

Time series Segmentation

Παρατίθεται το κείμενο προδιαγραφής απαιτήσεων.

Έχοντας διαβάσει μόνο τις προδιαγραφές (και όχι τη λύση που δίνεται παρακάτω)

Ζήτημα 0. Βρείτε τα use cases, και παραθέστε για καθένα (α) όνομα, (β) βασικούς actors, (γ) μια περιγραφή της μίας πρότασης

Ζήτημα 1: ΧΩΡΙΣ να δείτε παρακάτω, έχοντας δει ΜΟΝΟ την εκφώνηση, σχεδιάστε το διάγραμμα κλάσεων.

Ζήτημα 2: Για κάθε κλάση που φτιάξατε, συμπληρώστε στο διπλανό πίνακάκι αν είναι domain, business logic, client

<i>Class</i>	<i>Class type</i>
<i>SlopeManager</i>	<i>Business Logic</i>
...	...

Ζήτημα 3: Επιβεβαιώστε ότι για κάθε use case έχετε και μία μέθοδο που την διεκπεραιώνει, συμπληρώνοντας τον επόμενο πίνακα:

<i>Use case</i>	<i>Method</i>	<i>Class</i>	<i>Class type</i>	<i>Comment</i>
<i>computeSlope</i>	<i>reportSlopeToClient()</i>	<i>SlopeManager</i>	<i>Business Logic</i>	<i>... αν χρειάζεται ...</i>
...

Προχωρήστε παρακάτω και δείτε τη λύση που δίνεται για το υποσύστημα ανάλυσης

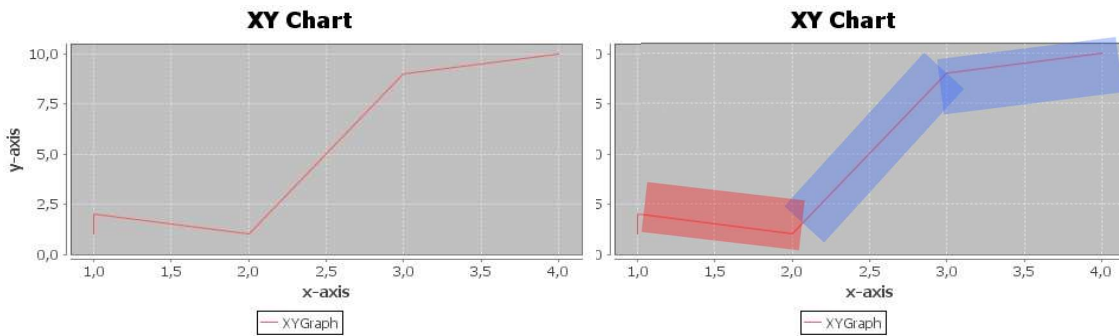
Ζήτημα 4 και 5: δείτε μέσα

Τέλος, δείτε τη συνολική λύση

Ζήτημα 6. Προσθέστε ελέγχους στη λύση (είτε τη δική σας, είτε αυτή που σας δίδεται). Απαιτείται τουλάχιστον η OREOS περιγραφή, και ιδεατά και η τετράδα pre-conditions, input, output assertions, post-condition assertions

Διευκρινίζεται ότι παραδοσιακά, στο διαγώνισμα τα ζητούμενα θέματα είναι της κατηγορίας 0, 1, 3 και 6.

