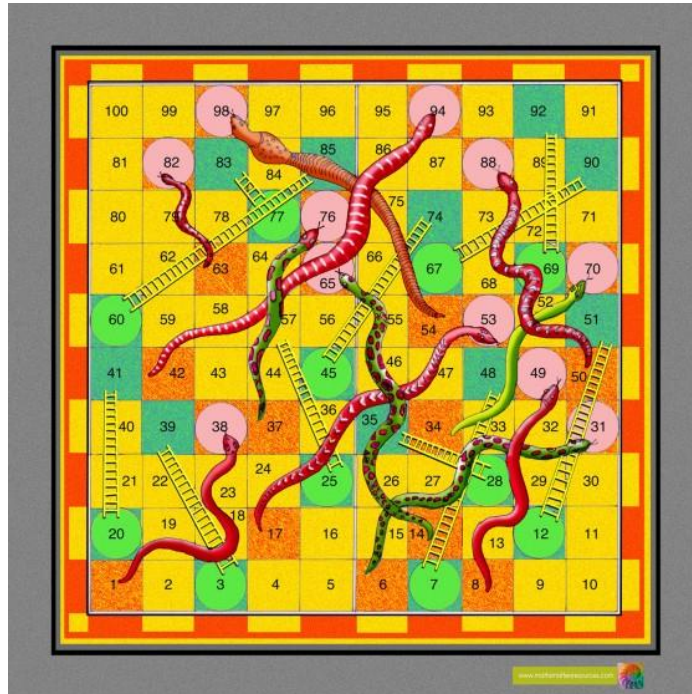


Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών
Υπολογιστών
Δομές Δεδομένων

3ο Εξάμηνο Σπουδών

Εργασία Α – Random Movement



Μία εργασία των φοιτητών:

Κωφοκώτσιου Ηλία με ΑΕΜ: 9380

Σκαπέτη Χρήστου με ΑΕΜ: 9378

Χειμερινό Εξάμηνο Ακαδημαϊκού Έτους 2018 - 2019

Περιγραφή του Προβλήματος

Στην εργασία αυτή, ζητήθηκε από τους φοιτητές να υλοποιήσουν το γνωστό επιτραπέζιο παιχνίδι «Φιδάκι» με μικρές αλλαγές στους κανονισμούς. Στα πλαίσια αυτής της υλοποίησης, ζητήθηκε από τους φοιτητές να γράψουν σε γλώσσα προγραμματισμού Java έξι διαφορετικές κλάσεις που θα αποτελούν την υλοποίηση του παιχνιδιού (κλάση Game), του ταμπλό (κλάση Board), των παικτών (κλάση Player), των φιδιών (κλάση Snake), των σκαλών (κλάση Ladder) και των μήλων (κλάση Apple).

Οι οδηγίες που περιέχονται στην εκφώνηση σκιαγραφούσαν κάποιες υποχρεωτικές μεταβλητές για κάθε κλάση, καθώς και τις απαιτούμενες μεθόδους που θα χρειαζόταν να υλοποιηθούν σε κάθε μια από τις κλάσεις αυτές. Εκτός από τις συναρτήσεις αρχικών συνθηκών (constructors) και τις συναρτήσεις απόδοσης και επιστροφής τιμής μεταβλητής (setters και getters), που ήταν απαραίτητες σε όλες τις κλάσεις, επιπλέον ζητήθηκε ένας αριθμός από επιπλέον συναρτήσεις. Αυτές ήταν: η συνάρτηση move της κλάσης Player, οι συναρτήσεις createBoard και createElementBoard της κλάσης Board και τέλος, η συνάρτηση main, εντός της κλάσης Game.

Τέλος, το σύνολο του προγράμματος αυτού θα πρέπει να μπορεί να προσομοιώνει ένα παιχνίδι δυο παικτών, στο οποίο θα παίζουν εναλλάξ μέχρι ένας από τους δυο να φτάσει στον τερματισμό. Επιπλέον, ζητήθηκε να εμφανίζονται οι γύροι που χρειάστηκαν μέχρι ένας από τους δυο παίκτες να νικήσει, καθώς και το σύνολο των πόντων που μάζεψε ο κάθε παίκτης κατά την διάρκεια του παιχνιδιού.

Επιπλέον, από την εκφώνηση δόθηκαν συγκεκριμένες παράμετροι για το παιχνίδι. Συγκεκριμένα, δόθηκε το μέγεθος του ταμπλό, ίσο με 20 x 10 πλακίδια, ο αριθμός των παικτών, ίσος με 2, καθώς και ο αριθμός των φιδιών, σκαλών και μήλων, ίσος με 3, 3, και 6 αντίστοιχα.

