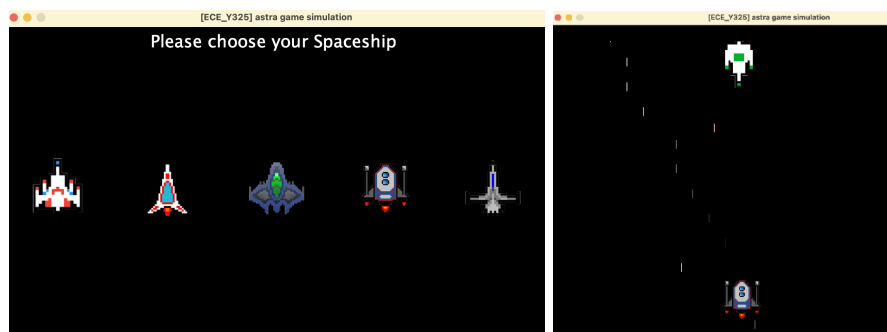










Άσκηση 6η

Ο σκοπός της 6^η άσκησης είναι η απόκτηση εμπειρικής γνώσης σχετικά με την αποδόμηση ενός πολύπλοκου προβλήματος σε επιμέρους κλάσεις, αφηρημένες κλάσεις και διεπαφές (interfaces). Η 6^η άσκηση είναι η πρώτη από μία σύνθετη εργασία τριών ασκήσεων (6^η, 7^η και 8^η) που έχουν ως στόχο το σχεδιασμό και την υλοποίηση ενός παιχνιδιού τύπου *arcade spaceship shooting game*. Τα παιχνίδια τύπου *arcade games* είναι δημοφιλή σε κινητές διαδραστικές συσκευές επιφέροντας ετήσιο τζίρο στις εταιρείες παραγωγής παιχνιδιών πολλών εκατοντάδων εκατομμυρίων ευρώ.

Το παιχνίδι μας στο τέλος της 8^{ης} άσκησης θα έχει δύο *panels*, το αριστερό σχήμα δείχνει το *panel* επιλογής του διαστημοπλοίου, ενώ το δεξί σχήμα δείχνει το *panel* του παιχνιδιού στο οποίο ο εχθρός (υπολογιστής) βρίσκεται στο πάνω μέρος, ενώ ο χρήστης χειρίζεται το επιλεγμένο διαστημόπλοιο στο κάτω μέρος του *panel*.



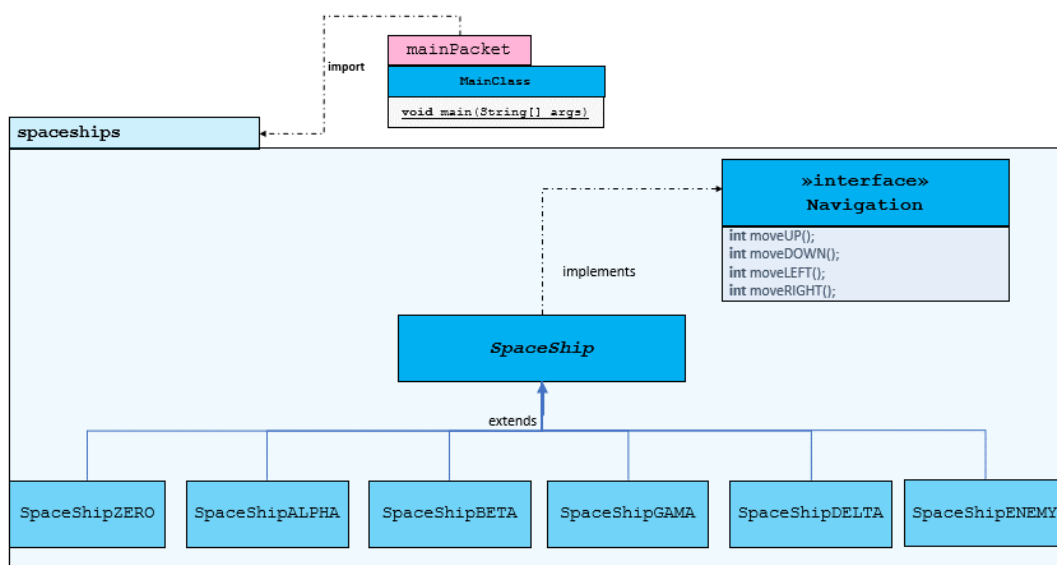
Περιγραφή του προβλήματος: Τα διαστημόπλοια κινούνται σε ένα καρτεσιανό χώρο δύο διαστάσεων (μπορείτε να ορίσετε τις x , y διαστάσεις σε pixels όπως επιθυμείτε). Υπάρχουν πέντε διαστημόπλοια που υλοποιούν τις συμπεριφορές κίνησης στον καρτεσιανό χώρο, ήτοι *moveUp*, *moveDown*, *moveLeft*, *moveRight*. Η διαφορά όμως που έχουν μεταξύ τους είναι η ταχύτητα μετακίνησης όπως φαίνεται στον παρακάτω πίνακα με ενδεικτικές τιμές (pixels) μετακίνησης της εικόνας.

