

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Το Φιδάκι

Το φιδάκι είναι ένα κλασσικό διασκεδαστικό επιτραπέζιο παιχνίδι, το οποίο παίζεται σε ένα ταμπλό χτισμένο σε επίπεδα, τα οποία ανεβαίνεις εάν πέσεις σε βάση σκάλας και κατεβαίνεις εάν πέσεις σε στόμα από φιδάκι. Το παιχνίδι παίζεται με 2-4 παίκτες και στόχος είναι να φτάσουν στο τελευταίο πλακίδιο του ταμπλό αποφεύγοντας τα εμπόδια και μαζεύοντας πόντους.



Εικόνα 1: Παράδειγμα ταμπλό παιχνιδιού διάστασης 5x6.

Στην αρχή του παιχνιδιού, όλοι οι παίκτες ξεκινάνε από το 0 και προσπαθούν να καταλήξουν στο τέρμα. Ρίχνουν το ζάρι και όποιος ρίξει το μικρότερο αριθμό παίζει πρώτος, εκείνος που έχει το δεύτερο μικρότερο αριθμό παίζει δεύτερος κ.ο.κ. Ο πρώτος παίκτης ρίχνει ξανά το ζάρι και προχωράει όσα βήματα του δείχνει αυτό. Εάν φτάσει σε κάποιο από τα πλακίδια που του δείχνει ότι ξεκινάει η σκάλα, την ανεβαίνει. Η κάθε σκάλα μπορεί να χρησιμοποιηθεί μόνο μια φορά κατά τη διάρκεια του παιχνιδιού. Αυτό σημαίνει ότι όταν ένας παίκτης ανέβει μια σκάλα, στη συνέχεια η σκάλα σπάει και δε μπορεί να ξαναχρησιμοποιηθεί. Αν όμως φτάσει σε κάποιο σημείο του ταμπλό στο οποίο βρίσκεται ένα φιδάκι, κατεβαίνει στο αντίστοιχο πλακίδιο που τελειώνει η ουρά του.

Εκτός όμως από τα φιδάκια και από τις σκάλες, ένα πλακίδιο μπορεί να περιέχει μήλο, το οποίο ανάλογα με το χρώμα του (κόκκινο ή μαύρο) προσθέτει ή αφαιρεί πόντους από το σκορ του παίκτη που πέφτει πάνω του. Όταν ένας παίκτης τρώει ένα μήλο, τότε αυτό εξαφανίζεται από το ταμπλό.

Οι παίκτες παίζουν κυκλικά σύμφωνα με τη σειρά που έχει οριστεί στην αρχή του παιχνιδιού, δηλαδή όταν τελειώσει ο πρώτος παίκτης τη σειρά του, συνεχίζει ο δεύτερος, στη συνέχεια ο τρίτος κ.ο.κ.

Νικητής του παιχνιδιού είναι ο παίκτης που θα καταφέρει να φτάσει στο τελευταίο πλακίδιο του ταμπλό όσο το δυνατόν γρηγορότερα και με το μεγαλύτερο δυνατό σκορ.

Εργασία A – Random Movement (0,75 βαθμοί)

Στην παρούσα εργασία υπάρχουν 2 βασικά ζητούμενα.

1. Δημιουργία ενός ταμπλό μεγέθους NxM με ορισμένο αριθμό φιδιών, σκαλών και μήλων, τυχαία καταναμεμμένα στο χώρο.
2. Δημιουργία παίκτη που παίζει με τη χρήση ζαριού (τυχαία κίνηση).

Στη συνέχεια παρουσιάζονται οι κύριες κλάσεις του παιχνιδιού που σας ζητείται να υλοποιήσετε.

Κλάση Snake

Η κλάση **Snake** θα αντιπροσωπεύει το φίδι και θα έχει τις εξής μεταβλητές:

- i. **int snakeId**: το id του φιδιού.
- ii. **int headId**: το id του πλακιδίου του ταμπλό όπου βρίσκεται το κεφάλι του φιδιού.
- iii. **int TailId**: το id του πλακιδίου του ταμπλό όπου βρίσκεται η ουρά του φιδιού.

Οι συναρτήσεις που πρέπει να υλοποιήσετε είναι οι εξής:

- a. Οι **constructors** της κλάσης. Μεταξύ άλλων, θα σας φανεί χρήσιμο να φτιάξετε και έναν constructor που να δέχεται σαν όρισμα ένα αντικείμενο τύπου **Snake**.
- b. Όλες οι συναρτήσεις **get** και **set** για τις μεταβλητές της κλάσης.

