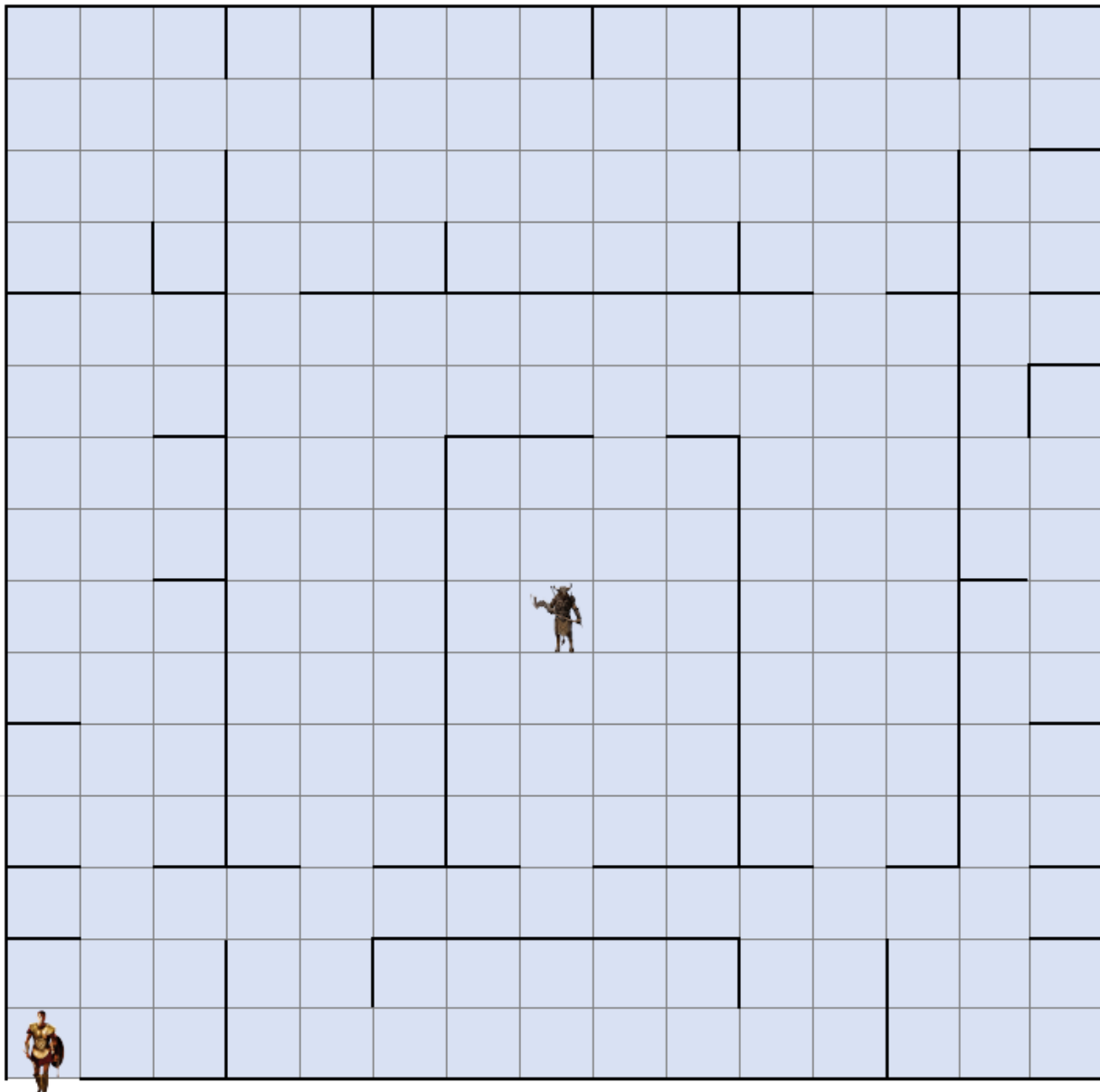


ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Λαβύρινθος

Ο Θησέας και ο Μινώταυρος

Στην ταινία «Μια νύχτα στο μουσείο» η οποία διαδραματίζεται στο Μουσείο Φυσικής Ιστορίας στην Αμερική, τα εκθέματα αποκτούν ζωή για ένα βράδυ και συμμετέχουν σε διάφορες περιπέτειες.



