

**Πρώτη Σειρά ασκήσεων**  
**Ημερομηνία Παράδοσης: Τετάρτη 15 Μαΐου 10:59 π.μ. (πριν το μάθημα)**

Για την άσκηση αυτή θα υλοποιήσετε σε Java δύο απλά παιχνίδια με χαρτιά. Η υλοποίησή σας θα γίνει σε βήματα.

**Βήμα 1.**

Στο πρώτο βήμα θα δημιουργήσετε δύο κλάσεις που είναι κοινές και στα δύο παιχνίδια. Την κλάση **Card** για ένα φύλλο της τράπουλας και την κλάση **CardStack** για μία στοίβα από χαρτιά.

Η κλάση **Card** έχει δύο πεδία, το χρώμα (suit) του χαρτιού και το rank (νούμερο/φιγούρα) του χαρτιού. Το πρώτο θα είναι String με τιμές "S" (Spade: Μπαστούνι), "H" (Heart: Κούπα), "D" (Diamond: Καρρό) και "C" (Club: Σπαθί). Το δεύτερο θα είναι ένας ακέραιος, όπου τα νούμερα τα κρατάτε όπως είναι, ενώ Βαλές (Jack), Ντάμα (Queen), Ρήγας (King) και Άσος αντιστοιχούν στα νούμερα 11, 12, 13, και 14.

Η κλάση θα έχει ένα constructor που θα αρχικοποιεί τα δύο πεδία και τις μεθόδους **toString** και **equals**. Η **toString** θα επιστρέφει την «κανονική» αναπαράσταση του χαρτιού. Έτσι, π.χ., το χαρτί Άσος Μπαστούνι, θα αναπαρίσταται με το String "AS" και όχι ως 14S όπως αποθηκεύεται. Η **equals** θα ελέγχει μόνο αν το rank είναι ίδιο. Για παράδειγμα ο βαλές καρό και ο βαλές μπαστούνι θα είναι ίδια. Επιπλέον μέθοδοι θα προστεθούν στα επόμενα βήματα.

Η κλάση **CardStack** θα υλοποιεί μια στοίβα με χαρτιά η οποία αποτελείται από μία ή περισσότερες τράπουλες. Η κλάση θα έχει ένα πεδίο πίνακα (όχι ArrayList) με χαρτιά (Card). Χρειάζεται επίσης να ξέρει σε ποια θέση είναι η κορυφή της στοίβας, τον τρέχοντα αριθμό χαρτιών, και τον συνολικό αριθμό χαρτιών. Η υλοποίηση είναι πολύ παρόμοια με την υλοποίηση της δομής Stack με πίνακα που κάναμε στην τάξη, και μπορείτε να την χρησιμοποιήσετε ως οδηγό.

Η κλάση θα έχει τις εξής μεθόδους:

- Ένα **constructor** χωρίς ορίσματα ο οποίος δημιουργεί μια στοίβα με μία τράπουλα. Ο constructor θα δημιουργήσει και τα 52 χαρτιά της τράπουλας και θα τα προσθέσει στην στοίβα. Συνίσταται να δημιουργήσετε βοηθητικές μεθόδους που δημιουργούν τα χαρτιά μιας τράπουλας και ίσως και τα χαρτιά ενός χρώματος (suit).
- Ένα **constructor** με όρισμα τον αριθμό από τράπουλες που θα έχει η στοίβα ο οποίος δημιουργεί πάλι την στοίβα με αυτό τον αριθμό από τράπουλες και τα αντίστοιχα χαρτιά.
- Τη μέθοδο **shuffle** η οποία ανακατεύει τα χαρτιά της στοίβας. Για να ανακατέψετε τα χαρτιά, μπορείτε να επιλέξετε τυχαία ένα χαρτί για την πρώτη θέση, μετά από τα υπόλοιπα ένα χαρτί για την δεύτερη θέση, κλπ.
- Τη μέθοδο **draw** η οποία «αφαιρεί» και επιστρέφει το πρώτο χαρτί από την κορυφή της στοίβας.
- Τη μέθοδο **isEmpty** που επιστρέφει μια Boolean τιμή αν έχει αδειάσει η στοίβα.
- Τη μέθοδο **isHalfEmpty** που επιστρέφει μια Boolean τιμή αν η στοίβα έχει λιγότερα από τα μισά χαρτιά που είχε αρχικά.
- Τη μέθοδο **toString** που επιστρέφει ένα String με τα χαρτιά που έχει η στοίβα σε μια γραμμή.