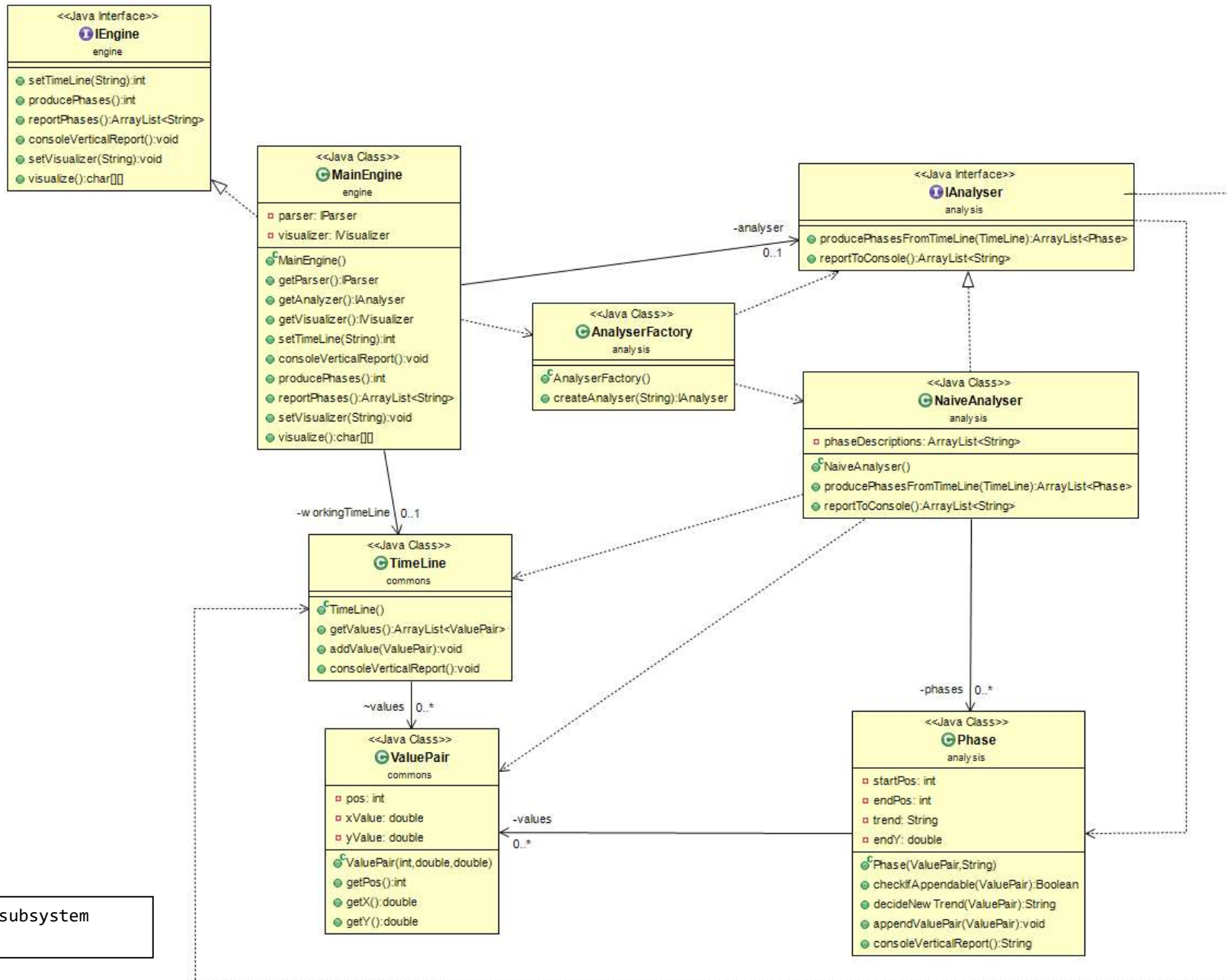
Έχουμε ένα project στο οποίο πρέπει να φτιάξουμε μια μηχανή η οποία να λαμβάνει ως είσοδο μια χρονοσειρά και να εξάγει φάσεις της χρονοσειράς. Μια χρονοσειρά είναι μια λίστα από σημεία μέτρησης, με κάθε σημείο να είναι ένα ζεύγος (α/α , μέτρηση). Μια φάση είναι ένα σύνολο συνεχόμενων σημείων με ίδια τάση. Στο παρακάτω παράδειγμα, φαίνεται η βασική ιδέα: κάθε σημείο έχει απεικονισθεί σε ένα διδιάστατο διάγραμμα, με τον α/α να απεικονίζεται στον οριζόντιο και τη μέτρηση στον κάθετο άξονα (αριστερό διάγραμμα) και οργανώνεται σε μια κόκκινη, πτωτική φάση και σε μια μπλε, αύξουσα φάση.



