

1^ο ατομικό φυλλάδιο ασκήσεων

Οδηγίες

- Η παράδοση των εργασιών θα γίνει ηλεκτρονικά μέσω του *eclass* την *Κυριακή 12/11/2017* στις *23:59*.
- Πρέπει να στείλετε ένα αρχείο zip με όνομα `loginName_Home01.zip` (π.χ., `icsd16001_Home01.zip`) στο οποίο θα περιέχονται τα εξής:
 - Project των ασκήσεων. Θα πρέπει να συμπεριλαμβάνονται όλα τα αρχεία του project. Στην αρχή του πηγαίου κώδικα θα αναγράφεται ο αριθμός μητρώου, το ονοματεπώνυμό σας και ο αριθμός της άσκησης. Ο κώδικάς σας *πρέπει να περιέχει σχόλια*.
 - Ένα αρχείο pdf με οθόνες εκτέλεσης (screenshots) των προγραμμάτων σας, στο οποίο να φαίνεται ξεκάθαρα το αποτέλεσμα της εκτέλεσης του κάθε προγράμματος. Στην αρχή του αρχείου θα αναγράφεται ο αριθμός μητρώου και το ονοματεπώνυμό σας.

Άσκηση 1 – Κυκλική περιστροφή αλφαριθμητικού

Αν ένα αλφαριθμητικό x ταυτίζεται με ένα αλφαριθμητικό y όταν οι χαρακτήρες του περιστραφούν κυκλικά για ένα τυχαίο αριθμό θέσεων, τότε λέμε ότι το x αποτελεί μια κυκλική περιστροφή του y . Για παράδειγμα, το αλφαριθμητικό `faceinter` είναι μια κυκλική περιστροφή του `interface` ή το `actabstr` είναι μια κυκλική περιστροφή του `abstract` (και το αντίστροφο, φυσικά).

Γράψτε ένα πρόγραμμα που δέχεται δύο Strings από το πληκτρολόγιο και ελέγχει αν το ένα είναι κυκλική περιστροφή του άλλου.

Bonus: Αν ο έλεγχος υλοποιηθεί σε μια γραμμή μόνο με τη χρήση των `indexOf()` και `length()` της κλάσης `String` και της συνένωσης `String` (concatenation).

Άσκηση 2 - Κουμπαράς

Σε αυτήν την άσκηση καλείστε να υλοποιήσετε έναν εικονικό κουμπαρά στον οποίο αποθηκεύονται χρήματα για τις διακοπές σας. Ο κουμπαράς (κλάση `Piggy`) χαρακτηρίζεται από τα χρήματα που περιέχει. Δημιουργείτε στην κλάση σας ένα αντίστοιχο πεδίο για το σκοπό αυτό με το όνομα `poson`.



Δημιουργήστε δύο constructors. Ο πρώτος *δε θα* δέχεται κάποιο όρισμα και θα δημιουργεί έναν άδειο κουμπαρά. Ο δεύτερος δέχεται ως όρισμα ένα αρχικό ποσό και αρχικοποιεί κατάλληλα το διαθέσιμο υπόλοιπο. Πρέπει να γίνεται έλεγχος αν το αρχικό ποσό είναι θετικό.

Αν όχι, τότε πρέπει να δημιουργείται ένας κουμπαράς με μηδενικό υπόλοιπο.

Οι μέθοδοι που πρέπει να υλοποιήσετε είναι:

- `epistrotifiYpologiou`: επιστρέφει το υπόλοιπο που υπάρχει στον κουμπαρά.
- `katathesi`: προσθέτει στον κουμπαρά το ποσό που περνάμε ως παράμετρο. Μόνο θετικά ποσά είναι αποδεκτά, διαφορετικά δεν συμβαίνει τίποτα.
- `adeiasma`: αδειάζει τον κουμπαρά μηδενίζοντας το υπόλοιπο.
- `analipsi`: επιτρέπει την ανάληψη χρημάτων από τον κουμπαρά, αφαιρώντας από το υπόλοιπο το ποσό που δίνεται ως παράμετρος. Αν το ποσό είναι μεγαλύτερο από το υπόλοιπο, τότε ο κουμπαράς αδειάζει εντελώς. Αν το ποσό είναι αρνητικό, τότε αγνοείται. Η μέθοδος δεν επιστρέφει κάποια τιμή.