Δημιουργείστε μια **main** στην κλάση CardStack για να τεστάρετε τη μέθοδο σας. Η main θα κάνει τα εξής.

- Δημιουργεί μια στοίβα με μία τράπουλα και την εκτυπώνει. Καλεί την draw δύο φορές και εκτυπώνει τα χαρτιά που τραβήχτηκαν. Εκτυπώνει ξανά την στοίβα.
- Δημιουργεί μια νέα στοίβα με δύο τράπουλες και την εκτυπώνει. Ανακατεύει την στοίβα και την εκτυπώνει ξανά.

- Τραβάει συνέχεια δύο φύλλα από την στοίβα μέχρι να τραβήξει δύο φύλλα που είναι ίδια, ή να γίνει η στοίβα μισοάδεια. Εκτυπώνει κάθε φορά τα φύλλα που τράβηξε. Όταν τελειώσει η επανάληψη εκτυπώνει την στοίβα.

## Βήμα 2.

Στο βήμα αυτό θα υλοποιήσετε το παιχνίδι Between. Μπορείτε να διαβάσετε τους κανόνες του Between [εδώ](#). Στην εκδοχή που θα υλοποιήσετε υπάρχει ένας παίχτης και ένας dealer. Και οι δύο έχουν ξεκινάμε με κάποιο ποσό. Ο dealer τραβάει χαρτιά από μια στοίβα με 6 τράπουλες, ανακατεμένη τυχαία. Αν η στοίβα έχει πέσει στο μισό της μέγεθος, τότε δημιουργεί μια καινούρια στοίβα με 6 τράπουλες. Σε κάθε γύρο ο παίχτης βάζει ένα ευρώ (το ελάχιστο ποντάρισμα) για να μπει στο παιχνίδι. Ο dealer τραβάει δύο χαρτιά. Αν τα χαρτιά είναι ίδια, ή συνεχόμενα τότε ο παίχτης χάνει το ένα ευρώ (default loss). Αλλιώς, ζητάμε από τον παίκτη να ποντάρει. Δεν μπορεί να ποντάρει περισσότερα από τα χρήματα που έχει, ή τα χρήματα που έχει ο dealer (το pot). Ο dealer τραβάει ένα ακόμη χαρτί. Αν το χαρτί είναι ανάμεσα στα δύο αρχικά χαρτιά τότε ο παίχτης κερδίζει το ποσό που πόνταρε, αλλιώς χάνει και προστίθεται στο pot. Αν το χαρτί του παίκτη είναι ίδιο με κάποιο από τα δύο χαρτιά στο τραπέζι τότε χάνει. Το μικρότερο χαρτί είναι το 2 και το μεγαλύτερο ο Άσος.

Για την υλοποίηση θα πρέπει να τροποποιήσετε ή να δημιουργήσετε τις εξής κλάσεις:

**Κλάση Card:** Στην κλάση Card θα πρέπει να προσθέσετε τις εξής μεθόδους

- Τη μέθοδο **consecutive** η οποία παίρνει ως όρισμα ένα άλλο χαρτί και επιστρέφει μια Boolean τιμή αν το χαρτί-όρισμα και το χαρτί που καλεί τη μέθοδο είναι συνεχόμενα ή όχι.
- Τη μέθοδο **greaterThan** η οποία παίρνει ως όρισμα ένα άλλο χαρτί και επιστρέφει μια Boolean τιμή αν το χαρτί που καλεί τη μέθοδο είναι μεγαλύτερο από το χαρτί-όρισμα (μπορείτε εναλλακτικά να υλοποιήσετε και την lessThan).
- Τη μέθοδο **between** η οποία παίρνει σαν όρισμα ένα πίνακα με δύο χαρτιά και ελέγχει αν το χαρτί είναι μεταξύ του πρώτου και του δεύτερου (αυστηρά μεγαλύτερο από το πρώτο, και αυστηρά μικρότερο από το δεύτερο).