Το σύστημά μας θέλουμε να υποστηρίζει τις εξής λειτουργίες:

- Φόρτωση δεδομένων με επεκτάσιμο τρόπο (και συγκεκριμένα, εδώ, τουλάχιστον με (i) ένα απλό φόρτωμα από μια μέθοδο, και (ii) με επεξεργασία από αρχείο). Στην πρώτη περίπτωση απαιτείται μια απλή μέθοδος που θα κατασκευάζει και θα αναπαριστά μια μικρή χρονοσειρά (κυρίως για να χρησιμοποιηθεί σε περιπτώσεις ελέγχων). Στη δεύτερη περίπτωση, απαιτείται ο χρήστης να προσδιορίσει ένα αρχείο εισόδου από το οποίο τα δεδομένα θα φορτωθούν στο σύστημα και θα αναπαρασταθούν. Το αρχείο εισόδου θεωρούμε ότι έχει μια μέτρηση με 3 στοιχεία: $\langle \alpha/\alpha \rangle \langle \text{χρονόσημο} \rangle \langle \text{μέτρηση} \rangle$ σε κάθε γραμμή. Το χρονόσημο και το $\langle \alpha/\alpha \rangle$ μπορεί να ταυτίζονται αν ο ανθρώπινος χρόνος δεν είναι διαθέσιμος.
- Ανάλυση της χρονοσειράς σε φάσεις (επίσης με επεκτάσιμο τρόπο – εδώ με ένα απλό αλγόριθμο, αλλά στο μέλλον με αλγόριθμους που, π.χ., αγνοούν μικρά spikes). Ο βασικός αλγόριθμος στηρίζεται στην ιδέα ότι κάθε φάση έχει μια τάση (ανοδική, καθοδική, ή επίπεδη). Ο αλγόριθμος επεξεργάζεται ένα ένα τα σημεία της χρονοσειράς. Ας υποθέσουμε, για να καταλάβουμε τον αλγόριθμο, ότι βρισκόμαστε στη μέση της χρονοσειράς, και η τρέχουσα φάση είναι η Φ . Έστω, δε, ότι μόλις προσθέσαμε στη χρονοσειρά το σημείο στη θέση k , και πάμε να προσθέσουμε το σημείο $k+1$. Αν η τάση της φάσης Φ διατηρείται με την προσθήκη του νέου σημείου, τότε το σημείο $k+1$ προστίθεται στην υπάρχουσα φάση. Αν όχι, τότε ξεκινάμε μια νέα φάση. Η διατήρηση της τάσεως είναι απλή: αν η τάση είναι καθοδική (ανοδική/επίπεδη) και η μέτρηση του σημείου $k+1$ είναι μικρότερη (μεγαλύτερη/ίση) από την μέτρηση του σημείου k , τότε η τάση διατηρείται.
- Παρουσίαση της χρονοσειράς, επίσης με επεκτάσιμο τρόπο (εδώ: (i) στην κονσόλα και (ii) σε μια html σελίδα). Η παρουσίαση περιλαμβάνει την οπτικοποίηση των 2 αξόνων, X και Y , με τον άξονα X να αντιστοιχεί στο χρόνο και τον Y στην μετρούμενη ποσότητα και την παρουσίαση των σημείων μέτρησης της χρονοσειράς ως σημεία στο διδιάστατο χώρο που προκύπτει από τους 2 άξονες.
- Παρουσίαση των φάσεων που δημιουργούνται από την χρονοσειρά. Για κάθε φάση, καταγράφουμε ένα μήνυμα «Νέα φάση με τάση ...» ανάλογα με την τάση, και στη συνέχεια παρατίθενται τα σημεία της φάσεως, το ένα μετά το άλλο, το καθένα στη δικιά του γραμμή.