Διαστημόπλοιο		MoveUp/MoveDown	MoveLeft/MoveRight
SpaceShipZERO		Μετακίνηση της εικόνας κατά 5 pixels πάνω/κάτω	Μετακίνηση της εικόνας κατά 5 pixels αριστερά/δεξιά
SpaceShipALPHA		Μετακίνηση της εικόνας κατά 10 pixels πάνω/κάτω	Μετακίνηση της εικόνας κατά 10 pixels αριστερά/δεξιά
SpaceShipBETA		Μετακίνηση της εικόνας κατά 20 pixels πάνω/κάτω	Μετακίνηση της εικόνας κατά 20 pixels αριστερά/δεξιά
SpaceShipGAMA		Μετακίνηση της εικόνας κατά 30 pixels πάνω/κάτω	Μετακίνηση της εικόνας κατά 30 pixels αριστερά/δεξιά
SpaceShipDELTA		Μετακίνηση της εικόνας κατά 40 pixels πάνω/κάτω	Μετακίνηση της εικόνας κατά 40 pixels αριστερά/δεξιά
SpaceShipENEMY		Μετακίνηση της εικόνας κατά 15 pixels πάνω/κάτω	Μετακίνηση της εικόνας κατά 15 pixels αριστερά/δεξιά



Ο χρήστης θα χειρίζεται το διαστημόπλοιο του μέσω του πληκτρολογίου (up/down/left/right arrows) και αναλόγως θα καλούνται οι αντίστοιχες μέθοδοι του κάθε διαστημοπλοίου για μετακίνηση. Θα πρέπει να υλοποιήσετε τις μεθόδους του κάθε διαστημοπλοίου έτσι ώστε το διαστημόπλοιο πάντα να βρίσκεται μέσα στον κόσμο των καρτεσιανών διαστάσεων x, y που έχετε ορίσει.

Ζητούμενα της 6^{ης} Άσκησης: Καλείστε να υλοποιήσετε μία εφαρμογή, η οποία χρησιμοποιώντας τις αρχές καλής πρακτικής του αντικειμενοστρεφούς σχεδιασμού, θα αποτελέσει την βάση για την περαιτέρω ανάπτυξη του παιχνιδιού. Μια ιδέα θα ήταν να χρησιμοποιήσετε ιεραρχία κλάσεων που υλοποιούν αφηρημένες κλάσεις και διεπαφές όπως φαίνεται στο παρακάτω ενδεικτικό διάγραμμα.



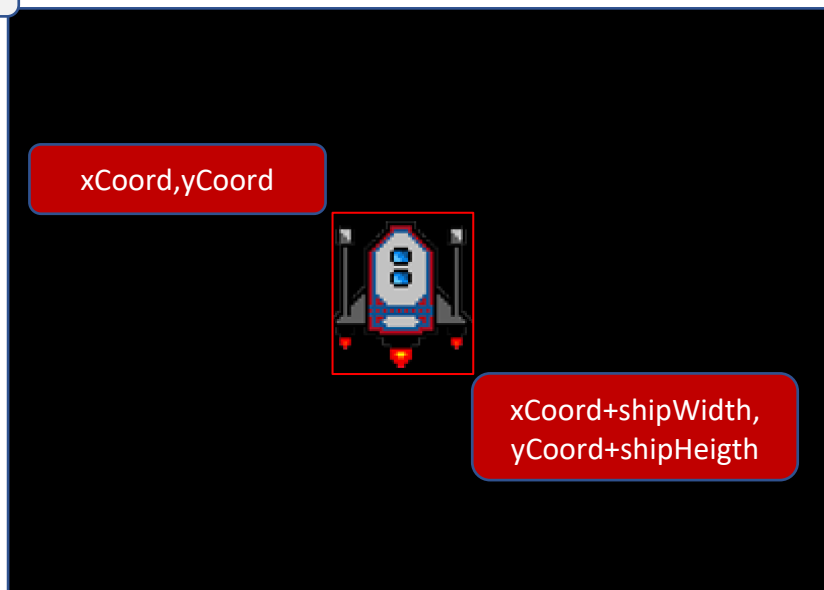
Στο παραδοτέο της 6^{ης} άσκησης η εφαρμογή θα δημιουργεί 6 αντικείμενα διαστημοπλοίων και θα τα βάζει σε μια λίστα. Στη συνέχεια θα ζητάει από το πληκτρολόγιο εντολή μετακίνησης των διαστημοπλοίων ως εξής: u για up , d για $down$, l για $left$, r για $right$. Ανάλογα με την εντολή, θα μετακινεί όλα τα διαστημόπλοια αναλόγως με τις προδιαγραφές που σας δόθηκαν παραπάνω. Ένα παράδειγμα εκτέλεσης για την εντολή up φαίνεται στο παρακάτω σχήμα.

```
Enter command to test spaceships movements:
u
uSpaceshipZERO Xcoords:0 Ycoords:475
SpaceshipALPHA Xcoords:0 Ycoords:470
SpaceshipBETA Xcoords:0 Ycoords:460
SpaceshipGAMA Xcoords:0 Ycoords:450
SpaceshipDELTA Xcoords:0 Ycoords:440
SpaceshipENEMY Xcoords:700 Ycoords:30
```

