ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
& ΠΛΗΡΟΦΟΡΙΚΗΣ

Πρώτη Σειρά ασκήσεων
Ημερομηνία Παράδοσης: Τρίτη 10 Μαΐου 13:00

Για την άσκηση αυτή θα προγραμματίσετε μια απλοποιημένη εκδοχή του παιχνιδιού Αγωνία (στα αγγλικά Crazy Eights) όπου θα μπορείτε να παίζετε με τον υπολογιστή.

Κανόνες του παιχνιδιού

Το παιχνίδι παίζεται με μία τράπουλα. Η τράπουλα ανακατεύεται, μοιράζονται από 5 χαρτιά σε κάθε παίχτη και ανοίγεται ένα χαρτί στο τραπέζι. Οι παίχτες παίζουν ο ένας μετά τον άλλο, και ο κάθε παίχτης στη σειρά του, αν έχει ένα χαρτί που συμφωνεί είτε στο χρώμα (suit) είτε στο νούμερο (number) με το τελευταίο χαρτί που ρίχτηκε στο τραπέζι, το πετάει. Αν έχει ένα χαρτί με το νούμερο 8, μπορεί να το ρίξει ανά πάσα στιγμή, και να καθορίσει αυτός το χρώμα του (το οποίο μπορεί να είναι διαφορετικό από αυτό που έχει το χαρτί). Αν δεν έχει κάποιο χαρτί που να μπορεί να ρίξει τότε τραβάει ένα χαρτί από την τράπουλα και χάνει τη σειρά του. Αν τελειώσει η στοίβα με τα χαρτιά που τραβάνε οι παίχτες, τότε βάζουμε τα χαρτιά που έχουν πεταχτεί πίσω στην τράπουλα και τα ανακατεύουμε. Όποιος παίχτης μείνει πρώτος χωρίς χαρτιά κερδίζει το παιχνίδι.

Υλοποίηση

Για την υλοποίηση σας θα δημιουργήσετε τις παρακάτω κλάσεις:

Card

Η κλάση Card κρατάει πληροφορίες για ένα χαρτί. Για απλότητα θα θεωρήσουμε ότι τα χαρτιά έχουν νούμερα 0 έως 9. Το χρώμα και το νούμερο του χαρτιού αρχικοποιούνται στον constructor. Η κλάση θα έχει επίσης μεθόδους πρόσβασης (**accessor**) και μέθοδο μετάλλαξης (**mutator**) για το χρώμα, και τη μέθοδο **toString** που επιστρέφει ένα String όπως φαίνεται στα παραδείγματα. Θα έχετε επίσης μια μέθοδο **isEight** που επιστρέφει μια Boolean τιμή αν το φύλλο είναι 8, και την μέθοδο **matches** που παίρνει ένα άλλο χαρτί και επιστρέφει true αν τα δύο χαρτιά συμφωνούν στο χρώμα ή στο νούμερο.

Pile

Η κλάση Pile κρατάει πληροφορίες για την στοίβα (τράπουλα) των χαρτιών από την οποία τραβάνε χαρτιά οι παίχτες. Έχει μια δομή για να κρατάει τα χαρτιά (αντικείμενα Card), και υποστηρίζει τις εξής μεθόδους:

- **fill**: «Γεμίζει» την στοίβα. Δημιουργεί τα 40 χαρτιά της τράπουλας, τα προσθέτει στην δομή που κρατάει τα χαρτιά και ανακατεύει την τράπουλα.
- Υπερφορτώστε την μέθοδο **fill** ώστε να παίρνει σαν όρισμα ένα ArrayList με χαρτιά. Η μέθοδος γεμίζει την στοίβα με τα χαρτιά στη λίστα, και τα ανακατεύει.
- **shuffle**: Μια ιδιωτική (private) μέθοδος που ανακατεύει την τράπουλα δημιουργώντας μια τυχαία αναδιάταξη. Μπορείτε να το υλοποιήσετε αυτό διαλέγοντας τυχαία το τελευταίο, το προτελευταίο, κλπ.
- **nextCard**: Αφαιρεί το χαρτί από την κορυφή της στοίβας και το επιστρέφει.
- **isEmpty**: Επιστρέφει true αν η στοίβα έχει αδειάσει.
- **print**: Εκτυπώνει τα χαρτιά στη στοίβα (σε μία γραμμή).
- **main**: Δημιουργήστε μια main για να τεστάρετε την κλάση σας. Στη main θα δημιουργήσετε μια στοίβα και θα τη γεμίσετε με όλη την τράπουλα, και μία άλλη που θα την γεμίσετε με μια λίστα με 4 χαρτιά. Εκτυπώστε τις στοίβες και καλέσετε την nextCard για να βεβαιωθείτε ότι λειτουργεί σωστά. Μπορείτε να κάνετε και άλλα τεστ που θέλετε.