ΧΡΗΣΙΜΟΠΟΙΗΣΤΕ ΤΗ ΣΕΛΙΔΑ ΓΙΑ ΤΟ ΔΙΑΓΡΑΜΜΑ ΚΛΑΣΕΩΝ



The analysis subsystem

Ζήτημα 4. Δείτε το παρόν απόσπασμα της λύσης, που αφορά στο υποσύστημα ανάλυσης. Πώς θα διορθώνατε:

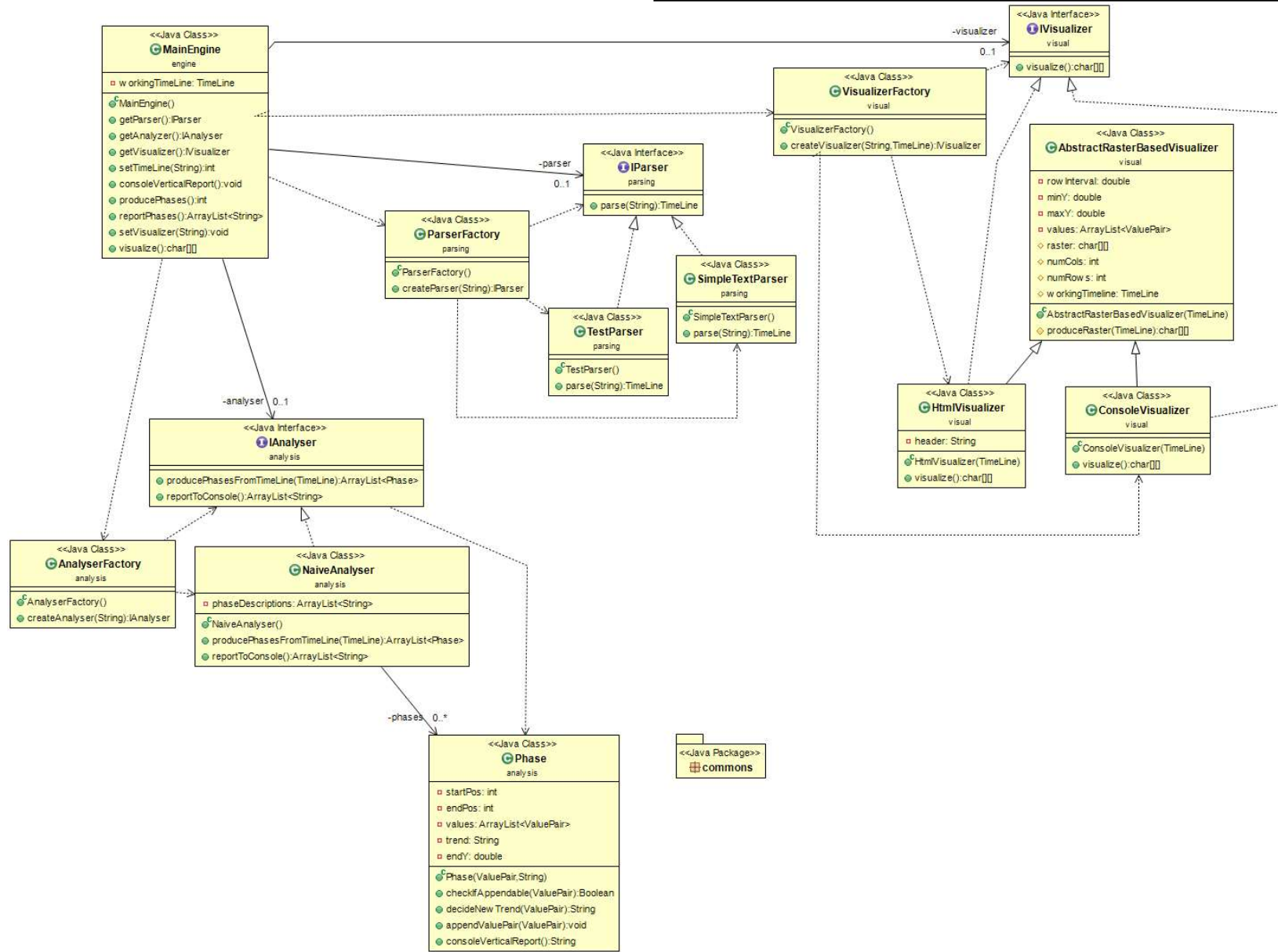
- Την ορατότητα των μεθόδων
- Την υπογραφή των μεθόδων (ονόματα, παράμετροι, τύπος επιστροφής)
- Την τοποθέτηση των μεθόδων (υπάρχουν μέθοδοι, δηλ., που δε θα έπρεπε να είναι εκεί ή είναι στη λάθος κλάση?)