Περιγραφή του Αλγορίθμου

Ακολουθεί μια σύντομη ανάλυση των αλγορίθμων που χρησιμοποιήθηκαν κατά την υλοποίηση του προβλήματος.

Είναι σημαντικό να σημειωθεί ότι οι κλάσεις Snake, Apple και Ladder κρίθηκαν πολύ απλές και ο τρόπος υλοποίησης τους αρκετά προφανής. Για αυτόν τον λόγο δεν θα αναλυθούν σε αυτήν την αναφορά.

Αντίθετα, οι κλάσεις Board, Game και Player, όντας πιο πολύπλοκες, θα αναλυθούν παρακάτω. Πιο συγκεκριμένα, θα παραλειφθούν από την ανάλυση οι constructors, getters και setters, των οποίων υπάρχει επαρκής σχολιασμός εντός των αρχείων .java του project, ενώ θα γίνει μια προσπάθεια να εξηγηθούν οι συναρτήσεις createBoard και createElementBoard της κλάσης Board, move της κλάσης Player και main, της κλάσης Game.

Κλάση Board

Συνάρτηση createBoard()

Η συνάρτηση αυτή είναι υπεύθυνη για την τοποθέτηση των προκαθορισμένων φιδιών, σκαλών και μήλων στο ταμπλό του παιχνιδιού, με τους λογικούς περιορισμούς που διέπουν αυτήν την διαδικασία

Αρχικά ορίζονται τρεις πίνακες τύπου int (taken, headTaken και upStepTaken) στους οποίους αποθηκεύονται οι θέσεις οι οποίες θεωρήθηκαν σημαντικές. Στην headTaken αποθηκεύονται οι θέσεις όπου τοποθετούνται τα κεφάλια των φιδιών, στην upStepTaken αποθηκεύονται οι θέσεις όπου τοποθετούνται οι βάσεις των σκαλών και στην taken αποθηκεύονται και τα δυο προηγούμενα στοιχεία.

Στον πρώτο βρόχο for αρχικά ορίζεται το id του κάθε φιδιού. Στην συνέχεια παράγεται, εντός επανάληψης, κάποιος ψευδοτυχαίος αριθμός στο (μαθηματικό) διάστημα [2,numTiles) όπου numTiles ο αριθμός των tiles του ταμπλό. Αυτός ο αριθμός αποδίδεται στην κεφαλί του φιδιού έπειτα από έλεγχο για την μη ταύτιση της θέσης του με άλλο κεφάλι (με την χρήση της μεταβλητής check, τύπου boolean, και τα δεδομένα του πίνακα headTaken). Εάν αυτός ο έλεγχος δείξει ότι υπάρχει άλλο κεφάλι φιδιού στο πλακίδιο αυτό, τότε παράγεται νέος ψευδοτυχαίος αριθμός.

Για την τοποθέτηση των ουρών των φιδιών, καθώς και των κορυφών των σκαλών, παράγεται και πάλι, εντός επανάληψης, ψευδοτυχαίος αριθμός, με συνθήκη της επανάληψης αυτήν την φορά να είναι πως η ουρά του φιδιού βρίσκεται κάτω από το κεφάλι.

Εργασία A – Random Movement

Ομοίως, ακολουθείται παρόμοια διαδικασία για τις βάσεις των σκαλών και τα μήλα ενώ με αντίστοιχη μέθοδο με αυτή της ουράς ενός φιδιού, τοποθετούνται και οι κορυφές των σκαλών, με προϋπόθεση η κορυφή της σκάλας να βρίσκεται πάνω από την βάση της (το οποίο επιτυγχάνεται με την χρήση του πίνακα `upStepTaken`).

Επιπλέον, φροντίζεται ώστε να μην υπάρχουν παραπάνω από μία βάση σκάλας ή ένα κεφάλι φιδιού ανά πλακίδιο (`tile`) ώστε να μην υπάρχει διένεξη ως προς την επιλογή του φιδιού ή της σκάλας (με την χρήση του πίνακα `taken`). Ακόμη, στα μήλα δίνεται τυχαία τιμή από το 1 μέχρι το 10 στην μεταβλητή `score`, δηλαδή στους πόντους που προσθέτει ή αφαιρεί το μήλο. Αυτός ο παράγοντας καθορίζεται από μια τυχαία επιλογή του χρώματος του μήλου (κόκκινο ή μαύρο αντίστοιχα)

Τέλος, αντιστοιχούνται οι συντεταγμένες στον αριθμό του κάθε στοιχείου του πίνακα `tiles` όπως θα φαίνονται στο ταμπλό. Ουσιαστικά, ξεκινώντας από την κάτω αριστερή γωνία του ταμπλό, με τον αριθμό 1, και αυξάνοντας κατά ένα κάθε φορά, απαριθμούμε τα πλακίδια πηγαίνοντας εναλλάξ προς τα δεξιά ή προς τα αριστερά, αλλάζοντας την φορά προσπέλασης των στοιχείων του πίνακα κάθε φορά που φτάνουμε στο τέλος μιας γραμμής και κινούμενοι πάντα προς τα επάνω.

