

**Τρίτη Σειρά ασκήσεων**  
**Ημερομηνία Παράδοσης: Τετάρτη 29 Μαΐου Απριλίου 11:59 μ.μ.**

Οι παρακάτω ασκήσεις πρέπει να παραδοθούν μέχρι τις 29 Μαΐου τα μεσάνυχτα. Στην υλοποίηση των κλάσεων σας δεν θα πρέπει να έχετε public μεταβλητές. Βαθμοί θα αφαιρεθούν για προγράμματα που δεν είναι καλά γραμμένα, δηλαδή δεν είναι σωστά στοιχισμένα ώστε να διαβάζονται εύκολα.

### Άσκηση 1

Ο στόχος αυτής της Εργασίας Προγραμματισμού είναι να δημιουργήσετε μια απλή δισδιάστατη προσομοίωση ενός ζεύγους θηρευτή-θηράματος. Σε αυτή την προσομοίωση, τα θηράματα είναι μυρμηγκία (ants) και οι θηρευτές είναι τα μυρμηγκοφάγα πλάσματα που ονομάζονται μυρμηγκοφάγοι (aardvarks). Αυτά τα πλάσματα ζουν σε ένα κόσμο που αποτελείται από ένα πλέγμα κελιών  $20 \times 20$ . Ένα πλάσμα (creature) μπορεί να καταλαμβάνει μόνο ένα κελί κάθε φορά. Το πλέγμα είναι περιφραγμένο, και έτσι δεν επιτρέπεται να απομακρυνθεί ένα πλάσμα πέρα από τις άκρες του πλέγματος. Ο χρόνος προσομοιώνεται με χρονικά βήματα. Κάθε πλάσμα κάνει μια κίνηση σε κάθε χρονικό βήμα.

Τα μυρμηγκία συμπεριφέρονται σύμφωνα με το παρακάτω πρότυπο:

- **Κίνηση.** Σε κάθε χρονικό βήμα, προσπαθούν με τυχαίο τρόπο να μετακινούνται πάνω, κάτω, αριστερά, ή δεξιά. Εάν το κελί στην επιλεγμένη κατεύθυνση είναι κατειλημμένο ή είναι εκτός του πλέγματος, τότε το μυρμηγκί παραμένει στο τρέχον κελί.
- **Αναπαραγωγή.** Εάν ένα μυρμηγκί επιβιώσει για τρία χρονικά βήματα, τότε στο τέλος του τρίτου χρονικού βήματος (δηλαδή, μετά τη μετακίνησή του) το μυρμηγκί θα αναπαραχθεί. Αυτό προσομοιώνεται με τη δημιουργία ενός νέου μυρμηγκιού σε ένα γειτονικό (πάνω, κάτω, αριστερά, ή δεξιά) κελί που είναι άδειο. Εάν δεν υπάρχει κανένα διαθέσιμο άδειο κελί, δε γίνεται αναπαραγωγή. Μόλις δημιουργηθεί ένας απόγονος, το μυρμηγκί δε μπορεί να παράγει νέο απόγονο μέχρι να περάσουν άλλα τρία χρονικά βήματα.

Οι μυρμηγκοφάγοι συμπεριφέρονται σύμφωνα με το παρακάτω πρότυπο:

- **Κίνηση.** Σε κάθε χρονικό βήμα, εάν υπάρχει κάποιο γειτονικό κελί (πάνω, κάτω, αριστερά, ή δεξιά) κατειλημμένο από ένα μυρμηγκί, τότε ο μυρμηγκοφάγος θα μετακινηθεί σε εκείνο το κελί και θα φάει το μυρμηγκί. Διαφορετικά, ο μυρμηγκοφάγος μετακινείται σύμφωνα με τους ίδιους κανόνες που ισχύουν και για τα μυρμηγκία. Πρέπει να σημειωθεί ότι ένας μυρμηγκοφάγος δε μπορεί να τρώει άλλους μυρμηγκοφάγους.
- **Αναπαραγωγή.** Εάν ένας μυρμηγκοφάγος επιβιώσει για οχτώ χρονικά βήματα, τότε θα αναπαραχθεί και θα δημιουργηθεί ένας νέος μυρμηγκοφάγος με τον ίδιο τρόπο που αναπαράγεται και δημιουργείται ένα μυρμηγκί.
- **Θάνατος από πείνα.** Εάν ένας μυρμηγκοφάγος δε φάει ένα μυρμηγκί μέσα στα επόμενα τρία χρονικά βήματα, τότε στο τέλος του τρίτου χρονικού βήματος θα πεινάσει και θα πεθάνει. Τότε ο μυρμηγκοφάγος θα πρέπει να απομακρυνθεί από το πλέγμα των κελιών.