Αν η εντολή μετακίνησης ενός διαστημοπλοίου το μετακινεί έξω από τον προσδιορισμένο καρτεσιανό χώρο τότε η μετακίνηση αυτή για το συγκεκριμένο διαστημόπλοιο δεν θα επιτρέπεται. Θεωρούμε ότι το σημείο με συντεταγμένες $0, 0$ είναι το πάνω αριστερό άκρο του πλαισίου (Frame) στο οποίο θα εξελιχθεί το παιχνίδι. Θεωρούμε ότι η αρχική θέση του επιλεγμένου διαστημοπλοίου του χρήστη είναι η κάτω αριστερή και ότι με τις εντολές μετακινείται στον καρτεσιανό χώρο όπως φαίνεται παρακάτω.



0,0



CosmosWidth,
CosmosHeight

Ενδεικτική δομή των πακέτων της εργασίας:

```
ProjectWeek6v1
├── JRE System Library [jdk-11.0.8.jdk]
├── src
│   ├── main
│   │   └── MainClass.java
│   └── spaceships
│       ├── Navigation.java
│       ├── SpaceShip.java
│       ├── SpaceShipALPHA.java
│       ├── SpaceShipBETA.java
│       ├── SpaceShipDELTA.java
│       ├── SpaceShipENEMY.java
│       ├── SpaceShipGAMA.java
│       └── SpaceShipZERO.java
```

Ενδεικτικός κώδικας:

```
public class MainClass {
    public static int cosmosWidth=800;
    public static int cosmosHeight=600;
    public static int spaceShipWidth=100;
    public static int spaceShipHeight=100;
    public static void main(String[] args) {
        LinkedList<SpaceShip> spaceshiplist=new LinkedList<SpaceShip>();
        SpaceShip z= new SpaceShipZERO();
        SpaceShip a= new SpaceShipALPHA();
        SpaceShip b= new SpaceShipBETA();
        SpaceShip g= new SpaceShipGAMA();
        SpaceShip d= new SpaceShipDELTA();
        SpaceShip e= new SpaceShipENEMY();

        spaceshiplist.add(z);
        spaceshiplist.add(a);
        spaceshiplist.add(b);
        spaceshiplist.add(g);
        spaceshiplist.add(d);
        spaceshiplist.add(e);
        System.out.println("Enter command to test spaceships movements:");
        Scanner sc = new Scanner (System.in);
        while(sc.hasNext()) {
            String s1 = sc.next();
            System.out.print(s1);
            int i=0;
            while (i < spaceshiplist.size()) {
                if(s1.equals("u"))spaceshiplist.get(i).moveUP();
                if(s1.equals("d"))spaceshiplist.get(i).moveDOWN();
                if(s1.equals("l"))spaceshiplist.get(i).moveLEFT();
                if(s1.equals("r"))spaceshiplist.get(i).moveRIGHT();
                spaceshiplist.get(i).printCoords();
                i++;
            }
            sc.close();
        }
    }
}
```

```
package spaceships;

public interface Navigation {
    int moveUP();//return ycoord
    int moveDOWN();//return ycoord
    int moveLEFT();//return xcoord
    int moveRIGHT();//return xcoord
}
```