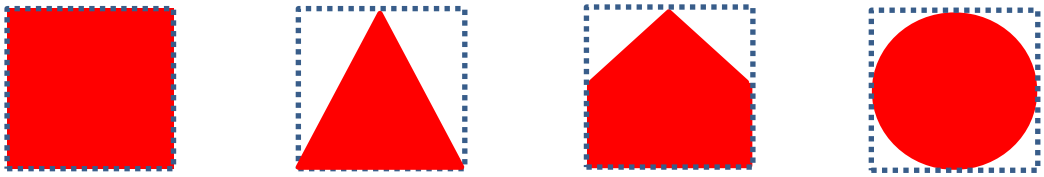
Τρίτη Σειρά ασκήσεων
Ημερομηνία Παράδοσης: 27 Μαΐου 2016, 6 μ.μ.

Οι παρακάτω ασκήσεις πρέπει να παραδοθούν μέχρι τις 27/5/2016, 6 μ.μ. . Στην υλοποίηση των κλάσεων σας δεν θα πρέπει να έχετε public πεδία. 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 ένα απλό παιχνίδι με σχήματα που μοιάζει (λίγο) με το Tetris. Έχουμε μια γεννήτρια που παράγει σχήματα επιλεγμένα τυχαία από τετράγωνο, τρίγωνο, πεντάγωνο και κύκλο (όπως φαίνονται στο σχήμα παρακάτω). Ο συνολικός αριθμός των σχημάτων που μπορεί να παράγει η γεννήτρια είναι προκαθορισμένος, και τα σχήματα παράγονται με τυχαία επιλεγμένο μέγεθος. Έχουμε ένα παίχτη ο οποίος μαζεύει τα σχήματα σε μία στοίβα περιορισμένου μεγέθους. Για κάθε νέο σχήμα που παράγεται, ο παίχτης πρέπει να επιλέξει αν θα κρατήσει αυτό το σχήμα ή όχι. Αν το κρατήσει, το σχήμα μπαίνει στην κορυφή της στοίβας. Για κάθε σχήμα που κρατάει, ο παίχτης παίρνει ένα αριθμό πόντων ίσο με το εμβαδόν του σχήματος. Αν το καινούριο σχήμα που μαζεύει ο παίχτης έχει το ίδιο εμβαδόν με το προηγούμενο στην κορυφή της στοίβας τότε οι πόντοι δεκαπλασιάζονται. Επίσης, αν το νέο σχήμα είναι ίδιου τύπου με αυτό στην κορυφή της στοίβας τότε τα δύο αυτά σχήματα αφαιρούνται από την στοίβα (ο παίχτης κρατάει τους πόντους). Το παιχνίδι τελειώνει όταν είτε τελειώσουν τα σχήματα της γεννήτριας, είτε γεμίσει η στοίβα του παίχτη. Ο στόχος του παίχτη είναι να συγκεντρώσει όσο περισσότερους πόντους γίνεται.



Για την υλοποίηση σας θα δημιουργήσετε τις παρακάτω κλάσεις:

Μια **αφηρημένη** κλάση **Shape** η οποία κρατάει πληροφορίες για ένα γενικό σχήμα. Έχει δύο πεδία: ένα String με τον τύπο του σχήματος, και ένα int με το εμβαδόν του τετραγώνου που περικλείει το σχήμα (περιβάλλον τετράγωνο). Το περιβάλλον τετράγωνο είναι αυτό που φαίνεται με διακεκομμένες γραμμές στο παραπάνω σχήμα. Τα πεδία δίνονται ως ορίσματα στον constructor. Ορίστε την **αφηρημένη** μέθοδο **computeArea** που επιστρέφει το εμβαδόν του σχήματος. Επίσης τις παρακάτω (ενυπόστατες) μεθόδους: **sameArea** συγκρίνει με ένα άλλο σχήμα αν έχουν το ίδιο εμβαδόν; **sameType** ομοίως συγκρίνει τον τύπο; **toString** επιστρέφει τύπο και εμβαδόν του σχήματος; Όποια μέθοδο πρόσβασης χρειαστείτε.

Τέσσερεις ενυπόστατες κλάσεις **Square**, **Triangle**, **Pentagon**, **Circle** οι οποίες κληρονομούν από την κλάση Shape. Ο constructor παίρνει σαν όρισμα μόνο το εμβαδόν του περιβάλλοντος τετραγώνου. Αν Β είναι το εμβαδόν του περιβάλλοντος τετραγώνου, το εμβαδόν του τετραγώνου είναι (επίσης) Β, του τριγώνου ½ Β, του πενταγώνου ¾ Β και του κύκλου $\frac{\pi}{4}$Β.

Μια κλάση **ShapeGenerator** η οποία υλοποιεί την γεννήτρια που παράγει σχήματα. Ο constructor της ShapeGenerator παίρνει σαν όρισμα τον αριθμό των σχημάτων που θα δημιουργήσει η γεννήτρια. Η κλάση έχει δύο βασικές μεθόδους: την **hasNextShape** που επιστρέφει false ή true αν έχουν τελειώσει ή όχι τα σχήματα της γεννήτριας, την **nextShape** η οποία επιστρέφει ένα τυχαίο σχήμα. Το μέγεθος του σχήματος (το εμβαδόν του περιβάλλοντος τετραγώνου) επιλέγεται τυχαία από ένα πίνακα με τα πιθανά μεγέθη. Για την υλοποίηση σας βάλετε τις τιμές {1,2,3,4,8,12,16} στον πίνακα.

Την κλάση **Player** η οποία υλοποιεί τον παίχτη. Ο constructor παίρνει σαν όρισμα το μέγεθος της στοίβας. Η κλάση θα κρατάει οπωσδήποτε την στοίβα με τα σχήματα και τους πόντους του παίχτη, και όποιο άλλο πεδίο χρειάζεστε.