Εικόνα 1: Παράδειγμα ταμπλό παιχνιδιού διάστασης 15x15.

Μια από τις περιπέτειες στις οποίες συμμετέχουν τα εκθέματα είναι αυτή του Θησέα και του Μινώταυρου.

Ο ατρόμητος Θησέας αποφασίζει να σκοτώσει το Μινώταυρο. Όταν αποβιβάζεται στο νησί, ο Θησέας συναντάει την Αριάδνη, την κόρη του Μίνωα. Η Αριάδνη για να τον βοηθήσει να δραπετεύσει του λέει «πρέπει να ανακαλύψεις τα διάσπαρτα λάφουρα μέσα στον λαβύρινθο. Μόλις τα μαζέψεις όλα μία καταπακτή θα ανοίξει και θα πέσει μέσα ο Μινώταυρος».

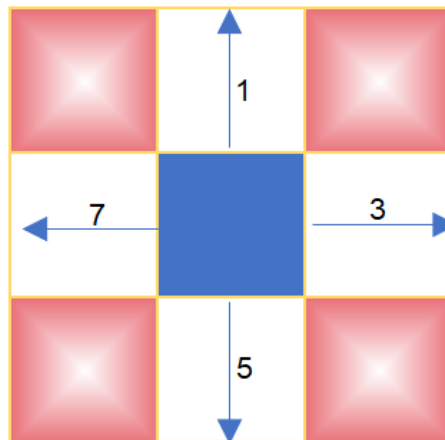
Θα καταφέρει ο Θησέας να ανακαλύψει όλα τα λάφουρα-εκθέματα πριν πέσει στα χέρια του Μινώταυρου;

Ταμπλό: Στην παρούσα εργασία, καλείστε να υλοποιήσετε τον λαβύρινθο που δημιουργείται από τα δωμάτια του μουσείου (Εικόνα 1), το ταμπλό σας θα είναι τετράγωνο ($N \times N$), και θα αποτελείται από πλακίδια. (N μονός αριθμός)

Παίκτης Θησέας: ο Θησέας βρίσκεται στην είσοδο του Λαβυρίνθου – θέση $(0,0)$, πλακίδιο : 0 , μετακινείται κατά μία θέση πάνω στο ταμπλό κάθε φορά, σύμφωνα με την Εικόνα 2.

Παίκτης Μινώταυρος: Ξεκινάει από το κεντρικό πλακίδιο του Λαβυρίνθου (Εικόνα 1) - θέση $(N \div 2, N \div 2)$, πλακίδιο $(N * N \div 2)$: . Μπορεί να κινείται κατά μία θέση πάνω στο ταμπλό κάθε φορά, σύμφωνα με την Εικόνα 2.

Λάφουρα: Τα λάφουρα (εκθέματα του μουσείου) είναι τοποθετημένα σε τυχαίες θέσεις πάνω στο ταμπλό και καθένα βρίσκεται πάνω σε ένα πλακίδιο.



Εικόνα 2 Διαθέσιμες κινήσεις Θησέα και Μινώταυρου

Στόχος του Θησέα είναι να βρει τα λάφουρα στα διάφορα σημεία του μουσείου χωρίς να τον εξοντώσει ο Μινώταυρος. Ο Θησέας χάνει την ζωή του αν βρεθεί στο ίδιο πλακίδιο με τον Μινώταυρο. Ο Θησέας έχει ένα φωτόσπαθο το οποίο φωτίζει το δρόμο του συνεχώς με ένα άσπρο φως. Όταν βρεθεί σε απόσταση 2 πλακιδίων από ένα λάφουρο το φωτόσπαθο γίνεται πράσινο. Αντίστοιχα όταν βρεθεί ο Θησέας σε απόσταση 2 πλακιδίων από τον Μινώταυρο το φωτόσπαθο γίνεται κόκκινο. Μόλις βρει ο Θησέας όλα

τα λάφουρα το παιχνίδι ολοκληρώνεται με νικητή τον Θησέα. Από την άλλη πλευρά, ο Μινώταυρος προσπαθεί να εντοπίσει το Θησέα και να τον σκοτώσει. Ο Μινώταυρος μπορεί να σκοτώσει το Θησέα εάν βρίσκονται στο ίδιο τετραγωνάκι.