Table

Η κλάση Table κρατάει πληροφορίες για το τραπέζι. Έχει πεδία μια στοίβα (Pile) από την οποία τραβάνε χαρτιά οι παίχτες, μία λίστα με τα χαρτιά που έχουν πεταχτεί, και το τελευταίο χαρτί (topCard) που έχει πεταχτεί. Έχει και τις εξής μεθόδους:

- Ο **constructor** ο οποίος γεμίζει την στοίβα με όλη την τράπουλα και ανοίγει το πρώτο χαρτί στο τραπέζι.
- **throwCard**: Παίρνει σαν όρισμα ένα χαρτί (αντικείμενο Card) που κάποιος πετάει στο τραπέζι και ενημερώνει κατάλληλα τα πεδία του αντικειμένου.
- **drawCard**: Επιστρέφει το επόμενο χαρτί (αντικείμενο Card) από τη στοίβα και χειρίζεται την περίπτωση που άδειασε η στοίβα.
- **getTopCard**: Μέθοδος πρόσβασης (accessor method) για το χαρτί στην κορυφή του τραπεζιού.
- **main**: Δημιουργήστε μια main όπου θα δημιουργήσετε ένα τραπέζι, θα καλέσετε την drawCard για όλα τα χαρτιά και θα πετάτε (throwCard) κάθε δεύτερο. Τεστάρετε και την getTopCard και αν χειρίζεστε σωστά την περίπτωση που αδειάσει η στοίβα.

Hand

Η κλάση Hand κρατάει πληροφορίες για το χέρι του παίχτη. Έχει μια λίστα με τα χαρτιά που έχει στο χέρι του ο παίχτης, και ένα HashMap που για κάθε χρώμα κρατάει το πλήθος των χαρτιών με αυτό το χρώμα (δεν μετράνε χαρτιά με αριθμό 8, εφόσον ουσιαστικά δεν έχουν χρώμα). Έχει τις εξής μεθόδους:

- Ο **constructor** αρχικοποιεί το HashMap ώστε όλοι οι μετρητές να είναι μηδέν.
- **addCard**: Παίρνει σαν όρισμα ένα χαρτί και το προσθέτει στο χέρι, ενημερώνοντας κατάλληλα όλα τα πεδία.
- **removeCard**: Παίρνει σαν όρισμα ένα χαρτί και το αφαιρεί από το χέρι, ενημερώνοντας κατάλληλα όλα τα πεδία.
- **getCard**: Παίρνει σαν όρισμα μια θέση στη λίστα και επιστρέφει το χαρτί σε εκείνη τη θέση
- **printHand**: Τυπώνει τη λίστα με τα χαρτιά μαζί με τη θέση τους. Παραδείγματα εξόδου δίνονται παρακάτω.
- **isEmpty**: Επιστρέφει true αν έχει αδειάσει η λίστα με τα χαρτιά.
- **findMatchingCard**: Η μέθοδος αυτή υλοποιεί την στρατηγική του παίχτη-υπολογιστή για το πως διαλέγει χαρτιά να πετάξει. Παίρνει σαν όρισμα ένα χαρτί (αντικείμενο Card). Αυτό είναι το τρέχον χαρτί που είναι στο τραπέζι. Επιστρέφει ένα χαρτί (αντικείμενο Card) που ταιριάζει με αυτό το χαρτί. Αυτό είναι το χαρτί που θα πετάξει ο παίχτης-υπολογιστής. Οποιοδήποτε χαρτί ταιριάζει με το χαρτί στο τραπέζι (π.χ., το πρώτο που θα βρείτε) είναι αποδεκτό (στο bonus προτείνεται μια πιο έξυπνη τακτική). Αν δεν υπάρχει χαρτί που ταιριάζει, και υπάρχει 8άρι στο χέρι, τότε επιστρέφει το 8άρι και θέτει το χρώμα να είναι το χρώμα με τα περισσότερα χαρτιά στο χέρι. Αν βρεθεί ένα χαρτί που συμφωνεί με το χαρτί στο τραπέζι η μέθοδος το επιστρέφει, αλλά δεν το αφαιρεί από το χέρι. Αλλιώς επιστρέφει null.
  Για να βρείτε το χρώμα με τα περισσότερα χαρτιά μπορείτε να υλοποιήσετε μια βοηθητική μέθοδο.
- **main**: Δημιουργήστε μια main για να τεστάρετε την κλάση σας. Δημιουργήσετε χαρτιά, προσθέστε και αφαιρέστε τα, εκτυπώστε, και δοκιμάστε αν η findMatchingCard δουλεύει σωστά (ειδικά στην περίπτωση που πρέπει να επιστρέψει 8). Μπορείτε να δημιουργήσετε επιπλέον μέθοδο για να εκτυπώσετε το HashMap.

