

Εργασία 2

Άσκηση 1

Να υλοποιήσετε ένα πρόγραμμα το οποίο διαχειρίζεται τη γραμμή παραγωγής μιας εταιρίας και υλοποιεί την έννοια της εταιρίας (**Company**) και των προϊόντων (**Product**) που παράγει κάθε μία εταιρία.

Η κλάση **Product** αντιπροσωπεύει ένα προϊόν και έχει τα ακόλουθα πεδία και μεθόδους:

- Πεδίο price: αντιπροσωπεύει την τιμή του προϊόντος και είναι double
- Πεδίο name: αντιπροσωπεύει το όνομα του προϊόντος και είναι αλφαριθμητικό
- Πεδίο code: αντιπροσωπεύει τον κωδικό του προϊόντος και αποτελείται από 2 χαρακτήρες και 6 αριθμούς (είναι αλφαριθμητικό).
- Μέθοδος δομητή χωρίς ορίσματα, η οποία αρχικοποιεί τα πεδία της κλάσης.
- Μέθοδος δομητή με ορίσματα. Τα ορίσματα είναι όσα και τα πεδία. Ο δομητής θέτει αρχική τιμή στα πεδία της κλάσης με βάση τις τιμές των ορισμάτων.
- Μεθόδους get (), που επιστρέφουν τα private πεδία της κλάσης
- Μεθόδους set (), που αναθέτουν τιμές στα private πεδία της κλάσης.
- Μέθοδος toString(), που επιστρέφει τις τιμές όλων των πεδίων σε μορφή String.
- Μέθοδος readProduct(), η οποία διαβάζει από το πληκτρολόγιο τις πληροφορίες ενός προϊόντος και αναθέτει τις τιμές στα πεδία του αντικειμένου που καλεί τη μέθοδο.
 - public void readProduct()
- Μέθοδος compare(Product p), η οποία συγκρίνει δύο προϊόντα και ελέγχει ποιο από τα δύο είναι ακριβότερο με βάση τη τιμή.
 - public boolean compare(Product p)
- Να δημιουργήσετε οποιαδήποτε άλλη μέθοδο χρειάζεστε.

Η κλάση **Company** έχει τα ακόλουθα πεδία και μεθόδους:

- Πεδίο companyName (όνομα εταιρίας)
- Πεδίο products τύπου ArrayList, στο οποίο θα αποθηκεύονται τα προϊόντα κάθε εταιρίας.
- Constructor χωρίς ορίσματα.
 - public Company()
- Constructor με ορίσματα, που θέτει αρχική τιμή στις μεταβλητές companyName και products.
 - public Company(String companyName, ArrayList products)
- Μέθοδος getNumOfProducts(), η οποία επιστρέφει τον αριθμό των προϊόντων.
 - public int getNumOfProducts()
- Set και get μεθόδους για όλα τα πεδία.
- Μέθοδος toString(), η οποία επιστρέφει τα δεδομένα ενός αντικειμένου Company στη μορφή ενός string.

- Να δημιουργήσετε οποιαδήποτε άλλη μέθοδο χρειάζεστε.

Να δημιουργήσετε μία κλάση **ProductManagement** η οποία θα περιέχει μία μέθοδο **main** και μία μέθοδο **search**. Η μέθοδος main θα υλοποιεί τα παρακάτω

1. Να δημιουργήσετε αντικείμενα της κλάσης Company διαβάζοντας τα δεδομένα από το πληκτρολόγιο. Ο χρήστης θα πρέπει για κάθε εταιρία να δίνει το όνομα της εταιρίας και τα προϊόντα που παράγει. Για κάθε προϊόν ο χρήστης θα πρέπει να δίνει το όνομα, το κωδικό και την τιμή. Ο χρήστης σταματάει να παρέχει δεδομένα όταν δώσει το αλφαριθμητικό "exit" ως όνομα της εταιρίας.
2. Να γίνεται έλεγχος για αρνητικές τιμές και κενές τιμές και εφόσον είναι αρνητικές ή κενές (π.χ. έχει πατήσει απλώς space ή tab) να αγνοούνται και ο χρήστης να ενημερώνεται ώστε να θέσει άλλη τιμή. Επίσης, να γίνεται έλεγχος για τον κωδικό εάν πληροί τα κριτήρια που έχουν τεθεί.
3. Να εμφανίσετε μήνυμα σχετικό με τον αριθμό προϊόντων που παράγει κάθε εταιρία.
4. Να εμφανίσετε μήνυμα σχετικά με το ποιο προϊόν είναι το ακριβότερο μεταξύ των προϊόντων όλων των εταιριών.
5. Να ζητήσετε από το χρήστη να δώσει ένα όνομα προϊόντος και να εμφανίσετε εάν υπάρχει. Ο χρήστης μπορεί να δίνει όσα ονόματα προϊόντων επιθυμεί. Η αναζήτηση προϊόντων σταματάει όταν ο χρήστης δώσει το αλφαριθμητικό quit.

