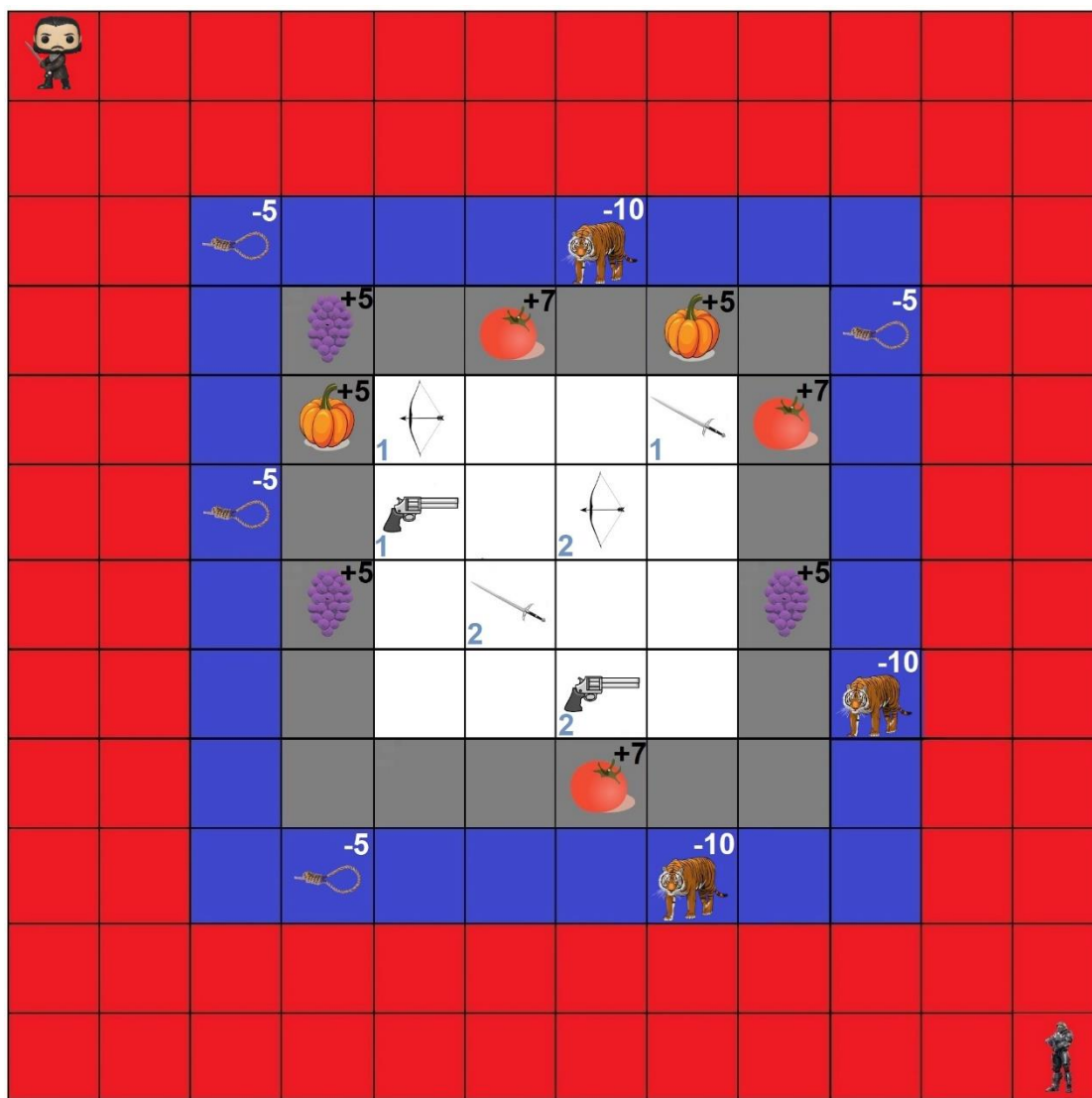


## ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

### Hunger Games

*May the odds be ever in your favor..*

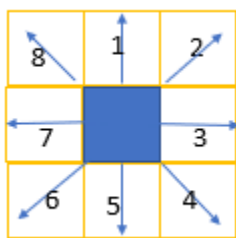
Το Hunger Games είναι ένα τηλεοπτικό ετήσιο γεγονός στο οποίο οι συμμετέχοντες είναι αναγκασμένοι να αγωνιστούν μέχρι θανάτου, έως ότου να υπάρξει κάποιος νικητής. Σε αυτό συμμετέχουν ένα αγόρι και μια κοπέλα από διάφορες περιφέρειες και λαμβάνει χώρα σε ένα απομονωμένο νησί.