Player

Η κλάση αυτή κρατάει πληροφορίες για ένα παίχτη, ο οποίος μπορεί να είναι είτε παίχτης-άνθρωπος, είτε παίχτης-υπολογιστής. Έχει πεδία ένα String με το όνομα του παίχτη, και το χέρι (Hand) που κρατάει ο παίχτης. Το όνομα αρχικοποιείται στον constructor. Έχει και τις εξής μεθόδους:

- **draw**: Παίρνει σαν όρισμα το τραπέζι (Table) και τραβάει ένα χαρτί από το τραπέζι, και το προσθέτει στο χέρι του παίχτη.
- Υπερφορτώστε την **draw** ώστε να παίρνει ως όρισμα το τραπέζι (Table) και ένα ακέραιο αριθμό και να τραβάει τόσα χαρτιά από το τραπέζι και να τα προσθέτει στο χέρι του παίχτη.
- **throwCard**: Παίρνει σαν όρισμα το τραπέζι (Table) και ένα χαρτί (Card), το ρίχνει στο τραπέζι και το αφαιρεί από το χέρι του παίχτη.

- **isDone**: Επιστρέφει true αν έχει αδειάσει το χέρι του παίχτη.
- **toString**: Επιστρέφει ένα String με το όνομα του παίχτη.
- **selectCard**: Η μέθοδος αυτή υλοποιεί την επιλογή του χαρτιού από τον παίχτη-άνθρωπο. Παίρνει σαν όρισμα το τραπέζι, και επιστρέφει το χαρτί που επέλεξε ο παίχτης. Εκτυπώστε το χέρι και ζητήστε από τον παίχτη να δώσει από την είσοδο το χαρτί (την θέση του) που θέλει να πετάξει, ή -1 αν θέλει να πασάρει. Αν ο παίχτης επιλέξει -1, επιστρέψτε null. Αλλιώς, αν το χαρτί που επέλεξε ο παίχτης είναι 8άρι, ζητήστε από την είσοδο να προσδιορίσει το χρώμα του χαρτιού, και θέστε το χρώμα του φύλλου ανάλογα, και επιστρέψτε το φύλλο. Αν δεν είναι 8άρι, εξετάστε αν το φύλλο που επέλεξε ο χρήστης ταιριάζει με το χαρτί στο τραπέζι. Αν ναι επιστρέψετε το χαρτί, αλλιώς επαναλάβετε την διαδικασία μέχρι να πάρετε μια αποδεκτή απάντηση.
- **computerSelectCard**: Η μέθοδος αυτή υλοποιεί την επιλογή του χαρτιού από τον παίχτη-υπολογιστή. Παίρνει σαν όρισμα το τραπέζι, και επιστρέφει το χαρτί που επέλεξε ο παίχτης. Το χαρτί επιλέγεται χρησιμοποιώντας την findMatchingCard της κλάσης Hand.
- **main**: Δημιουργήστε μια μέθοδο main για να τεστάρετε την κλάση σας. Τεστάρετε όλες τις μεθόδους και τις οριακές καταστάσεις.

CrazyEights

Η κλάση αυτή κρατάει πληροφορίες για ένα παιχνίδι Crazy Eights. Έχει πεδία ένα τραπέζι και ένα πίνακα με παίχτες, και τον αριθμό των παιχτών. Έχει τις εξής μεθόδους.

- Ένα **constructor** με όρισμα τον αριθμό των παιχτών, ο οποίος δημιουργεί τους παίχτες. Ο πρώτος παίχτης είναι παίχτης-άνθρωπος, και οι υπόλοιποι παίχτες υπολογιστές.
- **play:** Η μέθοδος που υλοποιεί το παιχνίδι. Οι παίχτες παίζουν με τη σειρά, διαλέγουν χαρτί και είτε πετάνε είτε τραβάνε. Όταν κάποιος παίχτης πετάξει όλα τα χαρτιά του το παιχνίδι σταματάει και ο παίχτης κερδίζει.

Game

Η κλάση αυτή έχει μόνο τη main. Μέσα στη main ζητάμε από τον χρήστη να δώσει τον αριθμό των παιχτών δημιουργούμε ένα αντικείμενο CrazyEights και καλούμε την μέθοδο play.