**Κλάση BetweenTable:** Η κλάση αυτή χειρίζεται το τραπέζι για ένα γύρο του παιχνιδιού. Έχει πεδία μια στοίβα με χαρτιά, ένα πίνακα με δύο χαρτιά που κρατάει τα χαρτιά που θα πέσουν στο τραπέζι, και ένα χαρτί που κρατάει το χαρτί που τραβάει ο παίχτης. Έχει επίσης τις εξής μεθόδους:

- Τον **constructor** ο οποίος παίρνει σαν όρισμα τη στοίβα με τα χαρτιά.
- Τη μέθοδο **drawTableCards** που τραβάει δύο χαρτιά από την στοίβα και τα αποθηκεύει στον πίνακα. Η μέθοδος θα πρέπει να τα αποθηκεύσει ώστε το μικρότερο χαρτί να αποθηκευτεί στην πρώτη θέση, και το μεγαλύτερο στην δεύτερη. Η μέθοδος επίσης εκτυπώνει τα δύο χαρτιά.
- Τη μέθοδο **defaultPlayerLoss** που επιστρέφει μια Boolean τιμή αν τα χαρτιά στο τραπέζι είναι τέτοια ώστε να έχουμε default loss του παίκτη.
- Τη μέθοδο **drawPlayerCard** η οποία τραβάει το χαρτί για τον παίκτη, το αποθηκεύει και το εκτυπώνει.
- Τη μέθοδο **betweenCard** που επιστρέφει μια Boolean τιμή αν το χαρτί του παίκτη είναι μεταξύ των δύο χαρτιών του τραπέζιού.

**Κλάση BetweenPlayer:** Η κλάση αυτή υλοποιεί τον παίκτη. Έχει πεδίο τα χρήματα του παίκτη τα οποία αρχικοποιούνται στον constructor. Έχει και τις εξής μεθόδους:

- Τη μέθοδο **bid** με την οποία ποντάρει ο παίχτης. Η μέθοδος παίρνει σαν όρισμα το ποσό που υπάρχει στο pot. Εκτυπώνει το ποσό καθώς και τα διαθέσιμα χρήματα του παίκτη και του ζητάει να δώσει από την είσοδο το ποσό που θέλει να ποντάρει. Το ποσό δεν μπορεί να είναι περισσότερα από το pot ή από τα χρήματα του παίκτη. Η μέθοδος επιστρέφει το ποντάρισμα (bid) του παίκτη. Όλα τα ποσά θα τα θεωρήσετε ακέραιους αριθμούς.
- Τη μέθοδο **addMoney** που παίρνει ένα ποσό και το προσθέτει στα χρήματα του παίκτη
- Τη μέθοδο **subtractMoney** που παίρνει ένα ποσό και το αφαιρεί από τα χρήματα του παίκτη
- Τη μέθοδο **isBroke** που επιστρέφει μια Boolean τιμή αν ο παίχτης έχει ξεμείνει από χρήματα ή όχι.

**Κλάση BetweenDealer:** Η κλάση αυτή υλοποιεί τον dealer του παιχνιδιού. Έχει πεδία μια στοίβα με χαρτιά, και τα χρήματα του dealer (το pot). Η κλάση έχει τις εξής μεθόδους.

- Τον **constructor** ο οποίος παίρνει σαν όρισμα τα χρήματα στο pot. Ο constructor επίσης δημιουργεί τη στοίβα με 6 τράπουλες και την ανακατεύει. Συνίσταται να κάνετε μια βοηθητική μέθοδο που θα δημιουργεί και θα ανακατεύει τη στοίβα γιατί θα σας χρειαστεί αργότερα.
- Την μέθοδο **emptyPot** η οποία επιστρέφει μια Boolean τιμή αν ο dealer έχει ξεμείνει από χρήματα.
- Τη μέθοδο **play** η οποία παίρνει σαν όρισμα ένα παίκτη (BetweenPlayer) και υλοποιεί ένα γύρο του παιχνιδιού μεταξύ του dealer και του παίκτη. Η μέθοδος ελέγχει καταρχάς αν η στοίβα είναι μισοάδεια. Αν ναι, την δημιουργεί και την ανακατεύει εκ νέου. Στη συνέχεια, δημιουργεί ένα αντικείμενο BetweenTable και υλοποιεί το παιχνίδι. Δηλαδή, τραβάει τα χαρτιά του τραπέζιού, και αν ο παίκτης χάνει by default του αφαιρεί ένα ευρώ και το προσθέτει στο pot. Αλλιώς ζητάει από τον παίκτη να ποντάρει, και τραβάει το χαρτί. Ανάλογα με το ποιος κερδίζει ενημερώνει τα χρήματα του παίκτη και το pot και τυπώνει αντίστοιχα μηνύματα.

