



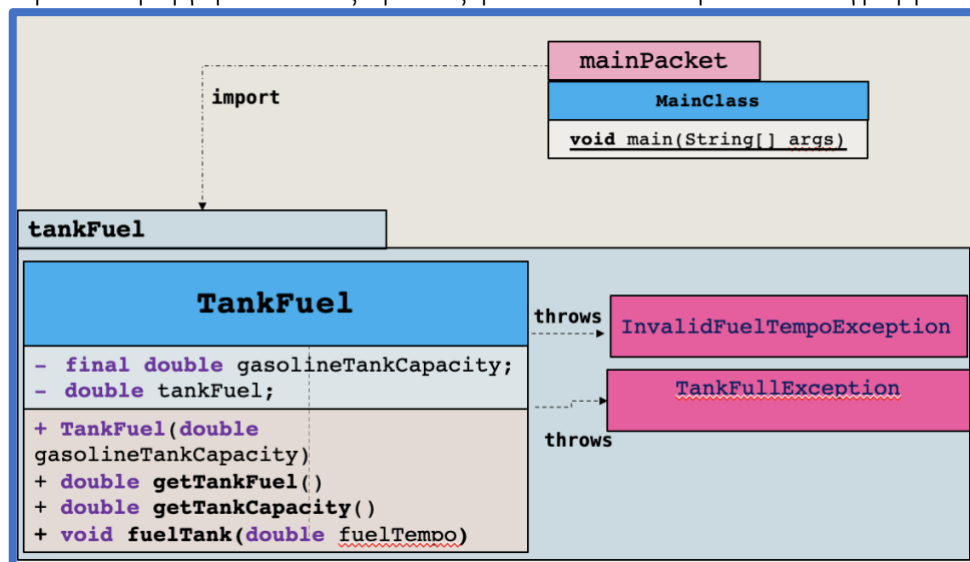
Άσκηση 3η

Περιγραφή του προβλήματος

Ο σκοπός της 3^{ης} άσκησης είναι η απόκτηση εμπειρικής γνώσης σχετικά με τη χρήση μεθόδων μεταλλαγής (*mutator*) και πρόσβασης (*accessor*) στα πλαίσια εφαρμογών αντικειμενοστρεφούς προγραμματισμού. Επίσης, μέσω της εργασίας θα εμπλακείτε και στη διαχείριση εξαιρέσεων (*exception handling*).

Ζητούμενα: Καλείστε να δημιουργήσετε μια κλάση **T a n k F u e l** η οποία αναπαριστά και υλοποιεί τη διαδικασία γεμίσματος του ρεζερβουάρ ενός μεταφορικού μέσου. Η κλάση έχει τις παρακάτω ιδιότητες: α) μια *final ιδιωτική* μεταβλητή τύπου *double* με όνομα **gasolineTankCapacity**, η οποία αρχικοποιείται μέσω του κατασκευαστή (constructor) της κλάσης, και αντιστοιχεί στην ονομαστική χωρητικότητα του ρεζερβουάρ του οχήματος και β) μια μεταβλητή **tankFuel**, η οποία αντιστοιχεί στην τρέχουσα τιμή του καυσίμου που υπάρχει στο ρεζερβουάρ (θεωρούμε ότι η αρχική τιμή της *tankFuel* είναι μηδέν).

Η κλάση υλοποιεί τις λειτουργίες πρόσβασης **getTankFuel** και **getTankCapacity** οι οποίες επιστρέφουν αντίστοιχα τις τιμές των προαναφερθέντων ιδιοτήτων. Επίσης, η κλάση υλοποιεί την μέθοδο μεταλλαγής **fuelTank** η οποία **κάθε φορά που καλείται** σηματοδοτεί το γέμισμα του ρεζερβουάρ του οχήματος με καύσιμο είτε κατά 0.1 είτε κατά 0.2 λίτρα. Τα όρια αυτά (0.1 ή 0.2) έχουν δοθεί ώστε να μη δημιουργείται μεγάλη πίεση στα ηλεκτρονικά του ρεζερβουάρ από το γέμισμά του. Για την ομαλή λειτουργία της διαδικασίας του γεμίσματος του ρεζερβουάρ του οχήματος, η μέθοδος **fuelTank** δημιουργεί δυο εξαιρέσεις: α) όταν ο απαιτούμενος ρυθμός γεμίσματος του ρεζερβουάρ δεν υποστηρίζεται και β) όταν το ρεζερβουάρ είναι ήδη γεμάτο. Οι εξαιρέσεις φαίνονται στο παρακάτω διάγραμμα:



Δημιουργείτε μια κλάση **M a i n C l a s s** η οποία θα δημιουργεί ένα αντικείμενο τύπου **TankFuel** και στη συνέχεια θα καλεί την μέθοδο **fuelTank** έως ότου γεμίσει το ρεζερβουάρ.



Ενδεικτικός Κώδικας της *main*:

```
import tankFuel.*;
public class MainClass {

    public static void main(String[] args) {
        TankFuel myTank=new TankFuel(50);
        boolean fillTank=true;
        double fuelTempo=0.1;
        while(fillTank){

            try {
                myTank.fuelTank(fuelTempo);
            }
            catch (InvalidFuelTempoException ex){
                fillTank=false;
                System.out.println(ex.getMessage());
            }
            catch (TankFullException ex) { fillTank=false;}

        }

        System.out.println(myTank.getTankFuel());
        myTank=null;
    }
}
```

Ενδεικτικός Κώδικας της *fuelTank*:

```
public void fuelTank(double fuelTempo)
    throws InvalidFuelTempoException, TankFullException {
    if ((fuelTempo!=0.1)&& (fuelTempo!=0.2))
        throw new InvalidFuelTempoException("Sorry. Fuel tempo is not supported.");

    if(gasolineTankCapacity==tankFuel) throw new InvalidFuelTempoException("Thank you
    else {
```

Ενδεικτικός Κώδικας της *TankFullException*:

```
class TankFullException extends Exception
{
    public TankFullException (String str)
    {
        super(str);
    }
}
```

Σημείωση: Μπορείτε να δημιουργήσετε και επιπρόσθετες εξαιρέσεις. Για παράδειγμα αν ο χρήστης της κλάσης επιχειρήσει να δημιουργήσει ένα ρεζερβουάρ με αρνητική τιμή χωρητικότητας καυσίμων η κλάση *TankFuel* τι πρέπει να κάνει; Προσπαθήστε να αλλάξετε την τιμή της ιδιότητας *gasolineTankCapacity* σε οποιαδήποτε setter ή getter μέθοδο. Τι παρατηρείτε;