Εικόνα 1: Παράδειγμα ταμπλό παιχνιδιού διάστασης 12x12.

Για να γίνεται πιο συναρπαστικό και γρήγορο το παιχνίδι, το νησί έχει την ιδιαιτερότητα ότι η διάμετρος του μικραίνει καθώς περνάει ο χρόνος και οι παίκτες έρχονται όλο και πιο κοντά. Οι παίκτες αρχίζουν το παιχνίδι δίχως να έχουν εφόδια, και συνεπώς για να επιβιώσουν πρέπει να αναζητήσουν τροφή. Όμως, το να επιβιώσουν δεν είναι αρκετό καθώς ορισμένοι αντίπαλοι τους, τους αναζητούν για να τους εξολοθρεύσουν. Για το λόγο αυτό, κάθε παίκτης κρίνεται απαραίτητο να έχει στην κατοχή του ένα όπλο. Αυτά βρίσκονται τοποθετημένα στο κέντρο του νησιού και οι παίκτες θα πρέπει να διασχίσουν μια μεγάλη και επικίνδυνη διαδρομή εάν θέλουν να αποκτήσουν ένα. Αυτό μπορεί να χρησιμοποιηθεί τόσο για να αποφύγει διάφορες παγίδες που είναι διασκορπισμένες στο νησί όσο και για να σκοτώσει κάποιο αντίπαλό του ή κάποιο άγριο ζώο που караδοκεί. Όπως γίνεται αντιληπτό, νικητής είναι αυτός που θα καταφέρει να παραμείνει ζωντανός και να επιβιώσει από όλους τους κινδύνους.

Στην παρούσα εργασία, καλείστε να υλοποιήσετε μια απλουστευμένη παραλλαγή του παιχνιδιού Hunger Games. Ο αριθμός των παικτών που συμμετέχουν στο παιχνίδι είναι μόνο δύο. Στην αρχή του παιχνιδιού, οι δύο παίκτες βρίσκονται αντιδιαμετρικά του ταμπλό, όπως φαίνεται στην Εικόνα 1 και κινούνται προς το κέντρο. Οι παίκτες παίζουν εναλλάξ και μετακινούνται κατά μία θέση πάνω στο ταμπλό κάθε φορά. Έχουν το δικαίωμα να κινηθούν πάνω, κάτω, δεξιά, αριστερά και διαγώνια, όπως φαίνεται στην Εικόνα 2. Οι κωδικοί των κινήσεων είναι οι εξής.



Εικόνα 2 Διαθέσιμες κινήσεις

Το ταμπλό είναι οργανωμένο ως εξής. Στο κέντρο του ταμπλό είναι κατανεμημένα τα όπλα, περιμετρικά από το κέντρο είναι τα εφόδια, ενώ τα τρόφιμα περικλείουν οι παγίδες, όπως φαίνεται στην Εικόνα 1. Τα όπλα είναι απαραίτητα για τους παίκτες για να πολεμήσουν και να αντιμετωπίσουν τις παγίδες, ενώ τα εφόδια είναι απαραίτητα για να ζήσουν. Κάθε φορά που ο παίκτης βρίσκεται σε ένα πλακίδιο που περιλαμβάνει κάποιο εφόδιο, προστίθενται στο σκορ του οι βαθμοί του πλακιδίου. Αντίθετα, όταν ο παίκτης βρεθεί σε πλακίδιο με παγίδα, αφαιρούνται από το σκορ του οι βαθμοί της παγίδας, εκτός και αν έχει εξοπλιστεί με το κατάλληλο όπλο για να την αντιμετωπίσει.

Πρώτος στόχος των παικτών είναι να κατευθυνθούν προς το κέντρο του ταμπλό με σκοπό να πάρουν τα όπλα που θεωρούν χρήσιμα για το παιχνίδι. Υπάρχουν τρία είδη όπλων για κάθε έναν από τους παίκτες: Πιστόλι, Τόξο και Σπαθί. Το **πιστόλι** αποτελεί απαραίτητο όπλο για να πολεμήσει τον αντίπαλο. Δεδομένου ότι το πιστόλι είναι περιορισμένης εμβέλειας, προκειμένου ένας παίκτης να σκοτώσει κάποιον άλλο, θα πρέπει η μεταξύ τους απόσταση να είναι μικρότερη από ένα όριο. Το τόξο και το σπαθί μπορούν να χρησιμοποιηθούν για την αντιμετώπιση παγίδων. Πιο συγκεκριμένα, το **τόξο** μπορεί να χρησιμοποιηθεί για την αντιμετώπιση των παγίδων που περιλαμβάνουν ζώα, ενώ το **σπαθί** για τις παγίδες που περιλαμβάνουν σκοιινιά. Σε περίπτωση που ο παίκτης δεν έχει το κατάλληλο όπλο για να

αντιμετωπίζει τις παγίδες, χάνει τους πόντους που αναγράφονται στο πλακίδιο όπου βρίσκεται η παγίδα.