Κατά τη διάρκεια ενός γύρου, όλοι οι μυρμηγκοφάγοι θα πρέπει να μετακινούνται πριν από τα μυρμηγκία.

Να γράψετε ένα πρόγραμμα που να υλοποιεί αυτή την προσομοίωση και να σχεδιάζει τον κόσμο που περιγράφηκε με τη χρήση των χαρακτήρων ASCII. Να συμβολίσετε με το χαρακτήρα "o" ένα μυρμηγκί και με το χαρακτήρα "X" ένα μυρμηγκοφάγο. Να δημιουργήσετε μια κλάση που θα αναπαριστά ένα πλάσμα γενικά, με όνομα **Creature**, η οποία θα ενθυλακώνει βασικά δεδομένα που είναι κοινά για τα μυρμηγκία και για τους μυρμηγκοφάγους. Αυτή η κλάση θα πρέπει να έχει μια υπερβαινόμενη μέθοδο με όνομα **move** που θα ορίζεται στις παραγωγές κλάσεις **Ant**

και **Aardvark**, οι οποίες αναπαριστούν τα μυρμηγκία και τους μυρμηγκοφάγους αντίστοιχα. Μπορεί να χρειαστείτε επιπρόσθετες δομές δεδομένων για να ελέγχετε ποια πλάσματα έχουν μετακινηθεί.

Να αρχικοποιήσετε τον κόσμο με 5 μυρμηγκοφάγους και 100 μυρμηγκία. Μετά από κάθε χρονικό βήμα, να προτρέπετε τους χρήστες να πιάσουν το πλήκτρο Enter για να προχωρούν στο επόμενο χρονικό βήμα. Θα πρέπει να δείτε να σχηματίζεται ένα κυκλικό πρότυπο ανάμεσα στον πληθυσμό των θηρευτών και των θηραμάτων, παρότι οι τυχαίες διαταραχές μπορεί να οδηγήσουν στον αφανισμό του ενός ή και των δυο ειδών.

## Άσκηση 2

Για την άσκηση αυτή θα υλοποιήσετε ένα παιχνίδι μεταξύ του χρήστη και του υπολογιστή. Ο χρήστης ελέγχει ένα παίχτη, και παίζει εναντίον `COMPUTER_PLAYERS = 10` παιχτών που ελέγχει ο υπολογιστής. Ο κάθε παίχτης (υπολογιστή, ή παίχτη) έχει μια δύναμη (`strength`). Για τον παίχτη του χρήστη, η δύναμη είναι μια τυχαία τιμή μεταξύ του 0 και του `MAX_USER_STRENGTH = 100`, για τους παίχτες του υπολογιστή είναι μια τυχαία τιμή μεταξύ 0 και `MAX_COMPUTER_STRENGTH = 20`. Όλοι οι παίχτες κινούνται πάνω σε ένα πλέγμα  $5 \times 5$  από 25 κελιά. Αν ο παίχτης του χρήστη συναντηθεί σε ένα κελί του πλέγματος με ένα παίχτη του υπολογιστή, τότε γίνεται μία «μάχη» μεταξύ των δύο παιχτών, όπου αφαιρούμε την δύναμη του παίχτη του υπολογιστή από αυτή του παίχτη του χρήστη. Αν το αποτέλεσμα είναι αρνητικό τότε ο χρήστης χάνει, και το παιχνίδι τελειώνει. Αλλιώς ο παίχτης του υπολογιστή καταστρέφεται και το παιχνίδι συνεχίζεται. Αν γίνει συνάντηση με παραπάνω από ένα παίχτες του υπολογιστή, τότε ο παίχτης του χρήστη πολεμάει ένα-ένα με όλους τους παίχτες του υπολογιστή.

