

ΕΡΓΑΣΙΕΣ ΟΝΤΟΚΕΝΤΡΙΚΟΥ 2017-2018 (Παράδοση 27/5/2018)

ΕΡΓΑΣΙΑ 1 (JAVA)

(70%)

Στα πλαίσια της εργασίας θα υλοποιηθεί το κλασσικό παιχνίδι «Ναυμαχία», όπου ο χρήστης θα παίζει με αντίπαλο τον υπολογιστή. Το παιχνίδι τελειώνει όταν βυθιστούν όλα τα πλοία ενός από τους παίκτες ή εναλλακτικά μετά από συγκεκριμένο αριθμό βολών. Το πρόγραμμα μπορεί να περιλαμβάνει τα ακόλουθα:

Κλάση Tile: Αναπαριστά ένα κελί από τα οποία απαρτίζεται ο πίνακας του κάθε παίκτη και διαθέτει: Συντεταγμένες (X, Y) και έναν Τύπο που μπορεί να έχει μία από τις παρακάτω τέσσερις τιμές: Sea, Ship, Hit, Miss. Ανάλογα με τον τύπο του κάθε κελιού εμφανίζεται αντίστοιχο σύμβολο στο κελί, π.χ. ~ (sea), s (ship), X (hit), o (miss). Για να προσδιορίσετε τις τιμές που παίρνει ο Τύπος, μπορείτε να χρησιμοποιήσετε δομή enum.

Μπορείτε να ορίσετε **getters** και **setters** για τα παραπάνω μέλη και επίσης μια **μέθοδο draw()** που τυπώνει το σύμβολο του κελιού. Μπορείτε να υπερφορτώσετε ή να συμπεριλάβετε μια Boolean παράμετρο *hidden* στη μέθοδο draw(), ώστε να μην αποκαλύπτει τα κελιά που περιέχουν πλοίο (κατά την εκτύπωση του πίνακα του αντιπάλου).

Κλάση Ship: Είναι μια αφηρημένη κλάση που αναπαριστά την έννοια των πλοίων. Κάθε πλοίο μπορεί να τοποθετηθεί ξεκινώντας από ένα συγκεκριμένο κελί είτε κάθετα προς τα δεξιά είτε οριζόντια προς τα κάτω. Επομένως περιλαμβάνει ως μέλη το Κελί Έναρξης, τον Προσανατολισμό (μία από ακριβώς δύο τιμές) και το Μέγεθος του πλοίου. Από την κλάση Ship κληρονομούν οι ακόλουθες κλάσεις: Carrier, 5 θέσεων, Battleship, 4 θέσεων, Cruiser, 3 θέσεων, Submarine, 3 θέσεων και Destroyer 2 θέσεων.

Μέθοδος placeShip(): Μια βασική μέθοδος που πρέπει να περιλαμβάνει η κλάση είναι η placeShip για την τοποθέτηση του πλοίου πάνω στον πίνακα. Η μέθοδος τροποποιεί τον τύπο των κελιών του πίνακα, ώστε να είναι ship, με βάση το Κελί Έναρξης και τον Προσανατολισμό του πλοίου, που δέχεται ως ορίσματα. Επίσης όρισμα είναι και ο πίνακας πάνω στον οποίο θα τοποθετηθεί το πλοίο. Μπορεί να δέχεται και μια παράμετρο Boolean verbose που να καθορίζει αν θα τυπωθούν διαγνωστικά μηνύματα ή όχι.

Θα πρέπει επίσης να κατασκευαστούν μέθοδοι που ελέγχουν αν η τοποθέτηση είναι 'νόμιμη', αν δηλαδή τηρεί τους παρακάτω περιορισμούς:

- Ένα πλοίο δεν μπορεί να βγαίνει εκτός των ορίων του πίνακα
- Ένα πλοίο δεν μπορεί να τοποθετηθεί σε κελί που ήδη έχει άλλο πλοίο
- Ένα πλοίο δεν μπορεί να εφάπτεται κάθετα ή οριζόντια με κανένα άλλο πλοίο, έστω και για ένα κελί. Αυτό σημαίνει ότι τα πλοία έχουν μεταξύ τους απόσταση τουλάχιστον ένα κελί sea.