Κλάση Ladder

Η κλάση **Ladder** θα αντιπροσωπεύει τη σκάλα και θα έχει τις εξής μεταβλητές:

- i. **int ladderId**: το id της σκάλας.
- ii. **int upStepId**: το id του πλακιδίου του ταμπλό όπου βρίσκεται η βάση της σκάλας.
- iii. **int downStepId**: το id του πλακιδίου του ταμπλό όπου βρίσκεται η κορυφή της σκάλας.
- iv. **boolean broken**: δείχνει αν η σκάλα είναι σπασμένη ή όχι.

Οι συναρτήσεις που πρέπει να υλοποιήσετε είναι οι εξής:

- a. Οι **constructors** της κλάσης. Μεταξύ άλλων, θα σας φανεί χρήσιμο να φτιάξετε και έναν constructor που να δέχεται σαν όρισμα ένα αντικείμενο τύπου **Ladder**.
- b. Όλες οι συναρτήσεις **get** και **set** για τις μεταβλητές της κλάσης.

Κλάση Apple

Η κλάση **Apple** θα αντιπροσωπεύει το μήλο και θα έχει τις εξής μεταβλητές:

- i. **int appleId**: το id του μήλου.
- ii. **int appleTileId**: το id του πλακιδίου του ταμπλό όπου βρίσκεται το μήλο.
- iii. **String color**: το χρώμα του μήλου (red ή black).
- iv. **int points**: οι πόντοι που θα κερδίζει ή θα χάνει ένας παίκτης (ανάλογα με το χρώμα του μήλου) όταν θα τρώει το μήλο.

Οι συναρτήσεις που πρέπει να υλοποιήσετε είναι οι εξής:

- a. Οι **constructors** της κλάσης. Μεταξύ άλλων, θα σας φανεί χρήσιμο να φτιάξετε και έναν constructor που να δέχεται σαν όρισμα ένα αντικείμενο τύπου Apple.
- b. Όλες οι συναρτήσεις **get** και **set** για τις μεταβλητές της κλάσης.

Κλάση Board

Η κλάση **Board** θα αντιπροσωπεύει το ταμπλό του παιχνιδιού και θα έχει τις εξής μεταβλητές:

- i. **int N, M**: οι διαστάσεις του ταμπλό NxM
- ii. **int[][] tiles**: ένα δισδιάστατος πίνακας NxM που περιέχει τα ids των πλακιδίων του ταμπλό. Τα ids παίρνουν τιμές από το 1 έως το NxM με τρόπο αντιστοιχο με αυτόν που φαίνεται στην Εικόνα 1.
- iii. **Snake[] snakes**: ένας πίνακας με αντικείμενα τύπου Snake.
- iv. **Ladder[] ladders**: ένας πίνακας με αντικείμενα τύπου Ladder.
- v. **Apple[] apples**: ένας πίνακας με αντικείμενα τύπου Apple.

Οι συναρτήσεις που πρέπει να υλοποιήσετε είναι οι εξής:

- a. Ο κενός **constructor** της κλάσης.
- b. Ο **2ος constructor** της κλάσης, ο οποίος θα πρέπει να δέχεται ως ορίσματα τις διαστάσεις του ταμπλό (N, M), τον αριθμό των φιδιών, τον αριθμό των σκαλών και τον αριθμό των μήλων που θα περιέχει το ταμπλό.
- c. Ο **3ος constructor** της κλάσης, ο οποίος θα δέχεται σαν όρισμα ένα αντικείμενο τύπου Board.
- d. Όλες οι συναρτήσεις **get** και **set** για τις μεταβλητές της κλάσης.
- e. Συνάρτηση **void createBoard()**: Η συνάρτηση αυτή θα δημιουργεί το ταμπλό του παιχνιδιού με τυχαίο τρόπο. Ποιο συγκεκριμένα, η συνάρτηση θα αρχικοποιεί τα αντικείμενα των πινάκων snakes, ladders και apples με τυχαίο τρόπο και τον πίνακα tiles με τα ids των πλακιδίων.

Προσοχή!!!

- Οι επιλογές τοποθέτησης του κάθε φιδιού και της κάθε σκάλας θα γίνεται τυχαία αλλά θα πρέπει να λάβετε υπόψη ότι το κεφάλι του φιδιού θα είναι πάντα σε ψηλότερο επίπεδο από την ουρά. Αντίστοιχοι περιορισμοί θα ισχύουν και για τις σκάλες.