Υπάρχουν επίσης και τρία όπλα: η χειροβομβίδα (`grenade`), το δακρυγόνο (`teargas`), και ο ιός (`virus`). Αν ο χρήστης έχει στην κατοχή του ένα όπλο, μπορεί να το χρησιμοποιήσει σε ένα κελί. Η χειροβομβίδα καταστρέφει όλους τους παίχτες του υπολογιστή στο κελί χωρίς να χάσει δύναμη ο παίχτης-χρήστης. Το δακρυγόνο έχει ως αποτέλεσμα όλοι οι παίχτες του υπολογιστή να μετακινηθούν εκτός του κελιού (σε κάποιο τυχαίο γειτονικό κελί) χωρίς να γίνει μάχη. Ο ιός έχει αποτέλεσμα η δύναμη του κάθε παίχτη-υπολογιστή στο κελί να μειωθεί κατά 10 μονάδες πριν γίνει η μάχη με τον παίχτη του χρήστη (αρνητική τιμή της δύναμης του παίχτη-υπολογιστή είναι αποδεκτή, θα προκαλέσει αύξηση της δύναμης του παίχτη-χρήστη). Με πιθανότητα 5% ένα όπλο είναι ελαττωματικό, οπότε η χρήση του δεν έχει κανένα αποτέλεσμα και εκτελείται η μάχη κανονικά (αυτή η πληροφορία δεν θα γίνεται γνωστή στο χρήστη παρά μόνο μετά τη χρήση του όπλου). Αν χρησιμοποιηθεί ένα όπλο, μετά καταστρέφεται. Αρχικά ο χρήστης δεν έχει στην κατοχή του όπλο. Τα τρία όπλα είναι τοποθετημένα (τυχαία) σε κελιά του πλέγματος (το κάθε κελί μπορεί να κρατάει το πολύ ένα όπλο). Ο χρήστης μπορεί να μαζέψει (`collect`) ένα όπλο από ένα κελί, να αφήσει (`drop`) ένα όπλο σε ένα κελί, ή να ανταλλάξει (`swap`) το όπλο που έχει με αυτό του κελιού. Ανά πάσα στιγμή δεν μπορεί να έχει στην κατοχή του πάνω από ένα όπλο.

Στην αρχή του παιχνιδιού ο παίχτης του υπολογιστή τοποθετείται στη θέση (0,0), και οι παίχτες του υπολογιστή σε τυχαίες θέσεις. Σε κάθε γύρο του παιχνιδιού γίνονται τα εξής βήματα. Οι παίχτες του υπολογιστή κινούνται τυχαία σε ένα γειτονικό κελί από αυτό που βρίσκονται. Τυπώνεται η νέα κατάσταση του πλέγματος με τις θέσεις των αντιπάλων και των όπλων. Ο χρήστης αποφασίζει αν θα μετακινήσει τον παίχτη του και προς τα πού. Μετά αποφασίζει αν θα μαζέψει, θα αφήσει, ή θα ανταλλάξει όπλο, αν υπάρχει τέτοια δυνατότητα στο νέο κελί. Αν έχει όπλο αποφασίζει αν θα το χρησιμοποιήσει ή όχι, και υλοποιείται η χρήση. Γίνεται η μάχη αν υπάρχουν παίχτες του υπολογιστή. Το παιχνίδι σταματάει όταν ο παίχτης-χρήστης ηττηθεί (νίκη υπολογιστή), ή σκοτωθούν όλοι οι παίχτες του υπολογιστή (νίκη χρήστη), και τυπώνεται το ανάλογο μήνυμα.

Στην υλοποίηση σας δημιουργήστε τις ακόλουθες κλάσεις:

- Μια βασική κλάση **Player** για τον παίχτη και δύο παράγωγες κλάσεις **ComputerPlayer** και **UserPlayer** για τους παίχτες του υπολογιστή και του χρήστη αντίστοιχα.
- Μία κλάση **Cell** που κρατάει πληροφορίες για ένα κελί του πλέγματος (αντίπαλους, όπλο)
- Μία **αφηρημένη** κλάση **Weapon** για το όπλο και παράγωγες κλάσεις **Grenade**, **Teargas**, και **Virus**. Στον constructor της **Weapon** καθορίζεται αν το όπλο είναι ελαττωματικό ή όχι. Η κλάση **Weapon** επίσης θα κρατάει πληροφορία για τον τύπο του όπλου και μια μέθοδο για να την τυπώνει. Επίσης έχει μια **ενυπόστατη** (όχι αφηρημένη) μέθοδο **Use**, που παίρνει σαν όρισμα ένα αντικείμενο τύπου **Cell** η οποία θα ελέγχει αν το όπλο είναι ελαττωματικό, αν είναι θα τυπώνει ένα μήνυμα, αν δεν είναι θα υλοποιεί τη χρήση του όπλου.