*Το κέντρο του ταμπλό αποτελεί και το κέντρο των αξόνων – σημείο αναφοράς για τις συντεταγμένες των πλακιδίων.  
Ενδεικτικά, στην*

Εικόνα 3 φαίνονται οι συντεταγμένες των πλακιδίων ενός ταμπλό διάστασης 4x4.

(-2,-2)	(-1,-2)	(1,-2)	(2,-2)
(-2,-1)	(-1,-1)	(1,-1)	(2,-1)
(-2,1)	(-1,1)	(1,1)	(2,1)
(-2,2)	(-1,2)	(1,2)	(2,2)

Εικόνα 3 Συντεταγμένες πλακιδίων ταμπλό διάστασης 4x4

Με το πέρασμα των γύρων του παιχνιδιού το ταμπλό μικραίνει διαμετρικά, διαγράφοντας το εξωτερικό του τετράγωνο. Το παιχνίδι τερματίζει όταν κάποιος από τους δύο παίκτες σκοτώσει τον αντίπαλο ή όταν το ταμπλό έχει πλέον διαστάσεις 4x4. Νικητής του παιχνιδιού είναι ο παίκτης που θα μείνει ζωντανός ή, σε περίπτωση που δε σκοτωθεί κάποιος, ο παίκτης που έχει το μεγαλύτερο δυνατό σκορ στο τέλος του παιχνιδιού.

## Εργασία Α – Random Movement (0,75 βαθμοί)

Στην παρούσα εργασία υπάρχουν 2 βασικά ζητούμενα.

1. Δημιουργία ενός ταμπλό μεγέθους NxM με ορισμένο αριθμό όπλων, εφοδίων και παγίδων, κατανομημένων σε συγκεκριμένες περιοχές στο χώρο.
2. Δημιουργία παίκτη που παίζει 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 [1,8] 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.

**Προσοχή!!!** Όπου αναφέρονται μεταβλητές  $x, y$ , απευθύνονται στο σύστημα συντεταγμένων με κέντρο το (0,0) όπως περιγράφεται στην

Εικόνα 3.

### Κλάση Weapon

Η κλάση **Weapon** θα αντιπροσωπεύει το όπλο και θα έχει τις εξής μεταβλητές:

- i. **int id**: το id του όπλου.
- ii. **int x**: η συντεταγμένη x του πλακιδίου του ταμπλό όπου βρίσκεται το όπλο.
- iii. **int y**: η συντεταγμένη y του πλακιδίου του ταμπλό όπου βρίσκεται το όπλο.
- iv. **int playerId**: το id του παίκτη στον οποίο ανήκει το όπλο.
- v. **String type**: το είδους του όπλου (pistol, bow ή sword).

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 **Weapon**.
- b. Όλες οι συναρτήσεις **get** και **set** για τις μεταβλητές της κλάσης.

### Κλάση Food

Η κλάση **Food** θα αντιπροσωπεύει το εφόδιο και θα έχει τις εξής μεταβλητές:

- i. **int id**: 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 points**: οι πόντοι που κερδίζει ο παίκτης που τρώει 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 **Food**.
- b. Όλες οι συναρτήσεις **get** και **set** για τις μεταβλητές της κλάσης.

### Κλάση Trap

Η κλάση **Trap** θα αντιπροσωπεύει την παγίδα και θα έχει τις εξής μεταβλητές:

- i. **int id**: το id της παγίδας.
- ii. **int x**: η συντεταγμένη x του πλακιδίου του ταμπλό όπου βρίσκεται η παγίδα.
- iii. **int y**: η συντεταγμένη y του πλακιδίου του ταμπλό όπου βρίσκεται η παγίδα.
- iv. **String type**: το είδος της παγίδας (ropes ή animals).
- v. **int points**: οι πόντοι που χάνει ο παίκτης όταν πέσει στην παγίδα και δεν έχει το κατάλληλο όπλο για να την αντιμετωπίσει.