Για κάθε μία από τις παραπάνω περιπτώσεις μπορείτε να ορίσετε τον **αντίστοιχο τύπο εξαίρεσης**: **OversizeException**, **OverlapTilesException**, **AdjacentTilesException**. Ο χειρισμός τους θα πρέπει να είναι τέτοιος, ώστε το πλοίο να μην τοποθετείται όταν εγερθεί τέτοια εξαίρεση και να τυπώνεται αντίστοιχο μήνυμα. Για την αρχικοποίηση του Κελιού Έναρξης μπορεί να γίνεται απλώς έλεγχος αν αυτό βρίσκεται εντός των ορίων του πίνακα. Η placeShip μπορεί να επιστρέφει true αν η τοποθέτηση ήταν επιτυχής και να τυπώνει στην έξοδο τον τύπο του πλοίου που τοποθετήθηκε.

Κλάση Board: Αντιστοιχεί στον πίνακα του κάθε παίκτη. Περιλαμβάνει έναν διδιάστατο Πίνακα από κελιά (Tiles) σταθερής διάστασης π.χ. 7X7. Μπορείτε να συμπεριλάβετε και μια Boolean μεταβλητή hidden που θα λαμβάνεται υπόψη όταν εκτυπώνεται ο πίνακας.

Κατασκευαστής: Αρχικοποιεί κατάλληλα τον πίνακα, ώστε όλα τα κελιά αρχικά να είναι τύπου sea.

Μέθοδος drawboards(): Εκτυπώνει στην έξοδο και τους δύο πίνακες. Μπορεί να είναι static. Η drawboards() καλείται κάθε φορά που τοποθετείται πλοίο του χρήστη και κάθε φορά που γίνεται μια βολή. Τα πλοία του αντιπάλου δεν θα πρέπει να αποκαλύπτονται (βλ. κ. μέθοδο draw() στην Tile).

	-	-	Y	O	U	-	-		-	O	P	P	O	N	E	N	T	-
	0	1	2	3	4	5	6		0	1	2	3	4	5	6			
0	~	~	~	~	~	~	s		0	~	~	~	~	~	~	~		
1	~	s	~	s	~	~	s		1	~	~	~	~	~	~	~		
2	~	s	~	s	~	~	s		2	~	~	~	~	X	~	~		
3	~	s	~	s	~	~	s		3	~	~	~	~	~	~	~		
4	~	~	~	~	~	~	~		4	~	~	~	~	~	~	~		
5	s	~	s	s	s	s	s		5	~	~	~	~	~	~	~		
6	s	~	~	O	~	~	~		6	~	~	~	~	~	~	~		

Εικόνα 1. Ενδεικτική έξοδος της drawboards() μετά από επιτυχή βολή και αστοχία του HY.

Μέθοδος getAdjacentTiles(): βοηθητική μέθοδος που δέχεται ένα κελί και επιστρέφει τα γειτονικά του. Χρήσιμη για τους ελέγχους τοποθέτησης ενός πλοίου.

Μέθοδος placeAllShips(): Δημιουργεί όλα τα πλοία και τα τοποθετεί πάνω στον πίνακα με τυχαίο τρόπο (πρέπει να τηρούνται οι περιορισμοί). Καλεί την placeShip(). Εκτός από τον υπολογιστή, μπορεί να τη χρησιμοποιήσει και ο χρήστης, όταν δεν θέλει να τοποθετήσει μόνος του τα πλοία του. Στην περίπτωση που χρησιμοποιείται η placeAllShips(), δεν θα πρέπει να τυπώνονται τα μηνύματα των εξαιρέσεων της placeShip().

Μέθοδος allShipsSunk(): Ελέγχει αν όλα τα πλοία του πίνακα έχουν βυθιστεί, δηλαδή αν δεν υπάρχει κελί τύπου ship στον πίνακα, οπότε και επιστρέφει true, διαφορετικά false. Σε περίπτωση που επιστρέψει true, για οποιονδήποτε πίνακα, το παιχνίδι τερματίζεται.

Κλάση Player: Συμβολίζει τους παίκτες. Θα δημιουργηθούν δύο στιγμιότυπα: χρήστη και υπολογιστής. Διαθέτει το Όνομα του κάθε παίκτη, καθώς και μετρητές στατιστικών για τον καθένα: Σύνολο Βολών, Αστοχών, Επιτυχημένων Βολών και Επαναλήψεων (βολή σε κελί που έχει ήδη γίνει βολή).

Μέθοδος placeAllShips(): Καλεί την placeAllShips() του board του παίκτη, η οποία τοποθετεί όλα τα πλοία τυχαία.