(Υπόδειξη: θα χρειαστεί να ορίσετε μια επιπλέον αφηρημένη μέθοδο για να υλοποιήσετε την **Use**).

Κατά την εκτέλεση του παιχνιδιού, τυπώνετε όση πληροφορία χρειάζεται για να μπορεί κάποιος να παρακολουθεί το παιχνίδι. Μπορείτε να δημιουργήσετε πιο εύκολες ή δύσκολες εκδοχές του παιχνιδιού αλλάζοντας τις σταθερές για τον αριθμό των παιχτών, τη μέγιστη δύναμη του παίχτη-χρήστη και παίχτη-υπολογιστή, και τον αριθμό των διαθέσιμων όπλων.

### Άσκηση 3

Στην άσκηση αυτή θα υλοποιήσετε μία πάρα πολύ απλή μηχανή αναζήτησης. Στη σελίδα του μαθήματος υπάρχει ένα αρχείο reviews.txt το οποίο περιέχει 15 (μεγάλες) γραμμές. Η κάθε γραμμή είναι ένα κείμενο, συγκεκριμένα ένα online review για ένα προϊόν. Τα reviews 1-3 είναι για laptops, τα 4-6 για φωτογραφικές μηχανές, τα 7-9 για κινητά τηλέφωνα, τα 10-12 για mp3 players, και τα 13-15 για τηλεοράσεις. Ο στόχος της άσκησης είναι να βρούμε όλες τις διακριτές λέξεις που εμφανίζονται σε αυτά τα κείμενα και για κάθε λέξη να βρούμε σε ποια κείμενα εμφανίζεται. Το κάθε κείμενο έχει ένα id, το νούμερο της γραμμής στην οποία βρίσκεται.

Διαβάσετε ένα-ένα τα κείμενα από το αρχείο, και σπάστε τα σε λέξεις. Θα αφαιρέσετε όλα τα σημεία στίξεως, και θα μετατρέψετε τα κεφαλαία σε μικρά (lowercase). Επίσης θα αφαιρέσετε τις πολύ κοινές λέξεις (stopwords) όπως the, and, a, κλπ. Ένα αρχείο με stopwords στα αγγλικά δίνεται στη σελίδα του μαθήματος. Για κάθε διαφορετική λέξη θα κρατάτε την λίστα με τα ids των κειμένων στα οποία εμφανίζεται. (Στην ορολογία των μηχανών αναζήτησης αυτό ονομάζεται inverted index).

Όταν ολοκληρωθεί η επεξεργασία των κειμένων, ο χρήστης μπορεί να κάνει ερωτήματα (queries). Κάθε ερώτημα είναι μία λέξη. Για κάθε ερώτημα θα επιστρέψετε τη λίστα με τα ids των κειμένων στα οποία εμφανίζεται η λέξη. Εκτυπώστε ένα μήνυμα αν η λέξη δεν εμφανίζεται σε κανένα κείμενο.

Όταν ο χρήστης σταματήσει να δίνει λέξεις, σώστε σε ένα αρχείο το inverted index που δημιουργήσατε. Το αρχείο σε κάθε γραμμή θα έχει μία λέξη, tab, και μια λίστα με τα ids των κειμένων στα οποία εμφανίζεται η λέξη χωρισμένα με κόμμα.

**Bonus:** Στο inverted index για κάθε κείμενο στο οποίο εμφανίζεται μια λέξη κρατήστε και τον αριθμό των εμφανίσεων της λέξης (συχνότητα) μέσα σε αυτό το κείμενο. Η συχνότητα θα πρέπει να εμφανίζεται στην απάντηση στο ερώτημα.

**Υπόδειξη:** Για την άσκηση αυτή θα πρέπει να κάνετε εκτεταμένη χρήση των δόμων της Java: ArrayList, HashSet, και HashMap.

### ΓΕΝΙΚΕΣ ΟΔΗΓΙΕΣ

Κάντε turnin τα προγράμματα σας στο [assignment3@ply212](mailto:assignment3@ply212).

π.χ. turnin assignment3@ply212 <τα αρχεία σας> .

Στον κώδικα να αναγράφονται σε σχόλια το όνομα το login και ο AM σας. Επίσης σε σχόλια γράψτε πριν από κάθε μέθοδο εν συντομία τι κάνει.