Το παιχνίδι τερματίζει και στην περίπτωση που εκτελεστούν $2n$ βήματα και δεν έχει βρεθεί νικητής, το παιχνίδι θεωρείται ισόπαλο, καθώς έχει ξημερώσει και τα εκθέματα θα πρέπει να πέσουν πάλι σε λήθαργο.

Το σύστημα συντεταγμένων του παιχνιδιού βρίσκεται στο σημείο από το οποίο εκκινεί ο Θησέας και έχει την αρίθμηση που φαίνεται στην Εικόνα 3.

(4,0)	(4,1)	(4,2)	(4,3)	(4,4)
(3,0)	(3,1)	(3,2)	(3,3)	(3,4)
(2,0)	(2,1)	(2,2)	(2,3)	(2,4)
(1,0)	(1,1)	(1,2)	(1,3)	(1,4)
(0,0)	(0,1)	(0,2)	(0,3)	(0,4)

Εικόνα 3 Συντεταγμένες πλακιδίων ταμπλό διάστασης 5x5

20	21	22	23	24
15	16	17	18	19
10	11	12	13	14
5	6	7	8	9
0	1	2	3	4

Εικόνα 4 ID's πλακιδίων ταμπλό διάστασης 4x4

Εργασία A – Random Movement (0,75 βαθμοί)

Στην παρούσα εργασία υπάρχουν 2 βασικά ζητούμενα.

1. Δημιουργία ενός ταμπλό μεγέθους $N \times N$ με ορισμένο αριθμό εφοδίων και τειχών, τυχαία καταναμημένα στο χώρο.
2. Δημιουργία παικτών που παίζουν με τη χρήση ζαριού (τυχαία κίνηση). Σε κάθε γύρο, ο κάθε παίκτης ρίχνει το ζάρι και κινείται σύμφωνα τον αριθμό που επιστρέφεται. Το ζάρι επιστρέφει έναν ακέραιο αριθμό στο διάστημα $\{1,3,5,7\}$ το οποίο αντιπροσωπεύει την κίνηση που θα κάνει ο παίκτης, όπως περιγράφεται στην Εικόνα 2.

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

Κλάση Supply

Η κλάση **Supply** θα αντιπροσωπεύει τα εφόδια και θα έχει τις εξής μεταβλητές:

- i. **int supplyId**: το id του εφοδίου.
- ii. **int x**: η συντεταγμένη x του πλακιδίου του ταμπλό όπου βρίσκεται το εφόδιο.
- iii. **int y**: η συντεταγμένη y του πλακιδίου του ταμπλό όπου βρίσκεται το εφόδιο.
- iv. **int supplyTileId**: το id του πλακιδίου του ταμπλό όπου βρίσκεται το εφόδιο.

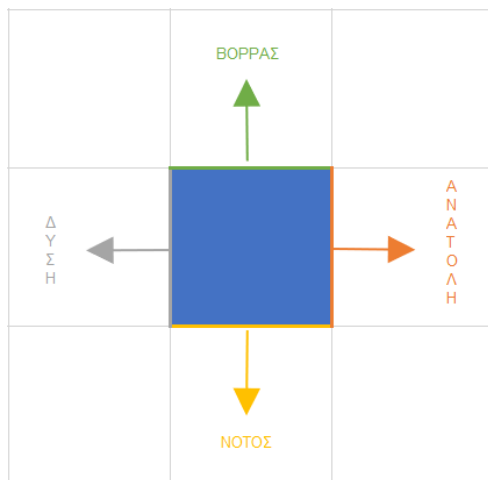
Οι συναρτήσεις που πρέπει να υλοποιήσετε είναι οι εξής:

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

Κλάση Tile

Η κλάση **Tile** θα αντιπροσωπεύει τα πλακίδια και θα έχει τις εξής μεταβλητές:

- i. **int tileId**: το id του πλακιδίου του ταμπλό.
- ii. **int x**: η συντεταγμένη x του πλακιδίου του ταμπλό.
- iii. **int y**: η συντεταγμένη y του πλακιδίου του ταμπλό.
- iv. **boolean up**: δείχνει εάν υπάρχει τοίχος στη βόρεια (πάνω) πλευρά του πλακιδίου.
- v. **boolean down**: δείχνει εάν υπάρχει τοίχος στην νότια (κάτω) πλευρά του πλακιδίου.
- vi. **boolean left**: δείχνει εάν υπάρχει τοίχος στη δυτική (αριστερή) πλευρά του πλακιδίου.
- vii. **boolean right**: δείχνει εάν υπάρχει τοίχος στην ανατολική (δεξιά) πλευρά του πλακιδίου.