**Κλάση Between:** Η κλάση αυτή έχει τη main που υλοποιεί το παιχνίδι. Ζητάει από το χρήστη να δώσει τα χρήματα για τον dealer και τον παίκτη. Δημιουργεί τα αντικείμενα και τρέχει το παιχνίδι μέχρι είτε ο παίκτης είτε ο dealer να μείνουν χωρίς χρήματα. Εκτυπώνει στο τέλος ένα μήνυμα ποιος κέρδισε.

Ένα παράδειγμα της εξόδου για την εκτέλεση του προγράμματος δίνεται στο τέλος της εκφώνησης.

### Βήμα 3.

Στο βήμα αυτό θα υλοποιήσετε το παιχνίδι της Ξερής. Αν δεν ξέρετε τις οδηγίες μπορείτε να τις διαβάσετε [εδώ](#) (στα Αγγλικά). Το παιχνίδι παίζεται με μια τράπουλα. Αρχικά, ο κάθε παίκτης παίρνει από 6 χαρτιά, και πέφτουν 4 χαρτιά στο τραπέζι σε μια στοίβα, όπου φαίνεται μόνο το χαρτί στην κορυφή της στοίβας (το τελευταίο). Οι παίκτες παίζουν εναλλάξ. Ο κάθε παίκτης στην σειρά του μπορεί να μαζέψει τα χαρτιά της στοίβας αν έχει ένα χαρτί που ταιριάζει στο νούμερο/φιγούρα με αυτό στην κορυφή της στοίβας, ή αν έχει βαλέ. Αν δεν μαζέψει τα χαρτιά ρίχνει ένα χαρτί στην στοίβα. Αν υπάρχει μόνο ένα χαρτί στην στοίβα και το μαζέψει ο παίκτης, κάνει ξερή. Όταν τελειώσουν τα χαρτιά στο χέρι, μοιράζονται εκ νέου 6 φύλλα σε κάθε παίκτη. Το παιχνίδι συνεχίζεται μέχρι να τελειώσει η τράπουλα. Για να προσδιοριστεί ο νικητής, κοιτάμε τα φύλλα που έχουν μαζέψει οι παίκτες, και τα βαθμολογούμε σύμφωνα με τις οδηγίες στον συνδεσμο.

Για την υλοποίηση θα πρέπει να τροποποιήσετε ή να δημιουργήσετε τις εξής κλάσεις:

**Κλάση Card:** Στην κλάση Card θα πρέπει να προσθέσετε Boolean μεθόδους που θα ελέγχουν για συγκεκριμένα χαρτιά: **isJack** (βαλές), **isHighCard** (μεγαλύτερο ή ίσο του 10), **isTwoClubs** (2 σπαθί), **isTenDiamond** (10 καρό).

**Κλάση CollectedCards:** Η κλάση αυτή κρατάει τα χαρτιά που μαζεύει από το τραπέζι ένας παίκτης. Έχει ένα πεδίο που είναι ένα ArrayList με τα χαρτιά, τις εξής μεθόδους:

- Δύο **constructors**, έναν χωρίς ορίσματα που δημιουργεί μια άδεια λίστα και έναν με όρισμα τη λίστα με τα χαρτιά.
- Τις Boolean μεθόδους **isEmpty**, **isXeri**, **isXeriWithJacks** η οποίες ελέγχουν αντίστοιχα αν η λίστα με τα χαρτιά είναι κενή, αν τα χαρτιά που μαζεύτηκαν δημιουργούν ξερή, και αν δημιουργούν ξερή με βαλέδες.
- **Accessor** μέθοδο για τη λίστα.

**Κλάση Pile:** Η κλάση υλοποιεί την στοίβα χαρτιών που έχουν πέσει στο τραπέζι. Για την υλοποίηση σας θα χρησιμοποιήσετε ένα ArrayList με χαρτιά, και θα κρατάτε και το χαρτί που βρίσκεται στην κορυφή της στοίβας (topCard), δηλαδή το τελευταίο χαρτί που προσθέσατε. Όταν η στοίβα είναι άδεια, το topCard θα είναι null. Η κλάση θα έχει τις εξής μεθόδους:

- Την μέθοδο **fill** η οποία παίρνει σαν όρισμα την τράπουλα (CardStack), και γεμίζει την στοίβα με τα 4 αρχικά χαρτιά.
- Την μέθοδο **drop**, με την οποία ο παίκτης παίζει ένα χαρτί, το οποίο δίνεται σαν όρισμα. Η μέθοδος δημιουργεί και επιστρέφει ένα αντικείμενο CollectedCards, με τα χαρτιά που μάζεψε ο παίκτης παίζοντας το χαρτί, σύμφωνα με τους κανόνες της Ξερής. Το αντικείμενο δεν θα περιέχει χαρτιά αν δεν μάζεψε χαρτιά ο παίκτης. Η μέθοδος ενημερώνει και τα πεδία της στοίβας ανάλογα.

