



11/11/2018

2

Υπεύθυνοι Φοιτητές :

Χάρης Φίλης

AEM : 9449

email : charisfilis@ece.auth.gr

Γρηγόρης Παυλάκης AEM : 9571

email: grigpavl@ece.auth.gr

ΤΟ ΠΡΟΒΛΗΜΑ

Το πρώτο μέρος της εργασίας περιλαμβάνει την υλοποίηση την υλοποίηση έξι κλάσεων (Board, Apple, Game, Player, Ladder, Snake), ώστε να στηθεί ένα τυπικό παιχνίδι-φιδάκι όπου οι δύο παίκτες παίζουν σε ένα ταμπλό 20x10, με 3 φίδια, 3 σκάλες, και 6 μήλα.

ΥΛΟΠΟΙΗΣΗ

Κλάση Snake

Η κλάση Snake έχει τις εξής ιδιωτικές μεταβλητές :

- `int snakeId` ~ ξεχωριστή ταυτότητα κάθε Snake
- `int headId` ~ ταυτότητα του πλακιδίου που βρίσκεται το κεφάλι
- `int tailId` ~ ταυτότητα του πλακιδίου που βρίσκεται στην ουρα του snake

Η κλάση διαθέτει επίσης τους βασικούς `getters` και `setters` που δίνουν πρόσβαση στις παραπάνω ιδιωτικές μεταβλητές. Τέλος έχει έναν `constructor` χωρίς ορίσματα, έναν με ορίσματα, και έναν με όρισμα ένα αντικείμενο Snake που μοιάζει με `copy constructor`.

Κλάση Ladder

Η κλάση Ladder έχει τις εξής ιδιωτικές μεταβλητές:

- `int ladderId` ~ ξεχωριστή ταυτότητα κάθε ladder
- `int upStepId` ~ ταυτότητα του πλακιδίου που βρίσκεται το επάνω μέρος της σκάλας
- `int downStepId` ~ ταυτότητα του πλακιδίου που βρίσκεται το κάτω μέρος της σκάλας

Η κλάση διαθέτει επίσης παρόμοιους `getters`, `setters`, `constructors` etc.

Κλάση Apple

Η κλάση apple διαθέτει τις εξής μεταβλητές :

- `int appleId` ~ ξεχωριστή ταυτότητα κάθε apple
- `int appleTileId` ~ ταυτότητα του πλακιδίου που βρίσκεται το μήλο
- `String color` ~ το χρώμα του μήλου
- `int Points` ~ οι πόντοι που δίνει το κάθε μήλο

Τα υπόλοιπα της κλάσης δεν διαφέρουν απο τις προηγούμενες

11/11/2018

3

Κλάση Player

Η κλάση *Player* έχει τις εξής ιδιωτικές μεταβλητές :

- *int playerId* ~ ο κωδικός – ταυτότητα του κάθε παίκτη
- *String name* ~ το όνομα του κάθε παίκτη
- *int score* ~ οι πόντοι κάθε παίκτη ανάλογα με τα μήλα που έχει φάει
- *Board board* ~ το ταμπλό που παίζει ο κάθε παίκτης

Η κλάση έχει τους κλασσικούς *getters*, *setters*, *constructors* και την συνάρτηση *int[] move (int id, int die)*

Επεξήγηση της συνάρτησης *int[] move(int id, int die)*

Η συνάρτηση αρχικά δημιουργεί έναν πίνακα πέντε θέσεων, *moveResults*. Τοποθετείται στο πρώτο κελί του πίνακα, το άθροισμα *id+die* που αντιστοιχεί στην θέση του παίκτη την συγκεκριμένη χρονική στιγμή. Στην συνέχεια, διατρέχεται ο πίνακας *board* και βρίσκεται το πλήθος των φιδιών στα οποία ο παίκτης έχει πέσει, το οποίο τοποθετείται στο δεύτερο κελί του πίνακα. Με τον ίδιο τρόπο βρίσκεται ο αριθμός των σκάλων που ανέβηκε και των μήλων κόκκινων οι μαύρων που έφαγε, τα οποία τοποθετούνται αντίστοιχα στην τρίτη, τεταρτη και πέμπτη θέση του πίνακα. Τέλος ο πίνακας επιστρέφεται από την συνάρτηση.