Εικόνα 5 Κατευθύνσεις τειχών

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 `Tile`.
- 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:

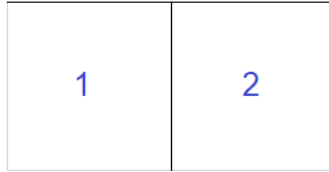
- i. **int N**: οι διαστάσεις του ταμπλό NxN
- ii. **int S**: ο αριθμός των εφοδίων που υπάρχουν στο ταμπλό.
- iii. **int W**: ο αριθμός των τειχών που μπορείτε να προσθέσετε στο λαβύρινθο.
- iv. **Tile[] tiles**: ένας πίνακας με αντικείμενα τύπου Tile. Τα ids παίρνουν τιμές από το 0 έως το NxN - 1 με τρόπο αντίστοιχο με αυτόν που φαίνεται στην Εικόνα 4
- v. **Supply[] supplies**: ένας πίνακας με αντικείμενα τύπου Supply.

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, N), τον αριθμό των εφοδίων και τον αριθμό των τειχών 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 createTile()**: Η συνάρτηση αυτή θα αρχικοποιεί τα αντικείμενα του πίνακα Tile με τυχαίο τρόπο.
- f. Συνάρτηση **void createSupply()**: Η συνάρτηση αυτή θα αρχικοποιεί τα αντικείμενα του πίνακα supplies με τυχαίο τρόπο.
- g. Συνάρτηση **void createBoard()**: Η συνάρτηση αυτή θα δημιουργεί το ταμπλό του παιχνιδιού με ψευδό-τυχαίο τρόπο, χρησιμοποιώντας τις παραπάνω συναρτήσεις.

Προσοχή!!!

- Τα εφόδια τοποθετούνται σε τυχαίες θέσεις πάνω στο ταμπλό με μοναδικό περιορισμό να μην βρίσκονται σε πλακίδιο που υπάρχει παίκτης. Φυσικά, ένα εφόδιο δεν μπορεί να βρεθεί στο ίδιο πλακίδιο με ένα άλλο εφόδιο.
- Η δημιουργία τειχών στα πλακίδια έχει 2 περιορισμούς. Αρχικά, σε ένα πλακίδιο επιτρέπεται να τοποθετηθούν τείχη το πολύ σε δύο πλευρές του. Ακόμη, τα πλακίδια που βρίσκονται στα όρια του λαβυρίνθου θα πρέπει να έχουν τείχος στην εξωτερική πλευρά τους. Παραδείγματος χάρη, εάν δημιουργείτε τη γραμμή με τεταγμένη 0 τότε όλα τα πλακίδια πρέπει να έχουν τείχος στη νότια πλευρά τους, αντίστοιχα εάν δημιουργείτε τη στήλη με τεταγμένη 0 τότε όλα τα πλακίδια πρέπει να έχουν τείχος στη δυτική πλευρά τους.
- Ένα πλακίδιο δε μπορεί να περιέχει παραπάνω από ένα αντικείμενα.
- Εάν δυο πλακίδια είναι γειτονικά και ανήκουν για παράδειγμα στην ίδια γραμμή (όπως φαίνεται στην εικόνα 6) και το πλακίδιο 1 έχει τείχος στα ανατολικά του, τότε συνεπάγεται ότι θα έχει τείχος και το πλακίδιο 2 στα δυτικά του!!!



Εικόνα 6 Τείχη γειτονικών πλακιδίων

h. Συνάρτηση `String[][] getStringRepresentation(int theseusTile, int minotaurTile)`: Η συνάρτηση αυτή θα δημιουργεί και θα επιστρέφει έναν πίνακα διάστασης $(2*N + 1) \times N$. Κάθε πλακίδιο χρειάζεται δύο γραμμές για να αναπαρασταθεί. Στην 1^η γραμμή διατηρούνται τα οριζόντια τείχη (" + --- + "), ενώ στη δεύτερη γραμμή τα πλάγια τείχη και τα περιεχόμενα του πλακιδίου ("| M |" ή " "). Σε περίπτωση που το πλακίδιο δεν περιέχει κάποιο αντικείμενο, τότε θα παραμένει κενό.

