

**Πρώτη Άσκηση**  
**Ημερομηνία Παράδοσης: 10 Μαΐου 2021, 12 μ.μ.**

Για την άσκηση αυτή θα υλοποιήσετε σε Java ένα πρόγραμμα που θα υλοποιεί μια απλή εκδοχή του γνωστού επιτραπέζιου παιχνιδιού μνήμης. Στο παιχνίδι αυτό υπάρχουν  $2*N$  κάρτες που αποτελούν  $N$  ζευγάρια από εικόνες. Οι κάρτες τοποθετούνται κλειστές σε τυχαία διάταξη στο τραπέζι. Σε κάθε γύρο ο κάθε παίχτης ανοίγει δύο από τις κάρτες. Αν οι κάρτες είναι ίδιες ο παίχτης μαζεύει τις κάρτες και κερδίζει ένα πόντο. Οι κάρτες αφαιρούνται από το τραπέζι. Αλλιώς οι κάρτες κλείνουν και συνεχίζει ο επόμενος παίχτης. Το παιχνίδι τελειώνει όταν όλα τα ζευγάρια έχουν βρεθεί. Κερδίζει ο παίχτης που έχει μαζέψει τους περισσότερους πόντους.

Στην απλοποιημένη εκδοχή που θα υλοποιήσετε, αντί για ζευγάρια από 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 κάρτες θα τοποθετηθούν τυχαία πάνω σε μία γραμμή. Το παιχνίδι θα παίζεται είτε μεταξύ ενός παίχτη-ανθρώπου και ενός παίχτη-υπολογιστή, είτε μεταξύ δύο παιχτών-ανθρώπων. Σε κάθε γύρο ο παίχτης-άνθρωπος διαλέγει τις δύο κάρτες που θα ανοίξει ταυτόχρονα (δηλαδή επιλέγει την δεύτερη πριν ανοίξει την πρώτη). Ο παίχτης-υπολογιστής επιλέγει δύο κάρτες με μια τακτική που περιγράφεται παρακάτω.

Για την υλοποίηση 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 **MemoryGame** που υλοποιεί την ροή του παιχνιδιού.

**Board:** Η πιο σημαντική κλάση που κρατάει πληροφορία για τις «κάρτες» και το τραπέζι. Οι κάρτες-αριθμοί αποθηκεύονται σε τυχαία σειρά σε ένα μονοδιάστατο πίνακα ακεραίων με  $2*N$  θέσεις, όπου  $N$  είναι ο αριθμός των ζευγαριών. Τον πίνακα μπορείτε να τον σκέφτεστε σαν την διάταξη των καρτών σε μία γραμμή, όπου η κάθε θέση είναι μια κάρτα. Χρειάζεστε επίσης να κρατάτε την πληροφορία για το ποιες θέσεις του πίνακα περιέχουν ακόμη κάρτες και από ποιες θέσεις οι κάρτες έχουν αφαιρεθεί. Θα είναι επίσης χρήσιμο να κρατάτε το μέγεθος του πίνακα και τον αριθμό των καρτών που υπάρχουν ακόμη στο τραπέζι. Μπορείτε να έχετε κι άλλες δομές ή πεδία που σας εξυπηρετούν για την υλοποίηση των μεθόδων. Τέλος, θα έχετε σαν πεδίο και μια μεταβλητή τύπου Random την οποία θα χρησιμοποιήσετε για την παραγωγή τυχαίων αριθμών.

