

Εργασία 3

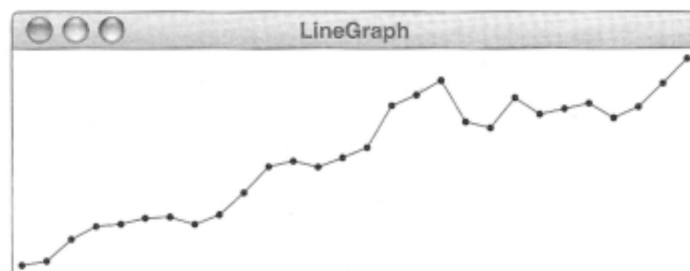
Ημερομηνία παράδοσης: 04/2/2017

Γενικές Οδηγίες παράδοσης εργασιών

1. Οι εργασίες είναι ατομικές.
2. Δεν επιτρέπεται η χρήση άλλων βιβλιοθηκών / πακέτων εκτός από τα προκαθορισμένα της Java και το `acm.*`.
3. Η χρήση του `acm.*` είναι απαραίτητη.
4. Δεν επιτρέπεται η χρήση «έτοιμων» μεθόδων/μπλοκ κώδικα που να λύνουν σημαντικό μέρος του προβλήματος. Το πρόγραμμα σας θα πρέπει να σχεδιάσετε και να υλοποιήσετε εσείς.
5. Απαγορεύεται η χρήση ελληνικών χαρακτήρων στον κώδικα, τα σχόλια και τα ονόματα των αρχείων. Ενθαρρύνεται η χρήση αγγλικών σε αυτά τα σημεία και όχι greeklish.
6. Θα παραδώσετε τα αρχεία κώδικα που θα έχετε γράψει (δηλ. αυτά με κατάληξη `.java`, όχι με `.class`) συμπιεσμένα σε ένα αρχείο (`.zip` ή `.rar` και όχι σε άλλη μορφή, με όνομα τον Α.Μ. σας, π.χ., `3010023.rar`) μέσω της πλατφόρμας του `eclass`. Συγκεκριμένα, στο χώρο του μαθήματος, στην περιοχή των Εργασιών στον κατάλογο Τρίτη προγραμματιστική άσκηση θα ανεβάσετε το συμπιεσμένο αρχείο της εργασίας σας.
7. Προσοχή! Ο κώδικας που θα υποβάλετε θα πρέπει να μεταγλωττίζεται και να τρέχει από την γραμμή εντολών και όχι από κάποιο άλλο εργαλείο/προγραμματιστικό περιβάλλον.

Άσκηση 1 (άσκηση 7, σελίδα 548, βιβλίο του Eric Roberts)

Όταν οι στατιστικολόγοι θέλουν να απεικονίσουν τη συμπεριφορά κάποιας ποσότητας που μεταβάλλεται με το χρόνο (π.χ. τη μέση τιμή της παγκόσμιας θερμοκρασίας σε ένα έτος, χρηματιστηριακούς δείκτες κτλ) μία απλή παρουσίαση αυτών των δεδομένων είναι το γράφημα γραμμής (line graph) στο οποίο σχεδιάζουμε τις τιμές των δεδομένων σε ένα επίπεδο x/y και έπειτα συνδέουμε κάθε γειτονικό ζευγάρι σημείων με μια ευθεία γραμμή. Σε τέτοια διαγράμματα ο άξονας x υποδηλώνει το χρόνο και ο άξονας y την τιμή των δεδομένων. Για παράδειγμα, το παρακάτω διάγραμμα αποτελεί ένα γράφημα γραμμής όπου τα σημεία δεδομένων αντιπροσωπεύουν τη μέση μηνιαία τιμή της αξίας μιας μετοχής της Google, Inc για κάποιο χρονικό διάστημα.



