**Αντικειμενοστραφής Προγραμματισμός - Java**

**ΚΑΡΑΓΙΑΝΝΗΣ ΓΕΩΡΓΙΟΣ - 2015508**

Για τις ανάγκες τις εργασίας δημιουργήθηκαν 5 διαφορετικές κλάσσεις:

* Trains.class

Η κλάσση αυτή περιέχει τη μέθοδο main. H main, δημιουργει ένα νέο αντικείμενο τύπου TFrame και εμφανίζει το κύριο παράθυρο της εφαρμογής μας.

* TFrame.class

Η κλάσση TFrame κάνει extend (κληρονομεί από) την κλάσση JFrame και περιέχει τις εξής μεθόδους:

* Public TFrame:

Αποτελεί τον Constructor της κλάσσης. Αρχικά ορίζει τις διαστάσεις και τη θέση του Frame που δημιουργείται. Στη συνέχεια, και αφού ορίσει τον Container του Frame, δημιουργεί ένα TCanvas αντικείμενο και δύο JButtons και τα προσαρμόζει στο Cintainer. Για τη δημιουργία των Buttons καλείται η μέθοδος addButton (βλ. παρακάτω) ενώ o εκάστοτε ActionListener ορίζεται από μία Anonymous Inner Class που δίνεται σαν όρισμα στη μέθοδο addButton().

* Public void addTrain:

Η μέθοδος αυτή δημιουργεί τέσσερα αντικέιμενα τύπου Train (τα τέσσερα τρένα της εργασίας) ορίζοντας τις παραμέτρους του καθενός (χρώμα, ταχύτητα, επιτάχυνση). Για κάθε ένα train που δημιουργεί, δημιουργεί και ένα TThread (ένα νέο Thread) που θα είναι “υπεύθυνο” για τη συμπεριφορά (κίνηση) ενός τρένου.

* Public void addButton:

Δημιουργεί ένα νέο JButton και το προσθέτει στον Container του Frame. Τέλος, προσθέτει στο νέο button το ActionListener που παίρνει σαν όρισμα κατά την κλήση της

* TCanvas.class

Η κλάσση TCanvas κάνει extend (κληρονομεί από) την κλάσση JPanel. Περιέχει την private ArrayList μεταβλητή που διατηρεί κάθε ένα από τα Train objects που έχουν δημιουργηθεί και περιέχει τις εξής μεθόδους:

* Public void add:

Παίρνει σαν όρισμα ένα Train object και το προσθέτει στο private ArrayList

* Public void paintComponent:

H standard μέθοδος που χρειάζεται πάντα για να σχεδιαστεί ένα γραφικό στο Panel. Αρχικά καλεί την paintComponent της υπερκλάσσης JPanel και στη συνέχεια τρέχει το όλο το μήκος του ArrayList και καλεί τη μέθοδο draw (βλ. παρακάτω) για κάθε ένα Train object του ArrayList

* Public int getTrainsNum:

Επιστρέφει το μέγεθος του ArrayList και κατά συνέπεια το πλήθος των Train objects που έχουν δημιουργηθεί.

* Public Train getTrain:

Παίρνει σαν όρισμα ένα ακέραιο και επιστρέφει το Train object που βρίσκεται στη θέση αυτή του ArrayList

* Train.class

Η κλάσση αυτή περιγράφει τις ιδιότητες και τη συμπεριφορά ενός τρένου. Οι ιδιότητες ενός Train object περιγράφονται από τις μεταβλητές:

* Distance: η απόσταση που έχει διανύσει
* x και y: περιγράφουν τη θέση του τρένου
* XSIZE και YSIZE: περιγράφουν το μέγεθος του τρένου (χρησιμοποιούνται κυρίως για σχεδιαστικούς λόγους)
* tColor: περιγράφει το χρώμα του τρένου (για σχεδιαστικούς λόγους)
* prevTrain: αποτελεί reference στο προηγούμενο Train object

**ΣΗΜΕΙΩΣΗ:** η υλοποίηση έγινε πριν διαβάσω για Iterator. Γνωρίζοντας για τους Iterators ίσως η λειτουργικότητα που προσπαθούσα να υλοποιήσω, να μπορούσε να γίνει με χρήση Iterator στο ArrayList του TCanvas πιο σωστά. Πάντως, it still works