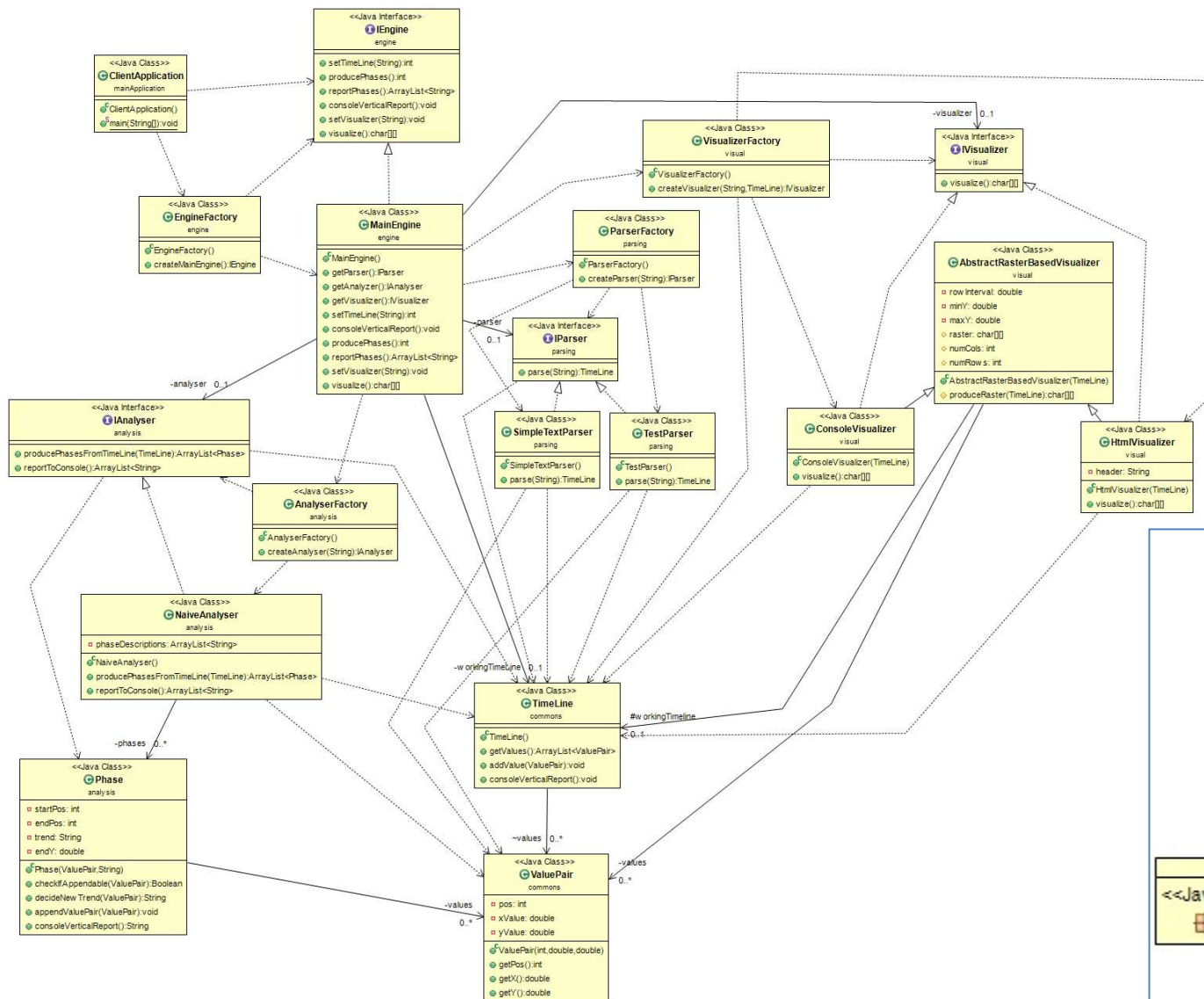
Checklist and hints for method placement (it is not obligatory that there exist misplaced methods in the given solution – redo the tables of Topics 2 and 3):