Κλάση Board

Η κλάση *board* περιλαμβάνει τις εξής ιδιωτικές μεταβλητές:

- *int N,M* ~ ο αριθμός των γραμμών και των στηλών του πίνακα
- *int[][] tiles* ~ ο πίνακας με τα πλακίδια που είναι η αρένα του παιχνιδιού
- *Snake[] snakes* ~ ο πίνακας των φιδιών
- *Ladder[] ladders* ~ ο πίνακας των σκαλών
- *Apple[] apples* ~ ο πίνακας των μήλων

Η κλάση έχει τις κλασσικές συναρτήσεις αρχικών συνθηκών, *getters*, *setters* και επιπλέον τις συναρτήσεις

- *void createBoard();*
- *void createElementBoard();*
- *void initialize();*

Επεξήγηση της συνάρτησης *void createBoard();*

Η συνάρτηση *createBoard* καλεί την συνάρτηση *initialize* για κάθε ένα από τους πίνακες *tiles, snakes, ladders, apples*. Αυτό το κάναμε για καλύτερη διαχείριση κώδικα και προσβασιμότητας των άλλων κλάσεων στην κλάση.

Επεξήγηση της συνάρτησης *private void initialize(int[][] tiles);*

Η συνάρτηση αρχικοποιεί τα νούμερα των πλακιδίων σύμφωνα με τις οδηγίες που δόθηκαν στην εκφώνηση της εργασίας (αύξηση του αριθμού του πλακιδίου με εναλλαγή καθώς ανεβαίνει ο αριθμός της γραμμής από κάτω προς τα πάνω).

Επεξήγηση της συνάρτησης *public void createElementBoard()*

Η συνάρτηση αυτή αρχικά, δημιουργεί τρεις πίνακες *elementBoardSnakes*, *elementBoardLadders*, *elementBoardApples*, διδιάστατους με διάσταση ίση με τον πίνακα *tiles*. Όλα τα στοιχεία των πινάκων αρχικοποιούνται με “_” και στην συνέχεια τοποθετείται όπου έχουμε *headId – SH*, etc., όπως ακριβώς ζητήθηκε στην εργασία. Τέλος οι τρεις πίνακες εκτυπώνονται με κατάλληλο τρόπο.

Κλάση Game – main Class

Η κλάση *Game* έχει ως στατική μετάβλητη την *static int round* γιατί η τιμή της δεν επηρεάζεται από τα αντικείμενα τύπου που δημιουργούνται. Επίσης, έχουμε και τους ανάλογους *constructors*, *getters*, *setters*. Η βασική συνάρτηση που υλοποιείται σε αυτό το κομμάτι κώδικα είναι η *main*.

Main

Στην *main* αρχικά, γίνεται αρχικοποίηση του ταμπλό με τα φιδιά τα μήλα τις σκάλες και το σωστό μέγεθος. Επίσης, δημιουργούνται δύο νέοι πάκτες *p1, p2*. Δημιουργούνται δύο νέοι πίνακες 5 θέσεων *p1Status*, *p2Status*, που αποθήκεύουν τα αποτελέσματα της συνάρτησης *player.move()*. Στην συνέχεια ακολουθεί η κύρια λούπα του παιχνιδιού με περιορισμό

ΑΝΑΦΟΡΑ ΠΡΩΤΗΣ ΕΡΓΑΣΙΑΣ ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ 2019 – SNAKEGAME

11/11/2018

να μην έχει τερματίσει κάποιος παίκτης. Στο παιχνίδι ρίχνονται ζαριές και επιπλέον οι παίκτες μετακινούνται αφού ρίξει ο κάθε παίκτης ζαριά ο γύρο τελειώνει και αυξάνεται ο μετρητής των γύρων που είναι βασική μεταβλητή της κλάσης. Τέλος γίνεται ο έλεγχος των πινάκων για το ποιός παίκτης έχει τερματίσει, και εκτυπώνεται το ανάλογο μήνυμα μαζί με τους πόντους που έχει συλέξει. Νικητής δεν ανδεικνύεται αυτός που έχει τους περισσότερους πόντους αλλά αυτός που τερμάτισε πρώτος. Στο τέλος εκτυπώνονται και οι γύροι που παίχτηκαν μέχρι το τέλος του παιχνιδιού κάτι που είναι τυχαίο εφόσον είναι τυχαίος ο τρόπος παραγωγής ζαριάς