Bonus: Υλοποιήσετε την εξής τακτική για τον υπολογιστή στην findMatchingCard μέθοδο. Αρχικά για κάθε χρώμα υπολογίζει αν είναι υποψήφιο. Ένα χρώμα είναι υποψήφιο αν έχει μη μηδενικό αριθμό από χαρτιά και είτε συμφωνεί με το χρώμα του τραπεζιού, είτε έχει ένα χαρτί με νούμερο που συμφωνεί με το νούμερο του τραπεζιού. Από τα υποψήφια χρώματα ο υπολογιστής επιλέγει το χρώμα το οποίο έχει τα περισσότερα χαρτιά και παίζει το αντίστοιχο χαρτί.

Υποδείξεις

- Όταν θέλετε να διατρέξετε όλα τα χρώματα θα σας βοηθήσει να δημιουργήσετε ένα πίνακα που να τον αρχικοποιήσετε με όλα τα χρώματα.
- Κάποιες μέθοδοι μπορεί να επιστρέψουν την τιμή null. Θα πρέπει πριν χρησιμοποιήσετε το αντικείμενο που επιστρέφουν να βεβαιωθείτε ότι δεν είναι null, κάνοντας έλεγχο ισότητας (==) ή ανισότητας (!=). Ο έλεγχος αυτός είναι απαραίτητος γιατί η τιμή null σηματοδοτεί κάτι (π.χ., αδυναμία να βρεθεί χαρτί που ταιριάζει).
- Να είστε προσεκτικοί αν καλέστε την remove για την ArrayList, ενώ διατρέχετε τα αντικείμενα της λίστας.
- Μπορείτε να προσθέσετε επιπλέον εκτυπώσεις για να σας βοηθήσει να τεστάρετε τον κώδικα σας, αλλά στην τελική έκδοση του κώδικα η έξοδος θα πρέπει να είναι απλή και κατανοητή.
- Σε κάθε κλάση μπορείτε να υλοποιήσετε και άλλες (public ή private) μεθόδους πέραν από αυτές που αναφέρονται στην εκφώνηση, αλλά θα πρέπει να έχετε υλοποιήσει τις public μεθόδους που ζητάει η άσκηση ακριβώς όπως σας ζητούνται.

ΓΕΝΙΚΕΣ ΟΔΗΓΙΕΣ

- Μια κλάση που δεν κάνει compile μηδενίζεται αυτόματα.
- Δεν επιτρέπεται η χρήση public ή protected πεδίων στην άσκηση. Επίσης ο κώδικας θα πρέπει να είναι σωστά στοιχισμένος και καλά γραμμένος. Θα αφαιρεθούν βαθμοί από προγράμματα που είναι πολύ κακά γραμμένα.
- Θα τεστάρουμε και θα βαθμολογήσουμε την κάθε κλάση ξεχωριστά. Γι αυτό και θα πρέπει να σώσετε την κάθε κλάση σε ξεχωριστό αρχείο. Θα πρέπει επίσης να κρατήσετε τα ονόματα και τα ορίσματα των public μεθόδων ακριβώς όπως σας ζητούνται.
- Κάντε turnin τα προγράμματα σας στο assignment1@myy205.

  π.χ. turnin assignment1@myy205 MemoryGame.java

  Μπορείτε να κάνετε turnin πολλά αρχεία μαζί στην ίδια εντολή. Διαβάστε προσεκτικά τις οδηγίες για το trunin στο ecourse και βεβαιωθείτε ότι μπορείτε να κάνετε την διαδικασία κάποιες μέρες πριν την προθεσμία. Μπορείτε να κάνετε πολλαπλές φορές turnin τα ίδια αρχεία, θα κοιτάξουμε το τελευταίο. Κάθε φορά πρέπει να κάνετε turnin όλα τα αρχεία που θέλετε να παραδώσετε. Δεν μπορείτε να κάνετε turnin zip αρχείο, ή αρχείο με ελληνικούς χαρακτήρες.

  Στον κώδικα να αναγράφονται σε σχόλια το όνομα και ο ΑΜ σας (**με λατινικούς χαρακτήρες**).

Παραδείγματα Εξόδου:

Παρακάτω σας δίνονται μερικά παραδείγματα εξόδου. Δεν είναι ανάγκη η έξοδος σας να είναι ακριβώς έτσι αλλά πρέπει να είναι παρόμοια.