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 σε αύξοντα σειρά και να κάνετε μια τυχαία αναδιάταξη των καρτών. Ένας άλλος είναι για κάθε κάρτα να διαλέξετε τυχαία μια από τις εναπομείναντες θέσεις.  
Για να μετατρέψετε μια διάταξη σε τυχαία διάταξη μπορείτε για κάθε θέση να θέσετε την τιμή για αυτή τη θέση, ανταλλάσσοντας το στοιχείο στη θέση αυτή με το στοιχείο μιας τυχαία επιλεγμένης θέσης, από αυτές που δεν έχετε ήδη θέσει. Βολεύει να εξετάσετε τις θέσεις από το τέλος προς την αρχή.
- Την μέθοδο **print** η οποία θα εκτυπώνει το «τραπέζι». Για τις κάρτες στον πίνακα θα τυπώνει '\*', ενώ για τις κάρτες που έχουν αφαιρεθεί θα τυπώνει το κενό. Τυπώστε και μία επικεφαλίδα με τα νούμερα των θέσεων.
- Την μέθοδο **flash** η οποία παίρνει σαν όρισμα δύο θέσεις του πίνακα (τις οποίες έχει διαλέξει κάποιος παίχτης) και τυπώνει το τραπέζι όπως και πριν αλλά επιπλέον τυπώνει και τις κάρτες-αριθμούς στις δύο θέσεις τις οποίες διάλεξε ο χρήστης. Η μέθοδος υποθέτει ότι οι θέσεις που παίρνει σαν όρισμα περιέχουν κάρτες. Αυτή η εκτύπωση θα εμφανίζεται μόνο για μερικά δευτερόλεπτα και μετά θα τυπώνεται από πάνω το ίδιο όπως και με την print (δηλαδή θα τυπώνει '\*' και για τις θέσεις που διάλεξε ο παίχτης). Βοηθητικό κώδικα για το πώς να κάνετε την εκτύπωση και την καθυστέρηση θα βρείτε στις υποδείξεις.

- Την μέθοδο **openPositions** η οποία παίρνει σαν όρισμα δύο θέσεις του πίνακα και «ανοίγει τις κάρτες» σε εκείνες τις θέσεις. Η μέθοδος υποθέτει ότι οι θέσεις που παίρνει σαν όρισμα περιέχουν κάρτες. Επιστρέφει μια Boolean τιμή αν βρέθηκε ένα ζευγάρι. Η μέθοδος θα ελέγχει αν οι αριθμοί σε αυτές τις θέσεις είναι ίδιοι. Αν ναι, τυπώνει ένα μήνυμα για το ζευγάρι που βρέθηκε, αφαιρεί τις κάρτες, καλεί την print και επιστρέφει true. Αλλιώς καλεί την flash με όρισμα αυτές τις θέσεις και επιστρέφει false.
- Την μέθοδο **getRandomPosition** η οποία επιστρέφει μια τυχαία θέση στον πίνακα από αυτές που περιέχουν κάρτα.
- Υπερφορτώστε την **getRandomPosition** ώστε να παίρνει σαν όρισμα μια θέση και να επιστρέφει μια τυχαία θέση (που περιέχει κάρτα) διαφορετική από την θέση-όρισμα.
- Την μέθοδο **containsCard** η οποία παίρνει σαν όρισμα μια θέση και επιστρέφει μια Boolean τιμή αν η θέση περιέχει κάρτα ή όχι.
- Την μέθοδο **getCard** η οποία παίρνει μία θέση του πίνακα και επιστρέφει τον αριθμό-κάρτα σε εκείνη την θέση.
- Την μέθοδο **allPairsFound** η οποία επιστρέφει true αν όλα τα ζευγάρια έχουν βρεθεί, δηλαδή όλες οι κάρτες έχουν αφαιρεθεί.

Αποθηκεύστε τον κώδικα σας στο αρχείο **Board.java**.

Δημιουργήστε και μια μέθοδο **main** στην κλάση Board για να τεστάρετε τον κώδικα σας καλώντας όλες τις μεθόδους της κλάσης σας. Δημιουργήστε ένα Board με 3 ζευγάρια και καλέστε την print αρχικά και μετά την getCard για όλες τις θέσεις ώστε να ξέρετε την διάταξη. Χρησιμοποιήστε την getRandomPosition για να πάρετε δύο διαφορετικές τυχαίες θέσεις. Καλέστε την flash με αυτές τις θέσεις. Καλέστε μια φορά την openPositions για δύο θέσεις που δεν είναι ζευγάρι και μια φορά για δύο θέσεις που είναι ζευγάρι. Εκτυπώστε το τραπέζι κάθε φορά και καλέστε την containsCard για μία θέση με και χωρίς κάρτα. Καλέστε την openPositions για άλλο ένα ζευγάρι, και μετά πάρετε ξανά δύο τυχαίες θέσεις. Ανοίξτε και τις τελευταίες κάρτες και καλέστε την allPairsFound.

Θα σας βοηθήσει αν για την διενέργεια των τεστ σας δημιουργήσετε το αντικείμενο Random με όρισμα ένα σταθερό ακέραιο αριθμό (το seed), ώστε σε κάθε διαφορετικό τρέξιμο να κάνει πάντα τις ίδιες τυχαίες επιλογές.