- **Accessor** μέθοδο για το πεδίο topCard
- Την μέθοδο **printTopCard** η οποία τυπώνει το χαρτί στην κορυφή της στοίβας, ή το String “-” αν η στοίβα είναι άδεια. Μπορείτε να προσθέσετε επιπλέον πληροφορία στην εκτύπωση αν θέλετε (δείτε το παράδειγμα εξόδου).
- Την μέθοδο **toString** που επιστρέφει ένα String με όλα τα χαρτιά στη στοίβα, ή το String “-” αν η στοίβα είναι κενή. Μπορείτε να προσθέσετε επιπλέον πληροφορία στο String αν θέλετε.

**Κλάση Hand:** Η κλάση υλοποιεί το χέρι ενός παίχτη. Έχει πεδίο ένα ArrayList με τα χαρτιά που κρατάει στο χέρι του ο παίχτης, και τις εξής μεθόδους:

- Την μέθοδο **addCard** η οποία προσθέτει ένα χαρτί στο χέρι.
- Την μέθοδο **print** η οποία τυπώνει το χέρι. Τυπώστε το χέρι σε δύο γραμμές, μία με τα χαρτιά και μία με τις θέσεις των χαρτιών. Ένα παράδειγμα της εξόδου δίνεται στο τέλος της εκφώνησης.
- Την μέθοδο **getCard** η οποία παίρνει σαν όρισμα μια θέση, αφαιρεί από το χέρι το φύλλο σε αυτή τη θέση και το επιστρέφει.
- Την μέθοδο **selectComputerCard** η οποία παίρνει σαν όρισμα το χαρτί στην κορυφή της στοίβας και υλοποιεί την στρατηγική του παίχτη-υπολογιστή: Αν υπάρχει φύλλο στο χέρι που να ταιριάζει με το φύλλο στην κορυφή της στοίβας το επιστρέφει. Αλλιώς, αν υπάρχει βαλές στο χέρι, τον επιστρέφει. Αλλιώς επιστρέφει ένα τυχαίο φύλλο από το χέρι. Το φύλλο που επιστρέφεται, αφαιρείται από την στοίβα.
- Προαιρετικά την μέθοδο isEmpty που ελέγχει αν το χέρι έχει φύλλα (αν το χρειαστείτε).

**Κλάση XeriPlayer:** Η κλάση υλοποιεί το παιχνίδι ενός παίχτη. Έχει πεδία το όνομα του παίχτη που αρχικοποιείται στον constructor, ένα αντικείμενο Hand, ένα ArrayList με τα χαρτιά που έχει μαζέψει ο παίχτης, ένα μετρητή για το πόσες ξερές έχει κάνει, και ένα πεδίο που θα κρατάει τους πόντους (points) που έχει μαζέψει. Η κλάση θα έχει τις εξής μεθόδους:

- Την μέθοδο **getCards** η οποία παίρνει σαν όρισμα την τράπουλα (CardStack) και μοιράζει 6 χαρτιά στον παίχτη.
- Την μέθοδο **selectCard** η οποία εμφανίζει το χέρι στον παίχτη-χρήστη και του ζητάει την θέση του χαρτιού που θέλει να παίξει. Επιστρέφει το χαρτί που επέλεξε ο παίχτης.
- Την μέθοδο **selectComputerCard** η οποία επιστρέφει το χαρτί που επέλεξε ο παίχτης-υπολογιστής. Για σκοπούς debugging μπορείτε να τυπώνετε και το χέρι του παίχτη πριν την επιλογή.
- Την μέθοδο **collectCards** η οποία παίρνει όρισμα ένα αντικείμενο CollectedCards και αν το αντικείμενο έχει χαρτιά, τα προσθέτει στη λίστα. Επίσης ελέγχει αν ο παίχτης έχει κάνει ξερή και ενημερώνει τον μετρητή. Λάβετε υπόψη ότι ξερή με δύο βαλέδες μετράει διπλά.
- Την μέθοδο **computePoints** η οποία υπολογίζει τους πόντους που έχει στο χέρι του ο παίχτης και τους αποθηκεύει στο πεδίο points.
- Την μέθοδο **compareCardCount** που παίρνει σαν όρισμα ένα άλλο παίχτη και προσθέτει τους πόντους (3) στον παίχτη που έχει μαζέψει τα περισσότερα φύλλα (αν έχουν ίδιο αριθμό από φύλλα δεν τους παίρνει κανείς).
- Προαιρετικά την μέθοδο outOfCards η οποία ελέγχει αν έχει φύλλα το χέρι του παίχτη.