Παράδειγμα 1:
Το παιχνίδι είναι μεταξύ του παίχτη-ανθρώπου Human και του παίχτη-υπολογιστή Computer1. Εδώ εκτυπώνουμε και το χέρι του Computer1 ώστε να παρακολουθήσουμε καλύτερα το παιχνίδι. Το παράδειγμα δείχνει ένα στιγμιότυπο από το παιχνίδι.

```
Human turn:
Table top card: 6S
         0       1       2       3       4
        1H      1C      4C      3D      9H
Select a card to throw or -1 to pass
4
         0       1       2       3       4
        1H      1C      4C      3D      9H
Select a card to throw or -1 to pass
-1
Human drew a card

Computer1 turn:
Table top card: 6S
         0       1       2       3       4
        2H      3H      0C      4H      8C
Computer1 threw card 8H

Human turn:
Table top card: 8H
```

```
         0       1       2       3       4       5
        1H      1C      4C      3D      9H      0H
Select a card to throw or -1 to pass
4
Human threw card 9H

Computer1 turn:
Table top card: 9H
         0       1       2       3
        2H      3H      0C      4H
Computer1 threw card 2H
```

Παράδειγμα 2:
Το παιχνίδι είναι μεταξύ του παίχτη-ανθρώπου Human και των παίχτων-υπολογιστή Computer1 και Computer2. Η έξοδος είναι από την εξέλιξη ολόκληρου του παιχνιδιού.

```
>java Game
Give the number of players:
3

Human turn:
Table top card: 5D
         0       1       2       3       4
        3C      6S      0S      9D      7H
Select a card to throw or -1 to pass
3
Human threw card 9D

Computer1 turn:
Table top card: 9D
         0       1       2       3       4
        9H      6H      5H      5C      1C
Computer1 threw card 9H

Computer2 turn:
Table top card: 9H
         0       1       2       3       4
        9S      2C      8C      8D      0C
Computer2 threw card 9S

Human turn:
Table top card: 9S
         0       1       2       3
        3C      6S      0S      7H
Select a card to throw or -1 to pass
1
Human threw card 6S

Computer1 turn:
Table top card: 6S
         0       1       2       3
        6H      5H      5C      1C
Computer1 threw card 6H

Computer2 turn:
Table top card: 6H
         0       1       2       3
        2C      8C      8D      0C
Computer2 threw card 8C
```

```
Human turn:
Table top card: 8C
         0       1       2
        3C      0S      7H
Select a card to throw or -1 to pass
0
Human threw card 3C

Computer1 turn:
Table top card: 3C
         0       1       2
        5H      5C      1C
Computer1 threw card 5C

Computer2 turn:
Table top card: 5C
         0       1       2
        2C      8C      0C
Computer2 threw card 2C

Human turn:
Table top card: 2C
         0       1
        0S      7H
Select a card to throw or -1 to pass
-1
Human drew a card

Computer1 turn:
Table top card: 2C
         0       1
        5H      1C
Computer1 threw card 1C

Computer2 turn:
Table top card: 1C
         0       1
        8C      0C
Computer2 threw card 8C

Human turn:
Table top card: 8C
         0       1       2
        0S      7H      6D
Select a card to throw or -1 to pass
-1
Human drew a card

Computer1 turn:
Table top card: 8C
         0
        5H
Computer1 drew a card

Computer2 turn:
Table top card: 8C
         0
        0C
Computer2 threw card 0C

Player Computer2 won
```

Παράδειγμα 3:
Το παιχνίδι είναι μεταξύ του παίχτη-ανθρώπου Human και του παίχτη-υπολογιστή Computer1. Εδώ δεν εκτυπώνουμε το χέρι του υπολογιστή.

```
>java Game
Give the number of players:
2

Human turn:
Table top card: 1H
         0       1       2       3       4
        3S      5C      6D      8C      1D
Select a card to throw or -1 to pass
4
Human threw card 1D

Computer1 turn:
Table top card: 1D
Computer1 threw card 7D

Human turn:
Table top card: 7D
         0       1       2       3
        3S      5C      6D      8C
Select a card to throw or -1 to pass
2
Human threw card 6D

Computer1 turn:
Table top card: 6D
Computer1 threw card 8S

Human turn:
Table top card: 8S
         0       1       2
        3S      5C      8C
Select a card to throw or -1 to pass
0
Human threw card 3S

Computer1 turn:
Table top card: 3S
Computer1 threw card 8S

Human turn:
Table top card: 8S
         0       1
        5C      8C
Select a card to throw or -1 to pass
1
Select a suit
C
Human threw card 8C

Computer1 turn:
Table top card: 8C
Computer1 drew a card

Human turn:
```

```
Table top card: 8C
         0
        5C
Select a card to throw or -1 to pass
0
Human threw card 5C

Player Human won
```