



## Θ.Ε. ΠΛΣ50 (2017-18) – ΓΡΑΠΤΗ ΕΡΓΑΣΙΑ Ε1

Ημερομηνία ανάρτησης	21.10.2017
Ημερομηνία αποστολής	<b>Βάσει χρονοδιαγράμματος: 12.11.2017</b> <b>Δεκτή μέχρι την Τετάρτη 15.11.2017, ώρα 23:55</b>  (προσοχή, το σύστημα υποβολής θα κλείσει αυτόματα μόλις παρέλθει η παραπάνω χρονική στιγμή, σύμφωνα με το ρολόι του συστήματος, που μπορεί να διαφέρει ελαφρά από το δικό σας)
Ανακοίνωση ενδεικτικής επίλυσης	17.11.2015

### Θεματολογία-στόχος

Στην εργασία αυτή θα εξασκηθείτε σε μερικές από τις απαραίτητες γνώσεις της γλώσσας προγραμματισμού Java, που θα χρησιμοποιήσουμε καθ' όλη τη χρονιά σε πολλές από τις εργασίες της ΘΕ. Ειδικότερα, θα ασχοληθείτε με τύπους και μεταβλητές, τελεστές, εκφράσεις, είσοδο από το πληκτρολόγιο και έξοδο στην οθόνη, εντολές ελέγχου ροής και ανακύκλωσης, μονοδιάστατους πίνακες, αλφαριθμητικά, ορισμό κλάσεων, μεθόδους και πέρασμα παραμέτρων, κατασκευαστές και αντικείμενα, είσοδο από / έξοδο σε αρχείο κειμένου και μαθηματικές συναρτήσεις.

### Παρατηρήσεις

Περιμένουμε όλες οι εργασίες να ανεβούν στο [study.eap.gr](http://study.eap.gr). Ο πηγαίος κώδικας Java (**ένα ή περισσότερα αρχεία .java για κάθε θέμα ή υποερώτημα**) που θα συνοδεύει την εργασία θα πρέπει να βρίσκεται σε ξεχωριστούς υποκαταλόγους (ένας υποκατάλογος για κάθε θέμα ή υποερώτημα). Μην συμπεριλάβετε αρχεία .class, ή άλλα αρχεία που τυχόν δημιουργούνται από το Ολοκληρωμένο Περιβάλλον Ανάπτυξης (IDE) που χρησιμοποιείτε. Συνιστάται να συμπεριλάβετε επεξηγήσεις για τον τρόπο που λύσατε την εργασία **σε συνοδευτικό αρχείο κειμένου μορφής \*.doc ή \*.odt** (αρχεία \*.pdf γίνονται δεκτά μόνο όταν συνοδεύονται από το αντίστοιχο doc/odt). Στο [study.eap.gr](http://study.eap.gr), σε κάθε περίπτωση, ανεβάζετε **ένα μόνο συμπιεσμένο αρχείο** (\*.zip ή \*.rar) που θα περιέχει όλους τους επιμέρους υποκαταλόγους και αρχεία.

Μπορείτε να χρησιμοποιήσετε το Ολοκληρωμένο Περιβάλλον Ανάπτυξης (IDE), π.χ. BlueJ, Eclipse, Netbeans, της αρεσκείας σας. Εκτός από ενδεχομένως άλλο IDE, συνιστάται να εγκαταστήσετε και το BlueJ και να δοκιμάσετε ότι ο κώδικάς σας λειτουργεί σωστά και σε αυτό.