Οι συναρτήσεις που πρέπει να υλοποιήσετε είναι οι εξής:



- vi. **Weapon[] weapons** : ένας πίνακας με αντικείμενα τύπου **Weapon**.
- vii. **Food[] food** : ένας πίνακας με αντικείμενα τύπου **Food**.
- viii. **Trap[] traps** : ένας πίνακας με αντικείμενα τύπου **Trap**.

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 όπλων, τον αριθμό των εφοδίων και τον αριθμό των παγίδων που θα περιέχει το ταμπλό.
    - **Προσοχή!!! Το ταμπλό είναι τετράγωνο, δηλαδή ισχύει πάντα  $N==M$ .**
  - 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 createRandomWeapon()**: Η συνάρτηση αυτή θα αρχικοποιεί τα αντικείμενα του πίνακα **weapons** με τυχαίο τρόπο.
  - f. Συνάρτηση **void createRandomTrap()**: Η συνάρτηση αυτή θα αρχικοποιεί τα αντικείμενα του πίνακα **traps** με τυχαίο τρόπο. Οι πόντοι των παγίδων παίρνουν τυχαίες τιμές στο διάστημα **[-10,-1]**.
  - g. Συνάρτηση **void createRandomFood()**: Η συνάρτηση αυτή θα αρχικοποιεί τα αντικείμενα του πίνακα **food** με τυχαίο τρόπο. Οι πόντοι των εφοδίων παίρνουν τυχαίες τιμές στο διάστημα **[1,10]**.
  - h. Συνάρτηση **void createBoard()**: Η συνάρτηση αυτή θα δημιουργεί το ταμπλό του παιχνιδιού με ψευδό-τυχαίο τρόπο, χρησιμοποιώντας τις τρεις παραπάνω συναρτήσεις.
 

**Προσοχή!!!**

    - Τα όπλα τοποθετούνται στην κεντρική περιοχή στο ταμπλό, εντός των ορίων **weaponAreaLimits**, όπως φαίνεται στην Εικόνα 1.
    - Τα εφόδια τοποθετούνται περιμετρικά των όπλων, εντός των ορίων **foodAreaLimits**, όπως φαίνεται στην Εικόνα 1.
    - Οι παγίδες τοποθετούνται περιμετρικά των εφοδίων, εντός των ορίων **trapAreaLimits**, όπως φαίνεται στην Εικόνα 1.
    - Ένα πλακίδιο δε μπορεί να περιέχει παραπάνω από ένα αντικείμενα.
  - i. Συνάρτηση **void resizeBoard(Player p1, Player p2)**: Η συνάρτηση αυτή μικραίνει το μέγεθος του ταμπλό κατά μια μονάδα περιμετρικά, αφαιρεί δηλαδή την εξωτερικά πλακίδια. Η συρρίκνωση του ταμπλό συμβαίνει μόνο όταν κανένας παίκτης δε βρίσκεται στα εξωτερικά πλακίδια του ταμπλό.
  - j. Συνάρτηση **String[][] getStringRepresentation()**: Η συνάρτηση αυτή θα δημιουργεί και θα επιστρέφει έναν πίνακα διάστασης **NxM**. Κάθε στοιχείο του πίνακα θα περιέχει το αλφαριθμητικό που εκπροσωπεί το αντικείμενο που βρίσκεται πάνω στο αντίστοιχο πλακίδιο του ταμπλό, συνοδευόμενο από το **id** του. 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:*

- Όπλο Player 1: W1
- Όπλο Player 2: W2
- Εφόδιο: F
- Παγίδα: T

Παράδειγμα πίνακα `getStringRepresentation()` διάστασης 8x8:

—	—	—	—	—	—	—	—
—	—	—	T1	—	T2	—	—
—	—	F1	—	F2	—	—	—
—	T6	—	W11	W21	F3	T3	—
—	T5	—	W12	W22	—	—	—
—	—	F5	—	—	F4	—	—
—	—	—	T4	—	—	—	—
—	—	—	—	—	—	—	—