Συνάρτηση `createElementBoard()`

Η συνάρτηση αυτή είναι υπεύθυνη για την εκτύπωση των τριών πινάκων `string`, `elementBoardSnakes`, `elementBoardLadders` και `elementBoardApples` που εκφράζουν την θέση των φιδιών, σκαλών και μήλων αντίστοιχα που υπάρχουν στο ταμπλό. Εκεί όπου υπάρχει κάποιο αντικείμενο, θα εμφανίζεται σχετικός κωδικός, όπως καθορίζεται από την εκφώνηση, ενώ σε κάθε άλλη θέση θα εμφανίζεται το αλφαριθμητικό “___”.

Στην συνέχεια, για κάθε φίδι που περιέχεται στον πίνακα `snakes` του `Board`, αποθηκεύονται προσωρινά οι αριθμοί των πλακιδίων όπου βρίσκονται το κεφάλι (`headId`) και η ουρά (`tailId`). Στην συνέχεια, οι αριθμοί αυτοί, που είναι οι πραγματικοί αριθμοί του ταμπλό μετασχηματίζονται στο ζεύγος συντεταγμένων που αντιστοιχούν στην θέση αυτή στον πίνακα `tiles`. Έπειτα, με χρήση αυτών των συντεταγμένων τοποθετείται το κατάλληλο αλφαριθμητικό στην αντίστοιχη θέση του πίνακα `elementBoardSnakes`.

Με την ίδια διαδικασία, τοποθετούνται στον πίνακα `elementBoardLadders` οι σκάλες και στον πίνακα `elementBoardApples` τα μήλα.

Τέλος, οι πίνακες αυτοί εκτυπώνονται στην κονσόλα, σε διάταξη ο ένας δίπλα στον άλλο.

Κλάση Player

Συνάρτηση move(int id, int die)

Η συνάρτηση αυτή είναι η υλοποίηση της κίνησης του παίκτη στο ταμπλό του παιχνιδιού. Παίρνει δυο ορίσματα, την θέση του παίκτη και το αποτέλεσμα του ζαριού. Επιπλέον, έχει ως επιστρεφόμενη τιμή έναν πίνακα int, πέντε θέσεων, ο οποίος θα περιέχει στα κελιά του την τελική θέση στην οποία καταλήγει ο παίκτης, τον αριθμό από τα φίδια από τα οποία τσιμπήθηκε, τον αριθμό από τις σκάλες τις οποίες χρησιμοποίησε, τον αριθμό των κόκκινων μήλων που έφαγε και τον αριθμό των μαύρων μήλων που έφαγε.

Αρχικά, δημιουργείται ένας πίνακας int, που θα χρησιμεύσει ως επιστροφή της συνάρτησης.

Υπολογίζεται το άθροισμα $id + die$ και αποθηκεύεται σε μια μεταβλητή int με το όνομα start. Εάν αυτό το άθροισμα είναι μεγαλύτερο από τον αριθμό του τελευταίου πλακιδίου, τότε, επιστρέφεται απευθείας ο πίνακας των 5 θέσεων, με πρώτο στοιχείο τον αριθμό του τελευταίου πλακιδίου και τα υπόλοιπα μηδενικά. Σημειώνεται ότι στην συνάρτηση createBoard() έχει χρησιμοποιηθεί με τέτοιο τρόπο η συνάρτηση παραγωγής ψευδοτυχαίων αριθμών έτσι ώστε να μην υπάρχει κανένα αντικείμενο στο τελευταίο πλακίδιο του ταμπλό, εκτός ίσως από κορυφή σκάλας.

Εάν το άθροισμα αυτό ($id + die$) είναι μικρότερο από τον αριθμό του τελευταίου ταμπλό, η έλεγχος εισέρχεται σε βρόχο do – while, με συνθήκη εξόδου το περιεχόμενο της μεταβλητής start να είναι το ίδιο με το id στο τέλος του βρόχου. Στο σημείο αυτό ελέγχεται η ύπαρξη μήλου στο πλακίδιο που βρίσκεται ο παίκτης, με την βοήθεια μιας βοηθητικής συνάρτησης που ονομάζεται appleExists και έχει σαν όρισμα το id του πλακιδίου όπου θα γίνει η αναζήτηση. Η συνάρτηση αυτή εάν βρει μήλο στην θέση με αριθμό id επιστρέφει το μήλο αυτό, αλλιώς επιστρέφει ένα κενό μήλο. Εάν το επιστρεφόμενο μήλο δεν είναι κενό, τότε η βαθμολογία του παίκτη μεταβάλλεται ανάλογα με το χρώμα και τους βαθμούς του μήλου.

