπτωση που υπάρχει άσσος. Προσέξτε ότι δεν έχει σημασία αν το χέρι έχει πάνω από ένα άσσο εφόσον σε ένα χέρι το πολύ ένας άσσος να μετρήσει για 11 πόντους.
3. Θα προγραμματίσετε μία κλάση **Player** η οποία θα κρατάει πληροφορία για ένα παίχτη. Η κλάση θα πρέπει να κρατάει πόσα είναι τα κέρδη/ζημιές του παίχτη, καθώς και ποιο είναι το ποσό που έχει ποντάρει. Θα έχει μια μέθοδο **placeBet** η οποία θα ζητάει από τον χρήστη να καθορίσει πόσα θέλει να ποντάρει, και μέθοδο με την οποία ο παίχτης θα πληρώνει ή θα πληρώνεται ανάλογα με το αν κέρδισε ή έχασε. Θα έχει μια μέθοδο **play**, η οποία θα υλοποιεί το παίξιμο του παίχτη. Η μέθοδος αυτή θα παίρνει ως όρισμα ένα αντικείμενο της κλάσης River που ορίσατε παραπάνω ώστε να μπορεί να τραβάει χαρτιά, και θα δημιουργεί

ένα αντικείμενο της κλάσης Hand με το οποίο θα κρατάει τους πόντους του χεριού. Θα ζητάει από τον παίκτη να πάρει αποφάσεις αν θα κάνει double, αν θα τραβήξει, ή αν θα σταματήσει, και θα υλοποιεί τις αντίστοιχες ενέργειες. Όταν τραβάει χαρτιά και θα εκτυπώνει τι τράβηξε και πόσους πόντους έχει το χέρι. Θα σταματάει όταν το δηλώσει ο παίκτης, ή αυτόματα αν ο παίκτης καεί ή πιάσει 21. Θα επιστρέφει ένα ακέραιο με το τελικό σύνολο των πόντων του παίκτη.

4. Θα προγραμματίσετε την κλάση **BlackJack** η οποία θα περιέχει την μέθοδο main. Εκτός από τη main, η BlackJack θα έχει και άλλη μια μέθοδο **playDealer**, η οποία θα υλοποιεί το παίξιμο του dealer. Η υλοποίηση είναι παρόμοια με το παίξιμο του παίκτη αλλά ο αλγόριθμος του dealer είναι αυτός που περιγράψαμε στην αρχή. Στην **main** θα δημιουργήσετε ένα αντικείμενο της κλάσης River, και ένα αντικείμενο της κλάσης παίκτη Player. Ο παίκτης θα συνεχίσει να παίζει μέχρι να υποδείξει ότι θέλει να σταματήσει. Σε κάθε γύρο θα τυπώνονται οι πόντοι του χεριού του παίκτη και του dealer (αν παίξει) και θα μαθαίνουμε ποιος κέρδισε. Θα τυπώνονται επίσης τα κέρδη/ζημιές του παίκτη. Θα πρέπει να έχετε και ένα έλεγχο αν ο αριθμός των εναπομεινάντων χαρτιών στο ρεύμα των χαρτιών είναι πολύ μικρός, οπότε θα καλείτε την μέθοδο start που ξανά-ανακατεύει τα χαρτιά και ξεκινάει από την αρχή το ρεύμα. Αυτό μπορεί να υλοποιηθεί στη main, ή μέσα στη River.

Υποδείξεις

- Για τη δημιουργία τυχαίων αριθμών κοιτάξτε την συνάρτηση Random της Java. Παράδειγμα της χρήσης της έχουμε στο μάθημα 7.
- Σε κάθε κλάση μπορείτε να υλοποιήσετε και όποιες άλλες (public ή private) μεθόδους πιστεύετε ότι χρειάζονται, πέραν από αυτές που αναφέρονται στην εκφώνηση.
- Αν ορίσετε στην κλάση BlackJack μια μέθοδο που καλείται στην main θα πρέπει να οριστεί κι αυτή static.

ΓΕΝΙΚΕΣ ΟΔΗΓΙΕΣ

Κάντε turnin τα προγράμματα σας στο assignment1@ply212.

π.χ. turnin assignment1@ply212 BlackJack.java

Στον κώδικα να αναγράφονται σε σχόλια το όνομα το login και ο ΑΜ σας.