### Εισαγωγή

Δε χρειάζεται να κάνετε τίποτε περισσότερο από όσα σας ζητούνται. Συνιστάται, πριν ασχοληθείτε με την εργασία, να μελετήσετε τα παραδείγματα, τις ασκήσεις αυτοαξιολόγησης και τις δραστηριότητες από τα κεφάλαια του Τόμου Β', Γλώσσες Προγραμματισμού, ΕΑΠ, Πάτρα 2015, που αντιστοιχούν στις εβδομάδες μελέτης μέχρι την ημερομηνία αποστολής της εργασίας βάσει χρονοδιαγράμματος. Στο τέλος της Εργασίας περιλαμβάνεται **ενδεικτικός** πίνακας με τα βασικά γνωστικά αντικείμενα της γλώσσας προγραμματισμού Java που πρέπει να έχετε διαβάσει για να μπορείτε να αντιμετωπίσετε κάθε θέμα. Τονίζεται ότι ο πίνακας έχει σκοπό να σας βοηθήσει και όχι να σας υποχρεώσει στον ακριβή τρόπο υλοποίησης κάθε θέματος.

Τα μηνύματα που τυπώνουν τα προγράμματά σας στην οθόνη συνιστάται να είναι γραμμένα με λατινικούς χαρακτήρες (greeklish ή αγγλικά), προς αποφυγή προβλημάτων που σχετίζονται με την κωδικοποίηση των ελληνικών στο λειτουργικό σύστημα του υπολογιστή σας.

### Θέμα 1: Παραγγελία πίτσας

Είστε μια παρέα από  $X$  άτομα και θέλετε να παραγγείλετε πίτσες. Κάθε άτομο θα φάει είτε μία ατομική πίτσα, είτε μισή οικογενειακή. Θα παραγγείλετε είτε α)  $X$  ατομικές πίτσες, ή β) το πάνω ακέραιο μέρος του  $X/2$  οικογενειακές πίτσες. Θεωρείστε ότι η ποσότητα μιας πίτσας εκφράζεται από την επιφάνειά

της (το εμβαδόν της) και ότι μια πίτσα είναι ένας κύκλος. Η πιτσαρία σας γνωστοποιεί τη διάμετρο και την τιμή της οικογενειακής και της ατομικής πίτσας. Ποια παραγγελία, η (α) ή η (β), έχει τον καλύτερο (μικρότερο) λόγο συνολικής τιμής / συνολική επιφάνεια που θα φαγωθεί;

Να γράψετε στη γλώσσα προγραμματισμού Java πρόγραμμα που θα αποτελείται από μία κλάση με όνομα Pizza και

- θα ζητά από τον χρήστη να εισάγει από το πληκτρολόγιο τη διάμετρο (σε εκατοστά) και την τιμή (σε ευρώ) της οικογενειακής και της ατομικής πίτσας (ως δεκαδικούς αριθμούς), το πλήθος X των ατόμων (ως ακέραιο αριθμό), θα υπολογίζει ποια παραγγελία, η (α) ή η (β), έχει τον καλύτερο λόγο συνολικής τιμής / συνολική επιφάνεια που θα φαγωθεί και θα τυπώνει στην οθόνη μήνυμα που θα υποδεικνύει την πιο συμφέρουσα παραγγελία (π.χ. order 10 personal pizzas, the price/surface rate is 0.00796).

Επισήμανση: Καθώς κάθε άτομο θα φάει είτε μία ατομική πίτσα, είτε μισή οικογενειακή πίτσα, μια οικογενειακή πίτσα δε θα φαγωθεί ολόκληρη αν το πλήθος των ατόμων είναι μονός αριθμός.

## Θέμα 2: Εγκυρότητα ΑΦΜ

Ένας Αριθμός Φορολογικού Μητρώου (Α.Φ.Μ.) αποτελείται από 9 αριθμητικά ψηφία. Για να πιστοποιήσουμε αν ένας αριθμός με ψηφία  $A_1A_2A_3A_4A_5A_6A_7A_8A_9$  είναι έγκυρος Α.Φ.Μ., ακολουθούμε τον εξής αλγόριθμο:

Υπολογίζουμε το άθροισμα  $S = 2^8 \times A_1 + 2^7 \times A_2 + 2^6 \times A_3 + 2^5 \times A_4 + 2^4 \times A_5 + 2^3 \times A_6 + 2^2 \times A_7 + 2^1 \times A_8$

Υπολογίζουμε το υπόλοιπο Y της διαίρεσης του S με τον αριθμό 11.

