Οικονομικό Πανεπιστήμιο Αθηνών, Τμήμα Πληροφορικής
Μάθημα: Εισαγωγή στον Προγραμματισμό Υπολογιστών
Ακαδημαϊκό έτος: 2016–17

Εργασία 2

Ημερομηνία παράδοσης: 4/01/2017

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 Πρώτη 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

Στην άσκηση αυτή θα κατασκευάσετε ένα ψηφιακό ρολόϊ. Το ρολόϊ θα αναπαρίσταται στο πρόγραμμά σας ως ένα αντικείμενο της τάξης `Clock` την οποία θα πρέπει πρώτα να ορίσετε και υλοποιήσετε σύμφωνα με τις παρακάτω προδιαγραφές.

Τα αντικείμενα `Clock` θα πρέπει να δέχονται αποκλειστικά τα εξής μηνύματα:
`toString()`: επιστρέφει την ώρα ως `String` στη μορφή ΩΩ:ΛΛ:ΔΔ, όπου ΩΩ είναι τα δύο ψηφία που δείχνουν την ώρα, ΛΛ τα ψηφία των λεπτών και ΔΔ αυτά των δευτερολέπτων.
`setHour(int h)`: θέτει την ώρα στην τιμή `h`. (Θεωρήστε ότι πάντα ισχύει $0 \leq h \leq 23$.)
`setMin(int m)`: θέτει τα λεπτά στην τιμή `m`. (Θεωρήστε ότι πάντα ισχύει $0 \leq m \leq 59$.)
`setSec(int s)`: θέτει τα δευτερόλεπτα στην τιμή `s`. (Θεωρήστε ότι πάντα ισχύει $0 \leq s \leq 59$.)
`tick()`: προχωράει το ρολόϊ κατά 1 δευτερόλεπτο

Για παράδειγμα, οι εντολές

```
Clock clock = new Clock();
clock.setHour(16);
clock.setMin(28);
clock.setSec(58);
for ( int i = 0; i <= 3; i++ ) {
 println(clock.toString());
 clock.tick();
}
```

θα πρέπει να εμφανίζουν:

```
16:28:58

16:28:59

16:29:00

16:29:01
```

Θα πρέπει επίσης να υλοποιήσετε ένα κυρίως πρόγραμμα το οποίο θα αρχικοποιεί το ρολόϊ στην ώρα 16:28:58 (όπως στο παράδειγμα παραπάνω) και κατόπιν θα εμφανίζει τη νέα ώρα κάθε δευτερόλεπτο (πραγματικού χρόνου) που περνάει, για τα επόμενα 3 λεπτά. Μπορείτε να κάνετε χρήση της έτοιμης συνάρτησης `pause(int ms)` η οποία «παγώνει» την εκτέλεση του προγράμματός σας για `ms` χιλιοστά του δευτερολέπτου.

Άσκηση 2

Σε αυτή την άσκηση θα εξελίξετε την τάξη `Clock` ορίζοντας και υλοποιώντας την τάξη `AMPMClock`, στην οποία θα υπάρχει η επιλογή εμφάνισης της ώρας με την ένδειξη πμ ή μμ (δηλαδή προ/μετά μεσημβρίας).

Συγκεκριμένα, τα αντικείμενα `AMPMClock` θα δέχονται επιπλέον το εξής μήνυμα: `setAMPM(boolean yes)`: Στα επόμενα μηνύματα `toString()`, η ώρα που επιστρέφεται χρησιμοποιεί την ένδειξη πμ ή μμ εάν η `yes` έχει τιμή `true`. Στην περίπτωση που η `yes` έχει τιμή `false`, η ώρα δε θα χρησιμοποιεί την ένδειξη πμ ή αμ (δηλ. όπως γίνονταν και στην `Clock`).

Για παράδειγμα, οι εντολές

```
AMPMClock clock = new AMPMClock();

clock.setHour(16);

clock.setMin(28);

clock.setSec(58);

println(clock.toString());

clock.setAMPM(true);

println(clock.toString());

clock.tick(); clock.tick();

println(clock.toString());

clock.setAMPM(false);

clock.tick();

println(clock.toString());
```

θα πρέπει να εμφανίζουν:

```
16:28:58

04:28:58 μμ

04:29:00 μμ

16:29:01
```

Θα πρέπει επίσης να υλοποιήσετε ένα κυρίως πρόγραμμα το οποίο θα αρχικοποιεί το ρολόϊ στην ώρα 04:28:58 μμ και κατόπιν θα εμφανίζει τη νέα ώρα σε μορφή πμ/μμ κάθε δευτερόλεπτο (πραγματικού χρόνου) που περνάει (πάλι για τα επόμενα 3 λεπτά).

Αξιοποιήστε όσο τον δυνατόν περισσότερο την κληρονομικότητα έτσι ώστε Α) να μην αλλάξετε καθόλου τον ορισμό της τάξης `Clock`. Θα πρέπει να χρησιμοποιήσετε τον ίδιο ακριβώς ορισμό που δώσατε στην άσκηση 1. Β) να μην επαναλάβετε λειτουργικότητα της `Clock` στον ορισμό της τάξης `AMPMClock`.