Μέθοδος placeShip(): Δέχεται σαν παράμετρο ένα πλοίο και συντεταγμένες και καλεί την placeShip() για το πλοίο αυτό. Επιστρέφει το ίδιο αποτέλεσμα.

Μέθοδος fire(): Πραγματοποιεί μια βολή στον πίνακα του αντιπάλου. Δέχεται ως είσοδο τον πίνακα που θα γίνει η βολή και έναν πίνακα δύο συντεταγμένων. Πραγματοποιεί τη βολή μετατρέποντας τον τύπο του κελιού σε hit ή miss. Με βάση το αποτέλεσμα της βολής τυπώνει μήνυμα (π.χ. hit, miss, already hit, already miss) μαζί με το όνομα του παίκτη που την πραγματοποίησε. Παράλληλα, ενημερώνει τους μετρητές στατιστικών.

Μέθοδος getStats(): Επιστρέφει τα στατιστικά του παίκτη.

Κλάση Game: Αποτελεί τη κεντρική κλάση του παιχνιδιού, που περιλαμβάνει τη main και άλλες βοηθητικές static μεθόδους.

Main(): Δημιουργεί δύο παίκτες (χρήστη και υπολογιστής). Ρωτά το χρήστη αν θέλει να τοποθετήσει τα πλοία του τυχαία ή δίνοντας τις συντεταγμένες και τον προσανατολισμό για το καθένα από το πληκτρολόγιο (Y/N και V/H). Δημιουργεί και τοποθετεί τα πλοία του χρήστη. Τοποθετεί τυχαία τα πλοία του υπολογιστή. Καλεί την drawboards(). Στη συνέχεια, επαναλαμβάνει τα παρακάτω ώσπου να γίνει true η AllShipsSunk() ή μετά από συγκεκριμένο αριθμό βολών: 1. Διαβάζει συντεταγμένες (getInput) και πραγματοποιεί μια βολή (fire) για τον χρήστη. 2. Κάνει το ίδιο για τον υπολογιστή με τυχαίες συντεταγμένες. 3. Καλεί την drawboards(). Όταν τερματίσει ο βρόχος, τυπώνει τον νικητή, καθώς και τα στατιστικά κάθε παίκτη.

Μέθοδος getInput(): Διαβάζει δύο ακεραίους αριθμούς από το πληκτρολόγιο, χωρισμένους με κενό. Επιστρέφει τις συντεταγμένες (πίνακας δύο ακεραίων). Αγνοεί οποιαδήποτε είσοδο δεν αντιστοιχεί σε ακεραίους. Μπορείτε να χρησιμοποιήσετε την κλάση java.util.Scanner και τη μέθοδό της, nextInt(). Ελέγχει αν οι συντεταγμένες είναι εντός ορίων. Αν όχι, ζητά εκ νέου είσοδο.

Μέθοδος getRandInput(): Δημιουργεί και επιστρέφει δύο τυχαίες συντεταγμένες εντός ορίων.

Μέθοδος getOrientation(): Διαβάζει ένα γράμμα από το πληκτρολόγιο και επιστρέφει Orientation.horizontal αν "H" και Orientation.vertical αν "V".

Μέθοδος randomPlace(): Διαβάζει ένα γράμμα από το πληκτρολόγιο και επιστρέφει true αν "Y" και false αν "N".

- ❖ 1ο Bonus Ερώτημα για την JAVA μόνο (+10% στη βαθμολογία της εργασίας Java)
Αντί για απλή εκτύπωση της εξέδου με χαρακτήρες, χρησιμοποιήστε την βιβλιοθήκη Swing για μια απλή παραθυρική εφαρμογή.
- ❖ 2ο Bonus Ερώτημα για την JAVA μόνο (+10% στη βαθμολογία της εργασίας Java)
Ο υπολογιστής δεν πραγματοποιεί τυχαία βολή, αλλά όταν έχει hit, προσπαθεί να κάνει βολές σε γειτονικά κελιά και να «μαντέψει» τον προσανατολισμό του πλοίου του αντιπάλου, όπως θα έκανε ο χρήστης. Επίσης, λαμβάνει υπόψη τους περιορισμούς τοποθέτησης των πλοίων.