Αν Y ισούται με 10 και  $A_9$  ισούται με 0, τότε ο αριθμός είναι έγκυρος Α.Φ.Μ.

Διαφορετικά, αν  $A_9$  ισούται με Y, τότε ο αριθμός είναι έγκυρος Α.Φ.Μ.

Διαφορετικά, ο αριθμός δεν είναι έγκυρος Α.Φ.Μ.

Να γράψετε πρόγραμμα στη γλώσσα προγραμματισμού Java που θα αποτελείται από μία κλάση με όνομα AFM και

- θα διαβάζει από το πληκτρολόγιο ένα αλφαριθμητικό,
- θα ελέγχει ότι το μήκος του αλφαριθμητικού αντιστοιχεί σε Α.Φ.Μ. και αν όχι θα ενημερώνει το χρήστη με κατάλληλο μήνυμα και θα επαναλαμβάνει το βήμα της ανάγνωσης του αλφαριθμητικού από το πληκτρολόγιο,
- κάνοντας χρήση ανακύκλωσης, θα απομονώνει κάθε χαρακτήρα του αλφαριθμητικού με τη μέθοδο `charAt`, θα ελέγχει αν ο χαρακτήρας αυτός είναι αριθμητικό ψηφίο με τη μέθοδο `isDigit` (και αν όχι θα ενημερώνει τον χρήστη με κατάλληλο μήνυμα και θα επαναλαμβάνει το βήμα της ανάγνωσης του αλφαριθμητικού από το πληκτρολόγιο), θα μετατρέπει τον χαρακτήρα σε ακέραιο με τη μέθοδο `getNumericValue` και θα υπολογίζει τον επόμενο όρο του αθροίσματος S κάνοντας χρήση της μεθόδου `row` της κλάσης `Math`,
- θα υπολογίζει το υπόλοιπο Y της διαίρεσης του S με τον αριθμό 11,
- θα κάνει τους κατάλληλους ελέγχους ως προς το υπόλοιπο και θα ενημερώνει με κατάλληλο μήνυμα τον χρήστη για το αν ο Α.Φ.Μ είναι έγκυρος,
- θα ρωτά τον χρήστη αν θέλει να επαναλάβει τη διαδικασία ή θέλει να τερματίσει το πρόγραμμα.

Σημείωση: έγκυρος Α.Φ.Μ. δε σημαίνει και υπαρκτός Α.Φ.Μ. Με το πρόγραμμά σας ελέγχετε μόνο την εγκυρότητα ενός Α.Φ.Μ.

### Θέμα 3: Έλεγχος / Δημιουργία IBAN

Ο IBAN (International Bank Account) είναι μια τυποποιημένη μορφή ενός Τραπεζικού Λογαριασμού. Ο IBAN συντίθεται και επαληθεύεται με συγκεκριμένη μέθοδο και κανόνες. Μπορείτε να διαβάσετε περισσότερες λεπτομέρειες για τον IBAN στο εγχειρίδιο:

[http://www.hba.gr/2Tomeis/UplDocs/protypa/IBAN\\_ET\\_03\\_2007.pdf](http://www.hba.gr/2Tomeis/UplDocs/protypa/IBAN_ET_03_2007.pdf)

Να ορίσετε την κλάση IBAN που θα περιλαμβάνει

- μια μεταβλητή στιγμιότυπου grIBAN, τύπου πίνακα χαρακτήρων, στον οποίο θα διατηρείται ένας ελληνικός IBAN,
- έναν κατασκευαστή που θα δέχεται ως παράμετρο ένα αλφαριθμητικό και θα αρχικοποιεί κατάλληλα το grIBAN,
- μια μέθοδο checkIBAN που θα επαληθεύει αν τα ψηφία ελέγχου είναι τέτοια ώστε το grIBAN να είναι έγκυρο ελληνικό IBAN και θα επιστρέφει κατάλληλη boolean τιμή,
- μια μέθοδο makeIBAN που θα υπολογίζει και θα διορθώνει τα ψηφία ελέγχου του grIBAN ώστε να καταστεί έγκυρος ελληνικός IBAN,
- μια μέθοδο getIBAN που θα επιστρέφει ως string το grIBAN.