- *Is it reasonable to have methods implementing use cases not placed in business logic classes?*
- *Is it reasonable to publicly export methods at the Interface level, irrelevant to the package?*
- *Is it reasonable to have classes exporting methods as public, if they are not going to be used?*

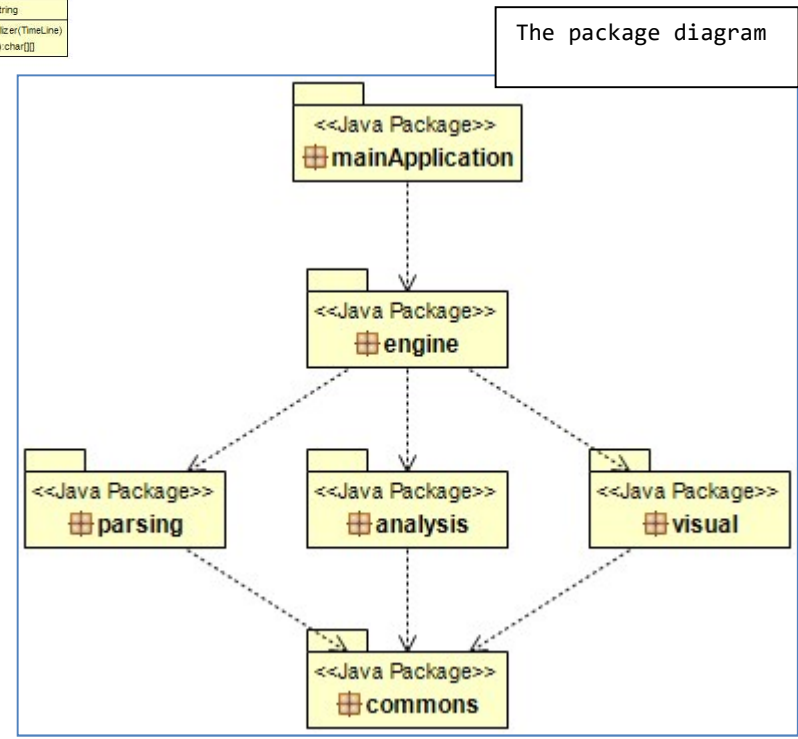
Ζήτημα 5. ΧΩΡΙΣ ΝΑ ΔΕΙΤΕ την συνολική λύση, επισκεφτείτε ξανά τη λύση σας και προσθέστε Interfaces & Packages

Ένα απλό διάγραμμα με ένα υποσύνολο των κλάσεων για τα βασικά στοιχεία του project





The full-blown class diagram



The package diagram