**Κλάση Xeri:** Η μέθοδος αυτή περιέχει τη main η οποία υλοποιεί το παιχνίδι. Δημιουργεί δύο παίχτες, ο πρώτος ελέγχεται από τον χρήστη και ο δεύτερος από τον υπολογιστή. Επίσης μια τράπουλα (CardStack) την οποία ανακατεύει, και μια στοίβα (Pile) την οποία γεμίζει και τυπώνει. Στην συνέχεια υλοποιεί το παιχνίδι των παιχτών. Πριν παίξει ένας παίχτης τυπώνεται το χαρτί στην κορυφή της στοίβας (για debugging μπορείτε να τυπώνετε και ολόκληρη την στοίβα). Όταν τελειώσει η τράπουλα υπολογίζουμε τους πόντους, και τους συγκρίνουμε για να αποφασίσουμε ποιος κερδίζει. Η υλοποίηση σας θα χρησιμοποιήσει τις μεθόδους των κλάσεων XeriPlayer και Pile, για να επιλεγεί το χαρτί που θα παίξει ο παίχτης, να το ρίξει στο τραπέζι, να μαζέψει τα χαρτιά, να μοιράσει νέα χαρτιά και να υπολογίσει τους πόντους. Ενδεικτική έξοδος για το πρόγραμμα δίνεται στο τέλος της εκφώνησης.

## 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.
- Τα προγράμματα σας θα ελεγχθούν με λογισμικό για εντοπισμό αντιγραφών, και εντοπισμό χρήσης Τεχνητής Νοημοσύνης. Για επιβεβαιωμένες αντιγραφές μεγάλης κλίμακας, θα υπάρχουν ποινές πέρα από το μηδενισμό της άσκησης (π.χ., αποκλεισμό από τις εξετάσεις).
- Κάντε turnin τα προγράμματα σας στο **assignment1@myy205**.

π.χ. turnin assignment1@myy205 MineSweeper.java

Μπορείτε να κάνετε turnin πολλά αρχεία μαζί στην ίδια εντολή. Διαβάστε προσεκτικά τις οδηγίες για το turnin στο ecourse και βεβαιωθείτε ότι μπορείτε να κάνετε την διαδικασία κάποιες μέρες πριν την προθεσμία. Μην το αφήσετε αυτό για την τελευταία στιγμή! Μπορείτε να κάνετε πολλαπλές φορές turnin τα ίδια αρχεία, θα κοιτάζουμε 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 εξόδου για το Βήμα 1:

```
>java CardStack
```

```
2S,3S,4S,5S,6S,7S,8S,9S,10S,JS,QS,KS,AS,2H,3H,4H,5H,6H,7H,8H,9H,10H,JH,QH,KH,AH
,2D,3D,4D,5D,6D,7D,8D,9D,10D,JD,QD,KD,AD,2C,3C,4C,5C,6C,7C,8C,9C,10C,JC,QC,KC,A
C,
AC
KC
2S,3S,4S,5S,6S,7S,8S,9S,10S,JS,QS,KS,AS,2H,3H,4H,5H,6H,7H,8H,9H,10H,JH,QH,KH,AH
,2D,3D,4D,5D,6D,7D,8D,9D,10D,JD,QD,KD,AD,2C,3C,4C,5C,6C,7C,8C,9C,10C,JC,QC,
2S,3S,4S,5S,6S,7S,8S,9S,10S,JS,QS,KS,AS,2H,3H,4H,5H,6H,7H,8H,9H,10H,JH,QH,KH,AH
,2D,3D,4D,5D,6D,7D,8D,9D,10D,JD,QD,KD,AD,2C,3C,4C,5C,6C,7C,8C,9C,10C,JC,QC,KC,A
C,2S,3S,4S,5S,6S,7S,8S,9S,10S,JS,QS,KS,AS,2H,3H,4H,5H,6H,7H,8H,9H,10H,JH,QH,KH,
AH,2D,3D,4D,5D,6D,7D,8D,9D,10D,JD,QD,KD,AD,2C,3C,4C,5C,6C,7C,8C,9C,10C,JC,QC,KC
,AC,
AC,10C,QC,7H,JD,8C,7C,6C,5C,4C,6S,AS,JD,KD,QS,6D,3S,7S,5D,KH,7D,10S,AS,3S,10H,4
S,AH,QH,JS,5S,10S,KC,6H,4D,3C,8D,2C,4C,8D,KS,7C,9S,8S,3D,4H,5S,8H,5H,2S,JS,7H,2
H,2D,7D,KD,QD,9C,3D,JH,3H,9S,3H,JH,10C,6C,10H,9D,6H,6D,AC,JC,4H,6S,QC,4D,KH,QS,
8H,4S,2H,9H,9H,5H,QH,10D,8C,KS,5D,2S,5C,AH,2D,8S,QD,9D,2C,AD,AD,3C,10D,7S,JC,9C
,KC,
KC 9C
JC 7S
10D 3C
AD AD
AC,10C,QC,7H,JD,8C,7C,6C,5C,4C,6S,AS,JD,KD,QS,6D,3S,7S,5D,KH,7D,10S,AS,3S,10H,4
S,AH,QH,JS,5S,10S,KC,6H,4D,3C,8D,2C,4C,8D,KS,7C,9S,8S,3D,4H,5S,8H,5H,2S,JS,7H,2
```