- Τα μήλα τοποθετούνται σε τυχαίες θέσεις πάνω στο ταμπλό, με μοναδικό περιορισμό να μη βρίσκονται σε πλακίδιο όπου υπάρχει κεφάλι φιδιού.
- f. Συνάρτηση **void createElementBoard()**: Η συνάρτηση αυτή θα δημιουργεί και θα εκτυπώνει τρεις πίνακες `String[][] elementBoardSnakes`, `String[][] elementBoardLadders`, `String[][] elementBoardApples`, ίδιων διαστάσεων με τον πίνακα `tiles`. Κάθε στοιχείο του κάθε πίνακα θα περιέχει το αλφαριθμητικό που εκπροσωπεί το αντικείμενο που βρίσκεται πάνω στο αντίστοιχο πλακίδιο του ταμπλό, συνοδευόμενο από το `id` του (`snakeId`, `ladderId` και `appleId` αντίστοιχα). Σε περίπτωση που το πλακίδιο δεν περιέχει κάποιο αντικείμενο, το αντίστοιχο στοιχείο του πίνακα θα παίρνει σαν τιμή το αλφαριθμητικό “_”.
- Χρησιμοποιήστε την ακόλουθη κωδικοποίηση αντικειμένων:
- Κεφάλι φιδιού: SH. Ουρά φιδιού: ST
 - Κάτω μέρος σκάλας: LD. Άνω μέρος σκάλας: LU
 - Μήλος κόκκινο: AR. Μήλο μαύρο: AB

Παράδειγμα πινάκων διάστασης 10x20:

| elementBoardSnakes | elementBoardLadders | elementBoardApples |
|---|---|---|
| <div>SH0</div> <div>SH1</div> <div>ST0</div> <div>ST1</div> <div>ST2</div> <div>SH2</div> | <div>LU0</div> <div>LD0</div> <div>LU1</div> <div>LD1</div> <div>LU2</div> <div>LD2</div> | <div>AB5</div> <div>AR3</div> <div>AR0</div> <div>AB1</div> <div>AR2</div> <div>AR4</div> |

Κλάση Player

Η κλάση **Player** θα αντιπροσωπεύει τον παίκτη και θα έχει τις εξής μεταβλητές:

- int playerId**: ο κωδικός του παίκτη.
- String name**: το όνομα του παίκτη.
- int score**: το σκορ του παίκτη, το οποίο θα καθορίζεται από τους πόντους που του δίνουν τα μήλα που τρώει.
- Board board**: το ταμπλό του παιχνιδιού.

Οι συναρτήσεις που πρέπει να υλοποιήσετε είναι οι εξής:

- Οι **constructors** της κλάσης.
- Όλες οι συναρτήσεις **get** και **set** για τις μεταβλητές της κλάσης.

- c. Συνάρτηση **int[] move(int id, int die)**: Η συνάρτηση αυτή θα δέχεται σαν όρισμα το id του πλακιδίου όπου βρίσκεται ο παίκτης πριν την κίνηση και τον αριθμό του ζαριού και θα επιστρέφει έναν πίνακα ακεραίων που θα περιλαμβάνει το id του αντίστοιχου πλακιδίου μετά την κίνηση, τον αριθμό των φιδιών από τα οποία τσιμπήθηκε, τον αριθμό των σκαλών που χρησιμοποίησε, τον αριθμό των κόκκινων και μαύρων μήλων που έφαγε ο παίκτης κατά τη συγκεκριμένη κίνησή του. Επίσης, θα εκτυπώνει στην κονσόλα αντίστοιχο μήνυμα κάθε φορά που ένας παίκτης δαγκώνεται από φίδι, ανεβαίνει κάποια σκάλα ή τρώει ένα μήλο.

Προσοχή!!!

- Υπάρχει πιθανότητα ο παίκτης σε ένα γύρο να χρειαστεί να κάνει παραπάνω από μία κινήσεις. Σκεφτείτε για παράδειγμα την περίπτωση που ο παίκτης πέφτει πάνω σε φίδι και κατεβαίνει στην ουρά του, αλλά στο πλακίδιο όπου κατέβηκε υπάρχει και σκάλα την οποία θα πρέπει να ανέβει, κ.ο.κ.
- Όταν ένας παίκτης τρώει ένα μήλο, τότε αυτό εξαφανίζεται από το ταμπλό. Για δική σας ευκολία, θα μπορούσατε απλά να μηδενίζεται τους πόντους του μήλου.
- Η κάθε σκάλα μπορεί να χρησιμοποιηθεί μόνο μια φορά κατά τη διάρκεια του παιχνιδιού. Αυτό σημαίνει ότι όταν ένας παίκτης ανέβει μια σκάλα, στη συνέχεια η σκάλα σπάει και δε μπορεί να ξαναχρησιμοποιηθεί.