Ορίστε μια κλάση IBANApp που θα περιλαμβάνει μια μέθοδο checkFormat, η οποία θα δέχεται ως παράμετρο ένα string και θα ελέγχει ότι το string αυτό έχει το κατάλληλο πλήθος χαρακτήρων και ότι οι 2 πρώτοι χαρακτήρες του string είναι κατάλληλοι για ελληνικό IBAN, θα ελέγχει ότι οι υπόλοιποι χαρακτήρες του string είναι αριθμητικά ψηφία και θα επιστρέφει κατάλληλη boolean τιμή.

Η κλάση IBANApp θα περιλαμβάνει τη μέθοδο main η οποία θα διαβάζει από το αρχείο κειμένου inIBAN.txt σε μεταβλητή string μία-μία τις γραμμές του αρχείου και θα καλεί την checkFormat για αυτή τη μεταβλητή. Αν ο έλεγχος της checkFormat δεν είναι επιτυχής, θα τυπώνει κατάλληλο ενημερωτικό μήνυμα στην οθόνη και η main θα συνεχίζει με την επόμενη γραμμή του αρχείου. Αν ο έλεγχος είναι επιτυχής, θα δημιουργεί ένα αντικείμενο IBAN για αυτό το string και θα καλεί την checkIBAN. Αν ο έλεγχος της checkIBAN δεν είναι επιτυχής, θα καλεί για αυτό το string την makeIBAN. Στη συνέχεια, θα καλεί την getIBAN, θα εμφανίζει το IBAN στην οθόνη και θα ρωτά το χρήστη αν επιθυμεί αποθήκευση του IBAN σε αρχείο. Αν η απάντηση είναι θετική, θα προχωρά στην αποθήκευση σε ξεχωριστή γραμμή του αρχείου κειμένου outIBAN.txt. Η λειτουργία της main θα συνεχίζεται μέχρι να διαβαστούν όλες οι γραμμές του αρχείου κειμένου inIBAN.txt.

Σημειώσεις: Το πρόγραμμα που σας ζητείται δε στοχεύει στο να ελέγξει την εγκυρότητα του τμήματος BBAN του IBAN. Καθώς η ύλη μας δεν έχει προχωρήσει μέχρι το κατάλληλο σημείο, το πρόγραμμα αυτό και τα προηγούμενα της 1<sup>ης</sup> εργασίας δε χρειάζεται να κάνουν χειρισμό εξαιρέσεων.

#### Γνωστικά Αντικείμενα ανά θέμα

	Θ1	Θ2	Θ3
Τύποι και μεταβλητές, τελεστές, εκφράσεις	X	X	X
Είσοδος πληκτρολογίου και έξοδος οθόνης	X	X	X
Εντολές ελέγχου ροής	X	X	X
Επανάληψη		X	X
Μονοδιάστατοι πίνακες			X
Αλφαριθμητικά (strings)		X	X



Μέθοδοι και πέρασμα παραμέτρων			X
Κατασκευαστές			X
Περισσότερες από μία κλάσεις και αντικείμενα			X
Είσοδος από / έξοδος σε αρχείο κειμένου			X
Μαθηματικές μέθοδοι	X	X	X

ΚΡΙΤΗΡΙΑ ΑΞΙΟΛΟΓΗΣΗΣ	
Θέμα 1: Παραγγελία πίτσας	20
Θέμα 2: Εγκυρότητα ΑΦΜ	30
Θέμα 3: Έλεγχος / Δημιουργία IBAN	50
Εικόνα εργασίας - σχολιασμός	10
<b>ΣΥΝΟΛΟ</b>	<b>110</b>
Ο συνολικός βαθμός θα διαιρεθεί δια 10, ώστε να προκύψει ο τελικός βαθμός της εργασίας.	

**Καλή Επιτυχία**