#### Υποδείξεις για την Board:

- Θα σας βολέψει να ορίσετε κάποιες βοηθητικές (private) μεθόδους για κάποιες σύνθετες λειτουργίες όπως η τοποθέτηση των καρτών στον πίνακα, ή για επαναλαμβανόμενες διαδικασίες όπως η εκτύπωση της επικεφαλίδας. Επίσης μην ξεχνάτε ότι μπορείτε να χρησιμοποιείτε και τις public μεθόδους ως βοηθητικές μεθόδους.
- 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 είσατε 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. Προσοχή, δεν πρέπει να καλέσετε println γιατί θα αλλάξει γραμμή και δεν μπορείτε να επιστρέψετε στην 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 ΒοηθητικόςΚώδικας.txt.

**ComputerPlayer:** Η κλάση αυτή υλοποιεί το παιχνίδι του παίκτη-υπολογιστή. Έχει ένα πεδίο που είναι το όνομα που θα δώσουμε στον παίκτη. Ο παίκτης-υπολογιστής θα διατηρεί ένα πίνακα με N θέσεις όπου για κάθε αριθμό-κάρτα θα κρατάει την θέση στην οποία είδε αυτόν τον αριθμό τελευταία φορά, ή -1 αν δεν τον έχει ξαναδεί. Επίσης θα έχει ένα πεδίο με τους πόντους που έχει μαζέψει ο παίκτης.

Θα έχει τις εξής μεθόδους:

- Τον **constructor** ο οποίος παίρνει σαν όρισμα το όνομα του παίκτη και τον αριθμό των ζευγαριών και αρχικοποιεί όλες τις μεταβλητές κατάλληλα.
- Την μέθοδο **play** η οποία θα παίρνει σαν όρισμα ένα αντικείμενο Board και θα υλοποιεί την στρατηγική του παίκτη. Την πρώτη θέση ο παίκτης την διαλέγει τυχαία, από τις θέσεις που έχουν κάρτα. Αν τον αριθμό σε αυτή την θέση τον έχει ξαναδεί σε κάποια άλλη θέση στο παρελθόν επιλέγει αυτή τη θέση σαν δεύτερη επιλογή. Αν όχι, επιλέγει και την δεύτερη θέση τυχαία, από τις θέσεις που περιέχουν κάρτα. Οι θέσεις που θα επιλέξει τελικά θα πρέπει να είναι διαφορετικές μεταξύ τους. Εκτυπώνονται οι θέσεις που διάλεξε ο υπολογιστής, και ανοίγονται οι θέσεις (κλήση της openPositions). Σε περίπτωση που δεν βρέθηκε ένα ζευγάρι, για κάθε αριθμό-κάρτα που άνοιξε ο παίκτης κρατάει την θέση στην οποία εμφανίστηκε. Αν βρέθηκε ζευγάρι, ενημερώνει τους πόντους του παίκτη.
- Μία **μέθοδο προσπέλασης (accessor method)** για τον αριθμό των πόντων που έχει μαζέψει ο παίκτης.
- Την μέθοδο **toString** που επιστρέφει ένα String με το όνομα του παίκτη.

Αποθηκεύστε τον κώδικα σας στο αρχείο **ComputerPlayer.java**.

Δημιουργήστε μια μέθοδο **main** μέσα στην ComputerPlayer για να τεστάρετε τον κώδικα σας. Δημιουργήστε αντικείμενα Board και ComputerPlayer, και καλέστε αρκετές φορές την play ώστε να εξετάσετε διαφορετικά σενάρια. Και πάλι θα βοηθήσει αν έχετε φιάρει το seed του Random αντικειμένου.

**HumanPlayer:** Η κλάση αυτή θα υλοποιεί το παιχνίδι του παίκτη-ανθρώπου. Έχει ένα πεδίο που είναι το όνομα του παίκτη. Η κλάση θα πρέπει να κρατάει πληροφορία για τους πόντους που έχει μαζέψει ο παίκτης.