Στην συνέχεια, με την βοήθεια δεύτερης βοηθητικής συνάρτησης με το όνομα snakeExists και όρισμα το id του πλακιδίου όπου γίνεται η αναζήτηση, ελέγχουμε την ύπαρξη κεφαλιού φιδιού στο πλακίδιο όπου βρίσκεται ο παίκτης. Εάν υπάρχει, τότε η συνάρτηση snakeExists επιστρέφει το φίδι αυτό, αλλιώς επιστρέφει κενό φίδι. Εάν το επιστρεφόμενο φίδι δεν είναι κενό, τότε ο παίκτης μεταφέρεται στην ουρά του φιδιού και μεταβάλλεται το id.

Ομοίως, αναζητούμε την ύπαρξη σκάλας στο πλακίδιο, εάν και μόνο αν δεν έχει βρεθεί φίδι στο πλακίδιο, με την βοήθεια τρίτης βοηθητικής συνάρτησης με το όνομα ladderExists, που λειτουργεί με τον ίδιο τρόπο με τις προηγούμενες. Στην

περίπτωση που υπάρχει, τότε ο παίκτης μετακινείται στην κορυφή της σκάλας και μεταβάλλεται το id.

Εάν το id είναι διαφορετικό από το start, τότε ο παίκτης βρίσκεται σε διαφορετικό πλακίδιο από αυτό που ξεκίνησε, οπότε χρειάζεται να ξαναγίνει ο έλεγχος για την ύπαρξη μήλου, φιδιού ή σκάλας. Αλλιώς, ο έλεγχος του προγράμματος βγαίνει από τον βρόχο και η συνάρτηση επιστρέφει τον πίνακα πέντε θέσεων με τις κατάλληλες πληροφορίες που περιγράφουν τον γύρο του παίκτη.

Κλάση Game

Συνάρτηση main():

Η συνάρτηση αυτή περιέχει όλη την υλοποίηση του παιχνιδιού. Δημιουργεί αρχικά ένα αντικείμενο της κλάσης Game, ένα αντικείμενο της κλάσης Board που είναι ο πίνακας του παιχνιδιού και έναν πίνακα δυο θέσεων, με αντικείμενα τύπου Player, τα οποία θα είναι οι παίκτες του παιχνιδιού. Ο πίνακας έχει, σύμφωνα με την εκφώνηση, διαστάσεις 20 x 10, 3 φίδια, 3 σκάλες και 6 μήλα. Καλείται η συνάρτηση createBoard() προκειμένου να τοποθετηθούν τα φίδια, οι σκάλες και τα μήλα στον πίνακα και στην συνέχεια εμφανίζεται το αποτέλεσμα στην κονσόλα, με την κλήση της συνάρτησης createElementBoard().

Επίσης, δημιουργείται ένας δισδιάστατος πίνακας 2 x 5, για να αποθηκεύονται κάθε φορά οι επιστρεφόμενες τιμές της συνάρτησης move του κάθε παίκτη.

Στην συνέχεια, ξεκινώντας από τον πρώτο γύρο και εντός επανάληψης, παίζουν εναλλάξ οι δυο παίκτες, με κλήση της συνάρτησης move του καθενός, με τα κατάλληλα ορίσματα. Κάθε φορά που παίζει κάποιος παίκτης εκτυπώνεται αντίστοιχο μήνυμα στην κονσόλα, ενώ η επαναληπτική διαδικασία αυτή τελειώνει όταν έστω ένας από τους δυο παίκτες φτάσει ή ξεπεράσει το τελευταίο πλακίδιο, δηλαδή το πλακίδιο 200.

Εάν και οι δυο παίκτες φτάσουν ή ξεπεράσουν το πλακίδιο 200 στον ίδιο γύρο, τότε νικητής είναι αυτός που έχει μαζέψει τους περισσότερους πόντους. Αλλιώς, εάν έχει φτάσει μόνο ένας, τότε αυτός είναι ο νικητής.

Σε κάθε περίπτωση, εκτυπώνονται το πλακίδιο που έφτασε ο κάθε παίκτης και η συνολική του βαθμολογία.