Χρησιμοποιήστε την ακόλουθη κωδικοποίηση αντικειμένων:

- **Εφόδια**: S1, S2, ... , Sn
- **Player 1 (Θησέας)**: T
- **Player 2 (Μινώταυρος)**: M

Παράδειγμα λαβύρινθου διάστασης 7x7 (`getStringRepresentation()` πίνακας 15x7)

```

+ --- + --- + --- + --- + --- + --- +
|           |           |           |
+  +  +  +  +  +  +  +
|  |           |           s2      |
+  +  +  +  + ---+ ---+ ---+
|           s1           |           |
+  +  + ---+ ---+ ---+  +  +
|           |           |           |
+  +  +  +  +  +  +  +
| s3      |           M      |           |
+  +  +  +  +  +  +  +
|           |           |           |
+  +  + ---+ ---+ ---+  +  +
|  T           |           |           |
+  + ---+ ---+ ---+ ---+ ---+

```

Κλάση Player

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

- i. **int playerId**: ο κωδικός του παίκτη.
- ii. **String name**: το όνομα του παίκτη.
- iii. **Board board**: το ταμπλό του παιχνιδιού.
- iv. **int score**: το σκορ του παίκτη, το οποίο ανεβαίνει κατά ένα κάθε φορά που βρίσκει ένα εφόδιο.
- v. **int x**: η συντεταγμένη x του πλακιδίου όπου βρίσκεται ο παίκτης
- vi. **int y**: η συντεταγμένη y του πλακιδίου όπου βρίσκεται ο παίκτης

Οι συναρτήσεις που πρέπει να υλοποιήσετε είναι οι εξής:

- a. Οι **constructors** της κλάσης.
- b. Όλες οι συναρτήσεις **get** και **set** για τις μεταβλητές της κλάσης.
- c. Συνάρτηση **int[] move(int id)**: Η συνάρτηση αυτή θα επιλέγει τυχαία μια από τις διαθέσιμες κινήσεις του παίκτη για τον υπολογισμό της επόμενης θέσης του παίκτη, θα μετακινεί τον παίκτη στη νέα θέση και θα επιστρέφει έναν πίνακα ακεραίων που θα περιλαμβάνει το id και τις συντεταγμένες του αντίστοιχου πλακιδίου μετά την κίνηση και το αναγνωριστικό των εφοδίων που μάζεψε κατά τη συγκεκριμένη κίνησή του. Επίσης, θα εκτυπώνει στην κονσόλα αντίστοιχο μήνυμα κάθε φορά που ένας μαζεύει ένα εφόδιο ή δεν μπορεί να κινηθεί.

Προσοχή!!!

- Όταν ένας παίκτης μαζεύει ένα εφόδιο, τότε αυτό εξαφανίζεται από το ταμπλό. Για δική σας ευκολία, θα μπορούσατε απλά να μηδενίζεται τις συντεταγμένες του εφοδίου.

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 15x15.
 - Εισαγωγή 4 εφοδίων στο ταμπλό.
 - Ορισμός 2 παικτών.

- Οι παίκτες παίζουν εναλλάξ (ρίχνουν το ζάρι και κινούνται στο ταμπλό) μέχρις ότου ένας από τους 2 τερματίσει ή να παρέλθουν οι $2n$ (όπου $n = 100$) ζαριές.
- Τα συνολικά τείχη που μπορούν να δημιουργηθούν είναι: $(N * N * 3 + 1) / 2$
- Σε κάθε γύρο θα εκτυπώνονται τα εξής: Το round, το ταμπλό του παιχνιδιού και η κίνηση του κάθε παίκτη. Για την εκτύπωση του ταμπλό θα χρησιμοποιηθεί η συνάρτηση `getStringRepresentation()` της κλάσης `Board`.
- Στο τέλος του παιχνιδιού θα εκτυπώνεται ο νικητής 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 "ΑΕΜ1_ΑΕΜ2_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 **Τρίτη 10 Νοεμβρίου, 23:59** (ηλεκτρονικά)
Δε θα υπάρξει καμία παρέκκλιση από την παραπάνω προθεσμία.