H, 2D, 7D, KD, QD, 9C, 3D, JH, 3H, 9S, 3H, JH, 10C, 6C, 10H, 9D, 6H, 6D, AC, JC, 4H, 6S, QC, 4D, KH, QS, 8H, 4S, 2H, 9H, 9H, 5H, QH, 10D, 8C, KS, 5D, 2S, 5C, AH, 2D, 8S, QD, 9D, 2C,

## Παράδειγμα εξόδου για το Βήμα 2

```
>java Between
Pot money?
100
Player money?
20
Table Cards: QC KC
Default loss of player
```

```
Table Cards: JC JC
Default loss of player
```

```
Table Cards: 6C 9C
Current pot:102
Your money:18
Place your bet:1
Player Card: 4C
Player loses 1
```

```
Table Cards: 8H AH
Current pot:103
Your money:17
Place your bet:1
Player Card: AD
Player loses 1
```

```
Table Cards: 3H 6D
Current pot:104
Your money:16
Place your bet:1
Player Card: 6H
Player loses 1
```

```
Table Cards: QD KH
Default loss of player
```

```
Table Cards: 8D 9S
Default loss of player
```

```
Table Cards: 7D JH
Current pot:107
Your money:13
Place your bet:1
Player Card: KC
Player loses 1
```

```
Table Cards: 3D 5D
Current pot:108
Your money:12
Place your bet:1
Player Card: AD
Player loses 1
```

```
Table Cards: 2D 5H
Current pot:109
Your money:11
```

Place your bet:1  
Player Card: 6H  
Player loses 1

Table Cards: JH QH  
Default loss of player

Table Cards: 7D 8S  
Default loss of player

Table Cards: 4D 7C  
Current pot:112  
Your money:8  
Place your bet:1  
Player Card: 2S  
Player loses 1

Table Cards: 5H AC  
Current pot:113  
Your money:7  
Place your bet:7  
Player Card: QC  
Player wins 7

Table Cards: 2D 10S  
Current pot:106  
Your money:14  
Place your bet:5  
Player Card: 2D  
Player loses 5

Table Cards: 3S KC  
Current pot:111  
Your money:9  
Place your bet:9  
Player Card: KD  
Player loses 9

Player lost!

### Παράδειγμα εξόδου για το Βήμα 3:

```
>java Xeri
```

Pile Cards:KC QC 7D 10C

Player Human turn

Top Card: 10C

6D	4S	10D	7S	JH	2C
0	1	2	3	4	5

Select Card position:2

Player Human selected card: 10D

Player Computer turn

Top Card: -

Player Computer selected card: 4H

Player Human turn

Top Card: 4H

6D	4S	7S	JH	2C
0	1	2	3	4

Select Card position:1

Player Human selected card: 4S

Player Human made xeri!