### Κλάση Player

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

- int id**: ο κωδικός του παίκτη.
- String name**: το όνομα του παίκτη.
- Board board**: το ταμπλό του παιχνιδιού.
- int score**: το σκορ του παίκτη, το οποίο θα καθορίζεται από τους πόντους που του δίνουν τα εφόδια και του αφαιρούν οι παγίδες σε περίπτωση που δεν έχει το κατάλληλο όπλο για να τις αντιμετωπίσει.
- int x**: η συντεταγμένη x του πλακιδίου όπου βρίσκεται ο παίκτης
- int y**: η συντεταγμένη y του πλακιδίου όπου βρίσκεται ο παίκτης
- Weapon bow**: το όπλο τύπου bow του παίκτη. Σε περίπτωση που ο παίκτης δεν διαθέτει το συγκεκριμένο όπλο, η τιμή του είναι null.
- Weapon pistol**: το όπλο τύπου pistol του παίκτη. Σε περίπτωση που ο παίκτης δεν διαθέτει το συγκεκριμένο όπλο, η τιμή του είναι null.
- Weapon sword**: το όπλο τύπου sword του παίκτη. Σε περίπτωση που ο παίκτης δεν διαθέτει το συγκεκριμένο όπλο, η τιμή του είναι null.

Οι συναρτήσεις που πρέπει να υλοποιήσετε είναι οι εξής:

- Οι **constructors** της κλάσης.
- Όλες οι συναρτήσεις **get** και **set** για τις μεταβλητές της κλάσης.
- Συνάρτηση **int[2] getRandomMove()**: Η συνάρτηση επιλέγει τυχαία μια από τις διαθέσιμες κινήσεις του παίκτη και επιστρέφει τη νέα του θέση – συντεταγμένες (x,y). **Δώστε προσοχή στις**

**οριακές συνθήκες του προβλήματος!!** Σε περίπτωση που ο παίκτης βρίσκεται στα σύνορα του ταμπλό (εξωτερικά πλακίδια), οι διαθέσιμες κινήσεις του παίκτη είναι λιγότερες από 8.

- d. Συνάρτηση **int[] move()**: Η συνάρτηση αυτή θα καλεί τη συνάρτηση **getRandomMove()** για τον υπολογισμό της επόμενης θέσης του παίκτη, θα μετακινεί τον παίκτη στη νέα θέση και θα επιστρέφει έναν πίνακα ακεραίων που θα περιλαμβάνει τις συντεταγμένες του αντίστοιχου πλακιδίου μετά την κίνηση, τον αριθμό των όπλων που μάζεψε, τον αριθμό των εφοδίων τα οποία μάζεψε και τον αριθμό των παγίδων που του εμφανίστηκαν κατά τη συγκεκριμένη κίνησή του. Επίσης, θα εκτυπώνει στην κονσόλα αντίστοιχο μήνυμα κάθε φορά που ένας μαζεύει ένα όπλο, μαζεύει ένα εφόδιο ή πέφτει πάνω σε μια παγίδα.

**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.
- Ο παίκτης μπορεί να μαζέψει μόνο τα όπλα που του ανήκουν. Υπάρχει ένα είδος όπλου διαθέσιμο για τον κάθε παίκτη. Όταν ο παίκτης μαζέψει ένα όπλο, τότε αυτό εξαφανίζεται από το ταμπλό.

### **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 20x20, με 1 όπλο από το κάθε είδος και για τον κάθε παίκτη (συνολικά 6 όπλα), 10 εφόδια και 8 παγίδες. Τα όπλα βρίσκονται στην περιοχή που περικλείεται από τα πλακίδια με συντεταγμένες (-2,-2), (2,-2), (2,2), (-2,2). Τα εφόδια βρίσκονται περιμετρικά της περιοχής των όπλων στο τετράγωνο που σχηματίζεται από τα πλακίδια με συντεταγμένες (-3,-3), (3,-3), (3,3), (-3,3). Οι παγίδες βρίσκονται περιμετρικά της περιοχής των εφοδίων στο τετράγωνο που σχηματίζεται από τα πλακίδια με συντεταγμένες (-4,-4), (4,-4), (4,4), (-4,4).
  - Ορισμός 2 παικτών.
  - Οι παίκτες παίζουν εναλλάξ (ρίχνουν το ζάρι και κινούνται στο ταμπλό) μέχρις ότου το ταμπλό έχει μέγεθος 4x4 - **Το πιστόλι δε δε θα χρησιμοποιηθεί στη συγκεκριμένη εργασία!**
  - Το ταμπλό του παιχνιδιού μικραίνει κατά μια μονάδα περιμετρικά κάθε 3 γύρους παιχνιδιού.



- 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 του σκορ του κάθε παίκτη.

**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 **Δευτέρα 4 Νοεμβρίου, 13:00** (ηλεκτρονικά)  
**Δε θα υπάρξει καμία παρέκκλιση από την παραπάνω προθεσμία.**