Να υλοποιήσετε μία μέθοδο αναζήτησης η οποία θα δέχεται ένα όνομα προϊόντος ως παράμετρο και θα επιστρέφει εάν έχει βρεθεί ή όχι μεταξύ όλων των προϊόντων όλων των εταιριών.

- public static boolean search(ArrayList al, String name)

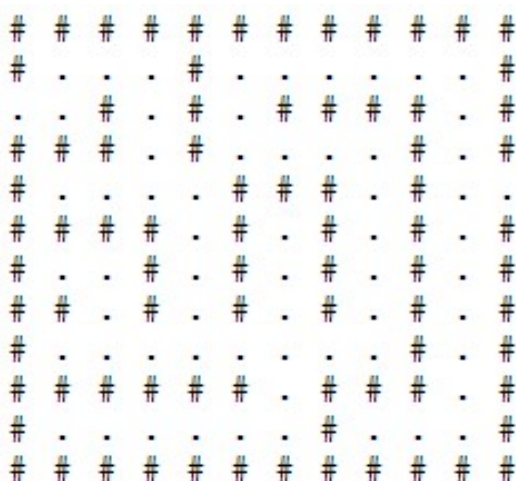
Άσκηση 2

Το πλέγμα από hashes (#) και τελείες (.) που απεικονίζεται στο παρακάτω σχήμα είναι ένας δισδιάστατος πίνακας που αναπαριστά έναν λαβύρινθο. Στο δισδιάστατο πίνακα, τα hashes (#) αναπαριστούν τους τοίχους του λαβύρινθου ενώ οι τελείες αναπαριστούν ελεύθερα μονοπάτια (σημεία που μπορεί να διασχίσει κάποιος) μέσα στο λαβύρινθο. Οι κινήσεις μπορούν να πραγματοποιηθούν μόνο σε θέσεις του πίνακα που περιέχουν τελείες. Η θέση (0,0) του πίνακα βρίσκεται στη πάνω αριστερά γωνία.

Υπάρχει ένας απλός αλγόριθμος για να μπορέσει κάποιος να διασχίσει ένα λαβύρινθο ο οποίος εγγυάται την εύρεση της εξόδου (θεωρώντας ότι υπάρχει μια έξοδος). Εάν δεν υπάρχει έξοδος, ο χρήστης θα κατευθυνθεί πάλι στην αρχική τοποθεσία εισόδου.

Αλγόριθμος διάσχησης λαβύρινθου:

Τοποθέτησε το δεξί σου χέρι στο τοίχο που βρίσκεται στα δεξιά σου και ξεκίνα να περπατάς. Μην απομακρύνεις ποτέ το χέρι σου από το τοίχο. Εάν ο λαβύρινθος στρίβει προς τα δεξιά, τότε θα ακολουθήσεις τον τοίχο προς τα δεξιά. Όσο δεν απομακρύνεις το χέρι σου από το τοίχο, κάποια στιγμή θα φτάσεις στην έξοδο του λαβύρινθου.



Εικόνα 1: Ο λαβύρινθος

Μπορεί να υπάρχει ένα πιο σύντομο μονοπάτι για να εξέλθετε από το λαβύρινθο αλλά ο αλγόριθμος εγγυάται να σας βγάλει από το λαβύρινθο αν τον ακολουθήσετε. Λοιπόν, ας δούμε ποια είναι η σημασία του αλγορίθμου με όρους υλοποίησης. Αρχικά θεωρούμε ότι η (0,0) θέση του λαβύρινθου/πίνακα βρίσκεται στην πάνω αριστερά γωνία. Εάν βρίσκεστε σε μια συγκεκριμένη θέση μέσα στον λαβύρινθο θα προσπαθήσετε να κινηθείτε σε μια νέα θέση με μια συγκεκριμένη σειρά. Ας θεωρήσουμε ότι εισέρχεστε στο λαβύρινθο από την αριστερή πλευρά. Στη συνέχεια θα προσπαθήσετε να βάλετε το χέρι σας στο δεξιό τοίχο. Αυτό σημαίνει ότι αρχικά θα προσπαθήσετε να πάτε προς τα κάτω (x+1,y), προς τα δεξιά (x, y+1), προς τα πάνω (x-1,y) και προς τα αριστερά (x, y-1). Για παράδειγμα, εάν υπάρχει ένα # στο σημείο από κάτω σου (έτσι δεν μπορείς να κινηθείς προς τα κάτω), προσπάθησε να κινηθείς προς τα δεξιά. Εάν και αυτό αποτύχει προσπάθησε να κινηθείς στην από πάνω θέση. Τέλος, εάν και αυτό αποτύχει προσπάθησε να κινηθείς προς τα αριστερά. Η κατεύθυνση και η κίνηση που