Γράψτε ένα πρόγραμμα **GraphicsProgram** με το όνομα **LineGraph** το οποίο θα διαβάζει ένα σύνολο αριθμητικών δεδομένων τύπου `double` από το χρήστη (το πλήθος των δεδομένων μπορεί επίσης να δίνεται από το χρήστη), θα τα αποθηκεύει σε ένα `array` και έπειτα θα παράγει το γράφημα γραμμής. Συγκεκριμένα στο πρόγραμμά σας θα πρέπει να υλοποιήσετε μια μέθοδο **drawLineGraph** που θα δέχεται ως είσοδο ένα `array` τύπου `double` και θα κατασκευάζει το γράφημα. Η υλοποίηση της μεθόδου **drawLineGraph** θα πρέπει να προσαρμόζει τους άξονες έτσι ώστε τα σημεία δεδομένων να καταλαμβάνουν ολόκληρο το χώρο στο εσωτερικό του παραθύρου εκτός από ένα μικρό περιθώριο σε κάθε πλευρά, όπως φαίνεται και στο παραπάνω σχήμα (δείγμα εκτέλεσης). Συγκεκριμένα, η κλίμακα στο άξονα x (δηλ. η απόσταση που απέχουν δύο διαδοχικά δεδομένα κατά τον οριζόντιο άξονα) θα ισούται με το διαθέσιμο οριζόντιο χώρο (δηλ. μείον τα μικρά περιθώρια) δια το πλήθος των δεδομένων πλην ένα. Για να προσαρμόσετε την κλίμακα στον άξονα y θα πρέπει απλώς να εξασφαλίσετε ότι όλα τα δεδομένα θα εμφανιστούν στον καμβά, το οποίο με τη σειρά του απαιτεί να υπολογίσετε την ελάχιστη και μέγιστη τιμή των τιμών του `array`.

Άσκηση 2 (άσκηση 5, σελίδα 547, βιβλίο του Eric Roberts)

Μια ευρέως διαδεδομένη σπαζοκεφαλιά λογικής ονομάζεται Σουντόκου. Στο Σουντόκου ξεκινάτε με ένα πλέγμα (δισδιάστατο πίνακα) αριθμών μεγέθους 9x9, στο οποίο ορισμένα από τα κελιά έχουν συμπληρωθεί με ψηφία μεταξύ του 1 και 9. Για να λύσετε την σπαζοκεφαλιά πρέπει να συμπληρώσετε κάθε ένα από τα κενά κελιά με ένα ψηφίο μεταξύ του 1 και 9 έτσι ώστε κάθε ψηφίο να εμφανίζεται ακριβώς μία φορά σε κάθε γραμμή, κάθε στήλη και σε κάθε ένα από τα μικρότερα τετράγωνα μεγέθους 3x3. Κάθε παιχνίδι Σουντόκου είναι προσεκτικά σχεδιασμένο ώστε να έχει μία μόνο λύση. Για παράδειγμα, δοθέντος του πλέγματος που φαίνεται στα αριστερά, η μοναδική λύση είναι αυτή στα δεξιά.

		2	4		5	8		
	4	1	8				2	
6				7			3	9
2				3			9	6
		9	6		7	1		
1	7			5				3
9	6			8				1
	2				9	5	6	
		8	3		6	9		

3	9	2	4	6	5	8	1	7
7	4	1	8	9	3	6	2	5
6	8	5	2	7	1	4	3	9
2	5	4	1	3	8	7	9	6
8	3	9	6	2	7	1	5	4
1	7	6	9	5	4	2	8	3
9	6	7	5	8	2	3	4	1
4	2	3	7	1	9	5	6	8
5	1	8	3	4	6	9	7	2

Γράψτε μία μέθοδο της μορφής

```
private boolean checkSudokuSolution(int[][] grid)
```

η οποία θα παίρνει ως είσοδο ένα δισδιάστατο array που αναπαριστά μια πιθανή λύση Σουντόκου και θα ελέγχει αν όντως είναι λύση (οπότε θα επιστρέφει true) ή δεν αποτελεί λύση (θα επιστρέφει false). Επίσης γράψτε και ένα κυρίως πρόγραμμα επίδειξης όπου θα καλείτε την μέθοδο για κάποιο παράδειγμα πιθανής λύσης Σουντόκου.