Μπορείτε να υλοποιήσετε την στοίβα όπως θέλετε (π.χ., με πίνακα, ή με την υλοποίηση ArrayDeque, ή LinkedList της Java). Η κλάση θα έχει την βασική μέθοδο **play** η οποία παίρνει σαν όρισμα ένα σχήμα και υλοποιεί το παιχνίδι του παίχτη για το σχήμα, την μέθοδο **isStackFull** που ελέγχει αν γέμισε η στοίβα του παίχτη, και την **printInfo** που εκτυπώνει την στοίβα και τους πόντους του παίχτη. Ορίστε όποια άλλη μέθοδο χρειάζεστε.

Τέλος, θα υλοποιήσετε την κλάση **ShapeGame** η οποία θα έχει την main που υλοποιεί το παιχνίδι.

Υπόδειξη: Χρησιμοποιήστε το Math.PI για την τιμή του π.

Άσκηση 2

Σε μια υπηρεσία όπως το Foursquare έχουμε ένα αρχείο το οποίο περιέχει ζευγάρια από χρήστες και επιχειρήσεις, για όλες τις επισκέψεις που κάνουν οι χρήστες στις επιχειρήσεις. Κάθε γραμμή του αρχείου έχει δύο IDs χωρισμένα με κενό. Το πρώτο είναι το user ID ενός χρήστη και το δεύτερο το business ID μιας επιχείρησης. Η γραμμή στο αρχείο σημαίνει ότι ο χρήστης με αυτό το ID επισκέφτηκε την επιχείρηση με το συγκεκριμένο ID.

Από το αρχείο που μας δίνεται θέλουμε να κρατήσουμε ένα υποσύνολο S από user IDs και business IDs, έτσι ώστε, για κάποια παράμετρο K, ο κάθε χρήστης που είναι στο υποσύνολο S έχει επισκεφτεί τουλάχιστον K από τις επιχειρήσεις που είναι στο υποσύνολο S, και αντίστοιχα η κάθε επιχείρηση που είναι στο S την έχουν επισκεφτεί τουλάχιστον K χρήστες που είναι στο S. Θα δημιουργήσουμε ένα νέο αρχείο με όλα τα ζευγάρια από το αρχικό αρχείο που έχουν user ID **και** business ID που ανήκουν στο S.

Υλοποιήστε ένα πρόγραμμα που κάνει αυτό το φιλτράρισμα των ζευγαριών. Το πρόγραμμα σας θα πρέπει να έχει δύο κλάσεις:

Την κλάση **FilterPairs** η οποία κάνει όλη την δουλειά. Ο constructor παίρνει σαν όρισμα την τιμή του threshold K. Επειδή το αρχείο εισόδου μπορεί να είναι πολύ μεγάλο, θα πρέπει να χρησιμοποιήσουμε τις κατάλληλες δομές. Η κλάση θα κρατάει δύο **HashMap**: Το πρώτο έχει σαν κλειδιά τα user IDs και τιμές **HashSet** με τα business IDs που έχει επισκεφτεί ο χρήστης. Το δεύτερο έχει σαν κλειδιά τα business IDs και τιμές HashSet με τα user IDs που έχουν επισκεφτεί την επιχείρηση. Η κλάση έχει τις εξής μεθόδους:

- Την μέθοδο **readFile** η οποία παίρνει σαν είσοδο το όνομα αρχείου της εισόδου, το διαβάζει, και δημιουργεί τα δύο HashMap. Η μέθοδος αυτή **δεν χειρίζεται τις εξαιρέσεις**.
- Την μέθοδο **filter** η οποία κάνει το φιλτράρισμα και αφαιρεί (επαναληπτικά) από τα HashMaps τα IDs που δεν πληρούν την συνθήκη μας.
- Την μέθοδο **printFile** η οποία παίρνει σαν είσοδο το όνομα αρχείου της εξόδου, και τυπώνει τα ζευγάρια από χρήστες και επιχειρήσεις με IDs που πληρούν τις συνθήκες. Η μέθοδος αυτή **δεν χειρίζεται τις εξαιρέσεις**.

Την κλάση **RunFilter** η οποία έχει μόνο την μέθοδο main. Τα ονόματα των αρχείων εισόδου και εξόδου, και το K δίνονται σαν ορίσματα εκτέλεσης. Άρα για παράδειγμα για το αρχείο εισόδου input.txt, αρχείο εξόδου output.txt, και κατώφλι 3 τρέχουμε το πρόγραμμα ως εξής:

java RunFilter input.txt output.txt 3 .

Η κλάση main είναι υπεύθυνη για να χειριστεί τις εξαιρέσεις.

Στο turnin παραδώστε τον κώδικα σας και το μικρό αρχείο εισόδου που σας δόθηκε. Το μεγαλύτερο αρχείο είναι για να τεστάρετε την ταχύτητα του προγράμματος σας.

Bonus Άσκηση

Στην προηγούμενη σειρά ασκήσεων υλοποιήσατε την κλάση BinarySearchTree. Τροποποιήστε την υλοποίηση σας ώστε να ορίσετε μια γενικευμένη κλάση BinarySearchTree που να δουλεύει με αντικείμενα που είναι Comparable. Δημιουργήσετε ένα παράδειγμα με μία κλάση που φτιάξετε εσείς και θα την περάσετε ως παράμετρο, για να δείξετε ότι η κλάση σας δουλεύει σωστά. Εξηγήστε τις επιλογές σας στα σχόλια του κώδικα.

Η άσκηση θα βαθμολογηθεί με βάση το κομμάτι που αφορά την γενικευμένη κλάση, όχι την αρχική υλοποίηση.