πρέπει να ακολουθήσει κάποιος για να βγει από το λαβύρινθο θα πρέπει να διέπετε από τα παρακάτω:

- Εάν κινείσαι προς τα κάτω (DOWN) σε ένα τετράγωνο, θα κοιτάξεις πρώτα για το τοίχο στα αριστερά (LEFT)
- Εάν κινείσαι προς τα δεξιά (RIGHT) σε ένα τετράγωνο, θα κοιτάξεις πρώτα για το τοίχο προς τα κάτω (DOWN)
- Εάν κινείσαι προς τα πάνω (UP) σε ένα τετράγωνο, θα κοιτάξεις πρώτα για το τοίχο στα δεξιά (RIGHT)
- Εάν κινείσαι προς τα αριστερά (LEFT) σε ένα τετράγωνο, θα κοιτάξεις πρώτα για το τοίχο προς τα πάνω (UP).

Να γράψετε ένα πρόγραμμα Java για να διασχίσετε το λαβύρινθο. Οι συναρτήσεις που χρειάζεται να γράψετε είναι:

- void traverse(char[][12], int, int, int)
Αναδρομική συνάρτηση που δέχεται ως ορίσματα το δισδιάστατο πίνακα, την τρέχουσα θέση x, τη τρέχουσα θέση y και τη κατεύθυνση που θέλεις να προσπαθήσεις πρώτα για την επόμενη κίνηση. Η συνάρτηση (1) θα τοποθετεί στο πίνακα ένα 'x' στο σημείο που βρίσκεστε, (2) θα τυπώνει το τρέχον λαβύρινθο ύστερα από κάθε κίνηση, (3) θα αποφασίζει εάν βγήκατε από το λαβύρινθο, (4) θα ελέγχει εάν η τρέχουσα θέση είναι η αρχική θέση (η επιστροφή πίσω στην αρχική θέση σημαίνει ότι ο λαβύρινθος δεν έχει άλλη έξοδο), (5) θα αποφασίζει για την επόμενη *έγκυρη* κίνηση. Μία έγκυρη κίνηση σημαίνει ότι θα κινηθείτε σε μια γειτονική θέση του πίνακα όπου δεν υπάρχει #.
- bool validMove(char[][12], int, int)
Συνάρτηση που δέχεται ως ορίσματα το δισδιάστατο πίνακα, τη θέση x και τη θέση y και επιστρέφει εάν είναι επιτρεπτή η κίνηση σε αυτή τη θέση.
- void printMaze(char[][12])
Συνάρτηση που δέχεται ως ορίσματα το δισδιάστατο πίνακα και εμφανίζει το λαβύρινθο
- bool isSolved(int, int)
Συνάρτηση που δέχεται ως ορίσματα τη θέση x και τη θέση y και επιστρέφει true ή false εάν έχει λυθεί ο λαβύρινθος

Χρήσιμες Πληροφορίες:

- Η εργασία είναι ομαδική των 2 ατόμων
- Η αντιγραφή ανιχνεύεται και τιμωρείται.
- **Παράδοση μέσω eclass. Upload θα κάνει το ένα από 2 άτομα της ομάδας.**
- Το όνομα του αρχείου (σε περίπτωση πολλών αρχείων θα φτιάξετε ένα zip αρχείο που θα τα περικλείει) που θα παραδώσετε θα έχει το παρακάτω format:
 - Επίθετο1_Όνομα1_ΑριθμόςΜητρώου1_Επίθετο2_Όνομα2_ΑριθμόςΜητρώου2
- Η εργασία θα πρέπει να παραδοθεί μέχρι τις 27/04/2023 και ώρα 23:59.