Κλάση Game

Η κλάση **Game** θα αντιπροσωπεύει το παιχνίδι και θα έχει τις εξής μεταβλητές:

- i. **int round**: ο τρέχον γύρος του παιχνιδιού.

Οι συναρτήσεις που πρέπει να υλοποιήσετε είναι οι εξής:

- a. Οι **constructors** της κλάσης.
- b. Όλες οι συναρτήσεις **get** και **set** για τη μεταβλητή της κλάσης.
- c. Συνάρτηση **public static void main()**: συνάρτηση εκκίνησης του παιχνιδιού. Στη συνάρτηση αυτή θα πρέπει να γίνεται μια ακολουθία ενεργειών:
 - Δημιουργία ταμπλό διάστασης 20x10, με 3 φίδια, 3 σκάλες και 6 μήλα.
 - Ορισμός 2 παικτών.
 - Οι παίκτες παίζουν εναλλάξ (ρίχνουν το ζάρι και κινούνται στο ταμπλό) μέχρις ότου ένας από τους 2 τερματίσει.
 - Εκτύπωση του αριθμού των γύρων παιχνιδιού που έπαιξαν οι παίκτες, του σκορ του κάθε παίκτη κατά τον τερματισμό και το νικητή του παιχνιδιού.

ΠΡΟΣΟΧΗ!!! Αν θέλετε να διαβάσετε τις τιμές των μεταβλητών ενός αντικειμένου μιας κλάσης ή να θέσετε τιμές στις μεταβλητές θα πρέπει να χρησιμοποιήσετε τους αντίστοιχους **getters** και **setters**.

Οδηγίες

Τα προγράμματα θα πρέπει να υλοποιηθούν σε Java, με πλήρη τεκμηρίωση του κώδικα. Το πρόγραμμά σας πρέπει να περιέχει επικεφαλίδα σε μορφή σχολίων με τα στοιχεία σας (ονοματεπώνυμο, ΑΕΜ, τηλέφωνα και ηλεκτρονικές διευθύνσεις). Επίσης, πριν από κάθε κλάση ή μέθοδο θα υπάρχει επικεφαλίδα σε μορφή σχολίων με σύντομη περιγραφή της λειτουργικότητας του κώδικα. Στην περίπτωση των μεθόδων, πρέπει να περιγράφονται και οι μεταβλητές τους.

Οι εργασίες που περιέχουν λάθη μεταγλώττισης θα μηδενίζονται αυτομάτως.

Είναι δική σας ευθύνη η απόδειξη καλής λειτουργίας του προγράμματος.

Παραδοτέα:

- 1. Ηλεκτρονική αναφορά** που θα περιέχει: εξώφυλλο, περιγραφή του προβλήματος, του αλγορίθμου και των διαδικασιών που υλοποιήσατε και τυχόν ανάλυσή τους. Σε καμία περίπτωση να μην αντιγράφεται ολόκληρος ο κώδικας μέσα στην αναφορά (εννοείται ότι εξαιρούνται τμήματα κώδικα τα οποία έχουν ως στόχο τη διευκρίνιση του αλγορίθμου).
Προσοχή: Ορθογραφικά και συντακτικά λάθη πληρώνονται.
- 2. Ένα αρχείο σε μορφή .zip με όνομα “AEM1_AEM2_PartA.zip”,** το οποίο θα περιέχει **όλο** το project σας στον eclipse καθώς και το αρχείο της γραπτής αναφοράς σε pdf (**αυστηρά**). Το αρχείο .zip θα γίνεται upload στο site του μαθήματος **στην ενότητα των ομαδικών εργασιών και μόνο**. Τα ονόματα των αρχείων πρέπει να είναι με **λατινικούς χαρακτήρες**.

Προθεσμία υποβολής:

Κώδικας και αναφορά **Δευτέρα 12 Νοεμβρίου, 23:59** (ηλεκτρονικά)
Δε θα υπάρξει καμία παρέκκλιση από την παραπάνω προθεσμία.