- ε. `tarakounise`: τυπώνει στο τερματικό το μήνυμα `Cling cling!!` και μια νέα γραμμή μόνο στην περίπτωση που ο κουμπαράς έχει κάποιο υπόλοιπο. Διαφορετικά δεν κάνει τίποτα.
- στ. `υροlogiseΥρολοιρο`: η μέθοδος επιστρέφει τη διαφορά μεταξύ του ποσού στον κουμπαρά και του ποσού που σκοπεύουμε να ξοδέψουμε στις διακοπές (παράμετρος). Το υπόλοιπο δεν επηρεάζεται. Αν το ποσό που σκοπεύουμε να ξοδέψουμε είναι αρνητικό (ή μηδέν) τότε η μέθοδος επιστρέφει το ποσό του κουμπαρά.
- ζ. `toString`: επιστρέφει το String `"Den γραχει υρολοιρο"` αν ο κουμπαράς σας είναι άδειος, ή το String `"Ο koumparas sas periei <amount> euro"` όπου `<amount>` είναι το υπόλοιπο του κουμπαρά σας. Η μέθοδος κάνει override την αντίστοιχη μέθοδο της Object.

Στο παραδοτέο παρουσιάστε τη λειτουργία κάθε μεθόδου καθώς και τι συμβαίνει όταν κάνετε `print` τον κουμπαρά σας.



Άσκηση 3 – Βιβλιοθήκη

Σκοπός της άσκησης είναι η προσομοίωση της λειτουργίας μιας βιβλιοθήκης. Η βιβλιοθήκη περιέχει αντίγραφα από βιβλία γραμμένα από κάποιο συγγραφέα. Το πρόγραμμά σας πρέπει αρχικά να είναι σε θέση να δημιουργεί συγγραφείς, τα βιβλία που αυτοί έχουν γράψει και να καταχωρεί στη βιβλιοθήκη αντίγραφα των βιβλίων. Έπειτα, θα πρέπει να είναι σε θέση:

- να τυπώνει όλα τα αντίγραφα που βρίσκονται στη βιβλιοθήκη.
- να τυπώνει όλα τα αντίγραφα που είναι γραμμένα σε μια συγκεκριμένη γλώσσα.
- να τυπώνει τα ονόματα των συγγραφέων που έχουν βραβευτεί και για τους οποίους η βιβλιοθήκη διαθέτει ένα αντίγραφο βιβλίου τους.
- να τυπώνει τον αριθμό των αντιγράφων ενός συγκεκριμένου βιβλίου.

Για την υλοποίηση θα χρειαστεί να δημιουργήσετε 4 κλάσεις:

- 1) Κλάση `Author`. Κάθε συγγραφέας χαρακτηρίζεται από το όνομά του (μεταβλητή τύπου `String`) και από μια `boolean` μεταβλητή που δείχνει αν έχει βραβευτεί ή όχι. Ο `constructor` της κλάσης πρέπει να δέχεται και τα δύο πεδία με αυτή τη σειρά. Μέθοδοι:

- α. Η μέθοδος `getName` που επιστρέφει το όνομα του συγγραφέα.
- β. Η μέθοδος `getPrize` που επιστρέφει `true` αν ο συγγραφέας έχει βραβευτεί.

- 2) Κλάση `Book`. Κάθε βιβλίο χαρακτηρίζεται από τον τίτλο του (μεταβλητή τύπου `String`), τον συγγραφέα του (μεταβλητή τύπου `Author`) και από τη γλώσσα στην οποία είναι γραμμένο (`String`). Ο `constructor` δέχεται τις παραμέτρους με την εξής σειρά: τίτλος, συγγραφέας, γλώσσα. Αν η γλώσσα δεν παρέχεται τότε πρέπει να πάρει τη `default` τιμή `"english"` (αγγλικά). Οι μέθοδοι αυτής της κλάσης, οι οποίες αποτελούν και μέρος της διεπαφής χρήσης είναι:

- α. `getTitle` που επιστρέφει τον τίτλο του βιβλίου.
- β. `getAuthor` που επιστρέφει το όνομα του συγγραφέα (όχι το αντικείμενο συγγραφέας.)
- γ. `getLang` που επιστρέφει τη γλώσσα στην οποία είναι γραμμένο το βιβλίο.