Θα έχει τις εξής μεθόδους:

- Τον **constructor** ο οποίος παίρνει σαν όρισμα το όνομα του παίκτη και αρχικοποιεί το πεδίο.
- Την μέθοδο **play** η οποία παίρνει σαν όρισμα ένα αντικείμενο Board και υλοποιεί το παιχνίδι του χρήστη. Η μέθοδος ζητάει από τον παίκτη να δώσει δύο θέσεις από την είσοδο. Ελέγχει αν οι θέσεις είναι αποδεκτές, δηλαδή είναι διαφορετικές μεταξύ τους και είναι θέσεις που περιέχουν κάρτες, και συνεχίζει να ζητάει θέσεις μέχρι να πάρει αποδεκτές τιμές. Ανοίγει τις θέσεις. Αν βρέθηκε ζευγάρι, ενημερώνει τους πόντους του παίκτη.
- Μία **μέθοδο προσπέλασης (accessor method)** για τον αριθμό των πόντων που έχει μαζέψει ο παίκτης.
- Την μέθοδο **toString** που επιστρέφει ένα String με το όνομα του παίκτη.

Αποθηκεύστε τον κώδικα σας στο αρχείο **HumanPlayer.java**.

**MemoryGame:** Η κλάση που περιέχει την μέθοδο **main**, η οποία υλοποιεί τη βασική ροή του παιχνιδιού. Η main θα παίρνει σαν command line argument τον αριθμό των ζευγαριών. Δημιουργεί το Board. Ρωτάει αν θέλουμε ένα παιχνίδι μεταξύ παίκτη-ανθρώπου και παίκτη-υπολογιστή, ή μεταξύ δύο παιχτών-ανθρώπων, δημιουργεί τα αντικείμενα για τους παίκτες, και υλοποιεί το παιχνίδι. Σε κάθε γύρο του παιχνιδιού, οι παίκτες παίζουν εναλλάξ. Στο τέλος κάθε 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).

Αποθηκεύστε τον κώδικα σας στο αρχείο **MemoryGame.java**.

### **Bonus:**

Στην τρέχουσα υλοποίηση ο παίχτης-υπολογιστής κρατάει μόνο την θέση στην οποία είδε την τελευταία φορά μία κάρτα-αριθμό. Αν πχ αρχικά δει την κάρτα «1» στην θέση 10, και μετά την δει στην θέση 13 θα κρατήσει μόνο την τιμή 13. Τροποποιήστε την κλάση ComputerPlayer ώστε να κρατάει και τις δύο θέσεις. Επίσης τροποποιήστε την στρατηγική του παίχτη ώστε αν υπάρχουν κάρτες για τις οποίες έχει αποθηκεύσει δύο θέσεις στον επόμενο γύρο να επιλέγει αυτές τις θέσεις (εφόσον δεν τις έχει ήδη επιλέξει ο άλλος παίχτης).

Υπάρχουν διάφοροι τρόποι να υλοποιηθεί η παραπάνω ιδέα. Ο κώδικας σας θα βαθμολογηθεί και για την ποιότητα της λύσης που δώσατε.

### **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 Simulation.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. Κάθε φορά πρέπει να κάνετε turnin όλα τα αρχεία που θέλετε να παραδώσετε. Μπορείτε να κάνετε πολλαπλές φορές turnin τα ίδια αρχεία, θα κοιτάξουμε το τελευταίο. 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).

### **Παράδειγμα Εξόδου:**

Παρακάτω δίνεται ένα ενδεικτικό παράδειγμα της εξόδου ενός γύρου. Δεν είναι ανάγκη η έξοδος σας να είναι ακριβώς έτσι αλλά πρέπει να είναι παρόμοια. Το παιχνίδι είναι μεταξύ του παίχτη-ανθρώπου Panayiotis και του παίχτη-υπολογιστή HAL. Στο ξεκίνημα τυπώθηκε το τραπέζι. Η κατάσταση του τραπέζιου στο ξεκίνημα ενός γύρου φαίνεται από την τελευταία εκτύπωση του προηγούμενου γύρου.

```
Panayiotis, select two positions to open: 15 19
Selection not valid
```

```
Panayiotis, select two positions to open: 15 16
```

```
 0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
-----
  *       *       *   *               *   *       *   *   *   *
```

```
Computer player HAL selected positions :16 5
Found pair (0,0)
```

```
 0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
-----
  *       *       *               *   *       *   *       *
```

```
Panayiotis points: 3    HAL points: 3
```

Όταν ανοίγουν οι θέσεις 15,16 εμφανίζεται για 10 δευτερόλεπτα το παρακάτω

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
-----																			
	*		*		*	*					*	*		*	9	0	*		

Μετά τα 10 δευτερόλεπτα γράφεται από πάνω αυτό που βλέπετε παραπάνω.