ΕΡΓΑΣΙΑ 2 (C++)

(30%)

Μετατρέψτε το πρόγραμμα σε C++, με τις ακόλουθες απλοποιήσεις:

- Υπάρχει μόνο ένας πίνακας (Board) 5X5, αυτός του υπολογιστή.
 - Στον πίνακα τοποθετούνται μόνο ένα BattleShip, ένα Cruiser και ένα Destroyer.
 - Τα πλοία τοποθετούνται τυχαία (ισχύουν οι περιορισμοί).
 - Ο χρήστης πραγματοποιεί 10 βολές (εκτός αν τα πλοία έχουν βυθιστεί νωρίτερα) και τυπώνονται τα στατιστικά του.
- ❖ Bonus Ερώτημα για την C++ μόνο (+10% στη βαθμολογία της εργασίας C++)
Κάντε υπερφόρτωση 3 τελεστών της επιλογής σας για απλοποίηση του κώδικα.
 - ❖ ΔΙΕΥΚΡΙΝΗΣΗ: Όσοι χρωστούν μόνο το project C++, είτε διότι χρωστούν το παλιό μάθημα Οντοκεντρικός Προγραμματισμός II είτε διότι έδωσαν το project Java άλλη χρονιά, θα πρέπει να κάνουν την πλήρη έκδοση της εφαρμογής σε C++, δηλαδή χωρίς τις απλοποιήσεις.

ΑΞΙΟΛΟΓΗΣΗ

Η βαθμολόγηση πέρα από την σωστή εκτέλεση του προγράμματος όπως περιγράφεται παραπάνω θα βασιστεί στην σωστή εφαρμογή των εννοιών οντοκεντρικού προγραμματισμού που διδάσκονται στο μάθημα:

1. Σωστή δήλωση των κλάσεων και ιεράρχησή τους όπως ζητείται.
2. Αποφυγή χρήσεως public όπου δεν είναι απαραίτητο
3. Υπερκάλυψη μεθόδων και πολυμορφική χρήση τους όπου χρειάζεται.
4. Χρήση Static μεταβλητών/μεθόδων για γενική πληροφορία που αφορά κάθε κλάση.
5. Έμφαση στην αντικειμενοστρέφεια και την δυνατότητα επαναχρησιμοποίησης κώδικα.
6. Λίστες / Πίνακες
7. Ποιότητα κώδικα, αναγνωσιμότητα και επαρκής σχολιασμός

8. Η έξοδος του προγράμματος να είναι καλά μορφοποιημένη και εύκολα ερμηνεύσιμη από τον χρήστη
9. Παραμετροποίηση του κώδικα. Κρατήστε σε ένα σημείο συγκεντρωμένες όλες τις παραμέτρους εκτέλεσης του προγράμματος (πχ διαστάσεις πινάκων, αριθμός βολών) ώστε να μπορεί κάποιος από ένα σημείο να τροποποιεί εύκολα τις παραμέτρους του παιχνιδιού.

Βαθμολογικά, η Εργασία 1 (Java) συνεισφέρει 70% στην τελική βαθμολογία των εργασιών του μαθήματος και η Εργασία 2 (C++) 30%.

ΠΑΡΑΔΟΣΗ

- Οι εργασίες θα πρέπει να παραδοθούν στο e-Class μέχρι και την Κυριακή 27/5/2018 (23:55).
- Οι εργασίες είναι ομαδικές (μέχρι 3 άτομα).
- Παραδίδετε ξεχωριστά αρχεία (ένα για τη JAVA και ένα για την C++). Τα αρχεία (zip ή rar) παραδίδει ένα μόνο άτομο της ομάδας. Στο όνομα του αρχείου να φαίνονται όλα τα ΑΜ και ποια εργασία από τις δύο είναι:

2859_2973_2924_PROJECT_1.rar // για την JAVA

2859_2973_2924_PROJECT_2.rar // για την C++

Σε κάθε εργασία πρέπει να συμπεριλάβετε σύντομη αναφορά όπου να αναφέρετε:

- Τα στοιχεία (ΟΝΟΜΑΤΑ - ΑΜ - email) των μελών της ομάδας
- Διάγραμμα κλάσεων (UML).
- Σύντομη περιγραφή της υλοποίησης σας

ΠΡΟΦΟΡΙΚΗ ΕΞΕΤΑΣΗ

Η εξέταση θα γίνει την εβδομάδα που αναφέρεται στο πρόγραμμα που υπάρχει στη σελίδα Πληροφορίες του e-Class.