- δ. `toString (@Override)` που επιστρέφει την αναπαράσταση του βιβλίου σε `String` με την εξής μορφή: <τίτλος>, <όνομα συγγραφέα>, <γλώσσα> ακολουθούμενο από μια νέα γραμμή.
- 3) Κλάση `Copy`. Η κλάση χρησιμεύει στο να αναπαραστήσει το αντίγραφο ενός βιβλίου. Κάθε ομότυπο της κλάσης αυτής χαρακτηρίζεται από (μια αναφορά σε) ένα βιβλίο `Book`. Ο constructor της κλάσης που δέχεται σαν όρισμα (μια αναφορά σε) ένα `Book` και τυπώνει το εξής μήνυμα: `New copy -> <τίτλος>, <όνομα συγγραφέα>, <γλώσσα>`. Μέθοδοι:
- α. Η μέθοδος `getBook` που επιστρέφει -προφανώς- το βιβλίο, τύπου `Book`
 - β. `toString (@Override)` που επιστρέφει την αναπαράσταση του αντιγράφου σε `String` με την εξής μορφή: `Copy of: <τίτλος>, <όνομα συγγραφέα>, <γλώσσα>` ακολουθούμενο από μια νέα γραμμή.
- 4) Κλάση `Library`. Μια βιβλιοθήκη χαρακτηρίζεται από το όνομά της (τύπου `String`) και περιέχει μια λίστα από αντίγραφα βιβλίων (`Copy`). Ο constructor που δέχεται σαν όρισμα το όνομα της βιβλιοθήκης και τυπώνει το μήνυμα: `The library <όνομα> has opened!` ακολουθούμενο από νέα γραμμή. Οι μέθοδοι της κλάσης πρέπει να είναι:
- α. Η μέθοδος `getName` που επιστρέφει το όνομα της βιβλιοθήκης.
 - β. Η μέθοδος `getNbOfCopies` που επιστρέφει τον αριθμό των αντιγράφων που υπάρχουν στη βιβλιοθήκη.
 - γ. Η μέθοδος `stock` που επιτρέπει την προσθήκη αντιγράφων στη βιβλιοθήκη. Πρέπει να δέχεται τις εξής παραμέτρους: την αναφορά σε ένα βιβλίο `Book` και τον αριθμό `N` των αντιτύπων που θέλουμε να προσθέσουμε. Αυτή η μέθοδος θα δημιουργεί κάθε αντίγραφο του βιβλίου και θα το προσθέτει στη λίστα από `Copy(ies)` της βιβλιοθήκης. Αν ο χρήστης δεν παρέχει τον αριθμό των αντιγράφων, τότε εννοείται προσθήκη ενός αντιγράφου μόνο (υπερφόρτωση).
 - δ. Η μέθοδος `listCopies` που επιστρέφει μια λίστα με όλα τα αντίγραφα των βιβλίων που είναι γραμμένα σε μια συγκεκριμένη γλώσσα (την οποία η μέθοδος δέχεται ως παράμετρο). Αν δε δίνεται γλώσσα, τότε η λίστα που επιστρέφεται πρέπει να περιέχει όλα τα αντίτυπα της βιβλιοθήκης.
 - ε. Η μέθοδος `countCopies` που επιστρέφει τον αριθμό των αντιγράφων ενός βιβλίου `Book`, το οποίο δέχεται ως όρισμα.
 - στ. Η μέθοδος `printAuthor` με παράμετρο τύπου `boolean` ή χωρίς παράμετρο, η οποία τυπώνει τα ονόματα των συγγραφέων των οποίων τα βιβλία υπάρχουν ως αντίγραφα στη βιβλιοθήκη. Αν η παράμετρος δίνεται και είναι `true`, τυπώνονται μόνο τα ονόματα των συγγραφέων που έχουν βραβευθεί. Αν είναι `false` τυπώνονται τα ονόματα των μη βραβευμένων συγγραφέων. Αν δεν παρέχεται παράμετρος στη μέθοδο, η μέθοδος θα τυπώνει μόνο βραβευμένους συγγραφείς.

Για την υλοποίηση της λίστας χρησιμοποιείτε μια δομή που θα σας επιτρέπει να προσθέτετε σε αυτή πολλαπλά αντίτυπα ενός αντικειμένου. Στη `main` μέθοδο του παραδοτέου πρέπει να περιέχεται κώδικας που να επιτρέπει τον έλεγχο όλων των παραπάνω μεθόδων.

Άσκηση 4 – Internet Slang

Γράψτε μια εφαρμογή η οποία «μεταφράζει» προτάσεις που εισάγει ο χρήστης, οι οποίες χρησιμοποιούν την αργκό του ίντερνετ σε προτάσεις που χρησιμοποιούν κοινή γλώσσα. Για παράδειγμα:



DAE use the main function as a test client for the class? -> *Does anyone else* use the main function as a test client for the class?

Java Strings are immutable FYI. -> Java Strings are immutable *for your information*.

Θα πρέπει να δίνεται η επιλογή στο χρήστη για αντίστροφη μετάφραση από την κοινή γλώσσα στη slang. Παράδειγμα:

In case you missed it there is a deadline for the exercises -> ICYMI there is a deadline for the exercises

Το πρόγραμμα σας θα πρέπει να κάνει χρήση ενός μικρού λεξικού χρησιμοποιώντας κατάλληλη δομή δεδομένων και μεθόδων της κλάσης StringBuilder.