* Speed: περιγράφει την αρχική ταχύτητα και το μέσο όρο αυτής
* Dx: περιγράφει τη στιγμιαία ταχύτητα του Train object
* Ddx: περιγράφει, κατά κάποιο τρόπο, τη στιγμιαία επιτάχυνση. Ουσιαστικά περιγράφει το νέο ποσό ταχύτητας (θετικό ή αρνητικό) που θα προστεθεί στη στιγμιαία, σε μία εκτέλεση της TThread.run (βλ. παρακάτω)

Οι μέθοδοι της κλάσσης είνια οι εξής

* Public Train:

Αποτελεί τον Cinstructor της κλάσσης. Δημιουργεί ένα νέο Train object με γνωρίσματα που παίρνει ως όρισμα

* Public Train:

Αποτελεί επίσης έναν Constructor της κλάσσης. Η διαφορά με τον προηγούμενο είναι στο πλήθος των ορισμάτων, όπου ο παρών παίρνει ένα επιπλέον όρισμα το reference στο προηγούμενο Train object

* Private void setValues:

Ορίζει τα γνωρίσματα ενός Train object (χρώμα, ταχύτητα, επιτάχυνση)

* Public void draw:

Δημιουργεί ένα οβαλ γραφικό με τις διαστάσεις και το χρώμα του train object

* Public void move:

Περιγράφει όλη τη λογική της κίνησης ενός Train οbject. Αυξάνει το distance με βάση τη στιγμιαία τα χύτητα. Ορίζει τη νέα θέση του στο TCanvas. Ορίζει την επιτάχυνση ελέγχοντας την απόσταση από το προηγούμενο Train object, ώστε το τρέχον Train object να σταματήσει για τυχαίο χρονικό διάστημα, να ελαττώσει την ταχύτητά του ή να την αυξήσει, κατά τυχαίο τρόπο.

* Public double getDistance:

Παίρνει σαν όρισμα ένα Train object και επιστρέφει την απόσταση από το Train object από το οποίο καλούμε τη μέθοδο

* TThread.class

Η κλάσση αυτή κάνει extend τη κλάσση Τhread και δημιουργεί ένα thread για τηδιαχείριση ενός Train object. Περιέχει μία private μεταβλητή τύπου Τrain για το Train object που πρέπει να διαχειριστεί και τις παρακάτω μεθόδους:

* Public TThread:

Αποτελεί τον Cinstructor της κλάσσης. Δημιουργεί ένα TThread το οποίο είναι υπεύθυνο για το Train object που παίρνει σαν όρισμα. To επιπλέον boolean όρισμα καθορίζει εάν το TThread θα ξεκινήσει αμέσως ή θα περιμένει πρώτα για 3 δευτερόλεπτα.

**ΣΗΜΕΙΩΣΗ:** Δυστυχώς αυτός ο τρόπος δημιουργεί το εξής πρόβλημα. Μόλις πατάμε το κουμπί Start, το πρόγραμμα μπαίνει σε μία λούπα μέχρι να δημιουργηθούν και να αρχίσουν να κινούνται τα τέσσερα ζητούμενα Train objects. Μέχρι να ολοκληρωθέι η λούπα αυτή το πρόγραμμα δεν ανταποκρίνεται στο πάτημα άλλων κουμπιών. Ωστόσο cach-άρει τα actions στα πλήκτρα και αυτά θα ενεργήσουν μόλις το πρόγραμμα βγει από τη λούπα. Αυτό δημιουργείτε λόγω της καθυστέρησης των 3 δευτερολέπτων για κάθε ένα από τα τρία επόμενα TThreads που δημιουργούνται, εκτός από το πρώτο. Δυστυχώς ο περιορισμένος χρόνος που είχα δεν μου επέτρεψε να ασχοληθώ περεταίρω ώστε να βρω μία κλύτερη λύση. But it still works..

* Public void run:

H standard μέθοδος που καλείται όσο ένα thread είναι ζωντανό. Εδώ καλείται η μέθοδος Train.move για την κίνηση του Train object, κάθε 2 ms.

* Public TThread getCurrentTThread:

Επιστρέφει ως reference το τρέχον TThread