Player Computer turn

Top Card: -

Player Computer selected card: 3C

Player Human turn

Top Card: 3C

6D	7S	JH	2C
----	----	----	----

0	1	2	3
---	---	---	---

Select Card position:2

Player Human selected card: JH

Player Computer turn

Top Card: -

Player Computer selected card: 9D

Player Human turn

Top Card: 9D

6D	7S	2C
----	----	----

0	1	2
---	---	---

Select Card position:0

Player Human selected card: 6D

Player Computer turn

Top Card: 6D

Player Computer selected card: 6H

Player Human turn

Top Card: -

7S	2C
----	----

0	1
---	---

Select Card position:0

Player Human selected card: 7S

Player Computer turn

Top Card: 7S

Player Computer selected card: KD

Player Human turn

Top Card: KD

2C

0

Select Card position:0

Player Human selected card: 2C

Player Computer turn

Top Card: 2C

Player Computer selected card: 5D

Player Human turn

Top Card: 5D

KH	5H	9H	AH	8C	QD
----	----	----	----	----	----

0	1	2	3	4	5
---	---	---	---	---	---

Select Card position:1

Player Human selected card: 5H



Player Computer turn  
Top Card: -  
Player Computer selected card: 6S

Player Human turn  
Top Card: 6S  
KH 9H AH 8C QD  
0 1 2 3 4  
Select Card position:3  
Player Human selected card: 8C

Player Computer turn  
Top Card: 8C  
Player Computer selected card: JD

Player Human turn  
Top Card: -  
KH 9H AH QD  
0 1 2 3  
Select Card position:1  
Player Human selected card: 9H

Player Computer turn  
Top Card: 9H  
Player Computer selected card: 3D

Player Human turn  
Top Card: 3D  
KH AH QD  
0 1 2  
Select Card position:0  
Player Human selected card: KH

Player Computer turn  
Top Card: KH  
Player Computer selected card: 10H

Player Human turn  
Top Card: 10H  
AH QD  
0 1  
Select Card position:1  
Player Human selected card: QD

Player Computer turn  
Top Card: QD  
Player Computer selected card: AS

Player Human turn  
Top Card: AS  
AH  
0  
Select Card position:0  
Player Human selected card: AH

Player Computer turn  
Top Card: -  
Player Computer selected card: 2H

Player Human turn  
Top Card: 2H

2D	7H	4D	QH	8H	8S
0	1	2	3	4	5

Select Card position:0

Player Human selected card: 2D

Player Human made xeri!

Player Computer turn

Top Card: -

Player Computer selected card: 5S

Player Human turn

Top Card: 5S

7H	4D	QH	8H	8S
0	1	2	3	4

Select Card position:3

Player Human selected card: 8H

Player Computer turn

Top Card: 8H

Player Computer selected card: 8D

Player Human turn

Top Card: -

7H	4D	QH	8S
0	1	2	3

Select Card position:3

Player Human selected card: 8S

Player Computer turn

Top Card: 8S

Player Computer selected card: 3S

Player Human turn

Top Card: 3S

7H	4D	QH
0	1	2

Select Card position:1

Player Human selected card: 4D

Player Computer turn

Top Card: 4D

Player Computer selected card: KS

Player Human turn

Top Card: KS

7H	QH
0	1

Select Card position:0

Player Human selected card: 7H

Player Computer turn

Top Card: 7H

Player Computer selected card: 10S

Player Human turn

Top Card: 10S

QH
0

Select Card position:0

Player Human selected card: QH

Player Computer turn  
Top Card: QH  
Player Computer selected card: QS

Player Human turn  
Top Card: -  
9S 3H 4C AD 6C 2S  
0 1 2 3 4 5  
Select Card position:5  
Player Human selected card: 2S

Player Computer turn  
Top Card: 2S  
Player Computer selected card: JS

Player Human turn  
Top Card: -  
9S 3H 4C AD 6C  
0 1 2 3 4  
Select Card position:2  
Player Human selected card: 4C

Player Computer turn  
Top Card: 4C  
Player Computer selected card: JC

Player Human turn  
Top Card: -  
9S 3H AD 6C  
0 1 2 3  
Select Card position:1  
Player Human selected card: 3H

Player Computer turn  
Top Card: 3H  
Player Computer selected card: 9C

Player Human turn  
Top Card: 9C  
9S AD 6C  
0 1 2  
Select Card position:0  
Player Human selected card: 9S

Player Computer turn  
Top Card: -  
Player Computer selected card: AC

Player Human turn  
Top Card: AC  
AD 6C  
0 1  
Select Card position:0  
Player Human selected card: AD  
Player Human made xeri!

Player Computer turn  
Top Card: -  
Player Computer selected card: 7C

Player Human turn

Top Card: 7C

6C

0

Select Card position:0

Player Human selected card: 6C

Player Computer turn

Top Card: 6C

Player Computer selected card: 5C

Player Human has 48 points, player Computer has 7 points

Human won!