

Δεύτερη Σειρά Ασκήσεων
Ημερομηνία Παράδοσης: 31 Μαΐου 2019, 21:00 μ.μ.

Στην άσκηση αυτή θα σχεδιάσετε και θα υλοποιήσετε ένα σύστημα το οποίο θα προσομοιώνει ένα σύστημα ελέγχου πυρκαγιάς (ΣΕΠ) σε ένα κτήριο. Κατά τη διάρκεια εκτέλεσης του προγράμματος ο χρήστης θα έχει τις εξής δυνατότητες:

- Εισαγωγή στο ΣΕΠ μιας τοποθεσίας του κτηρίου. Κάθε νέα τοποθεσία θα χαρακτηρίζεται από ένα μοναδικό κωδικό (τύπου int). Δεν γνωρίζετε εκ των προτέρων πόσες τοποθεσίες έχει ένα κτήριο. Επίσης κάθε τοποθεσία θα περιέχει 2 αισθητήρες. Καθένας από τους αισθητήρες θα χαρακτηρίζεται από ένα μοναδικό κωδικό. Ο πρώτος αισθητήρας θα μετράει τη θερμοκρασία και ο δεύτερος θα μετράει τη φωτεινότητα. Η θερμοκρασία θα είναι μια τυχαία ακέραια τιμή από 0 έως 100 (π.χ. 20, 45, 56, 78, κλπ.). Η φωτεινότητα θα είναι μια τυχαία πραγματική τιμή από 0 έως 200 (π.χ. 56.78, 103.67, 200.00, κλπ.). Ο κάθε αισθητήρας θα χαρακτηρίζεται επίσης από μια τιμή που θα καθορίζει το πόσο αξιόπιστος είναι. Η τιμή της αξιοπιστίας είναι η πιθανότητα να μετράει ο κάθε αισθητήρας λανθασμένες τιμές με την πάροδο του χρόνου t . Πιο συγκεκριμένα η τιμή της αξιοπιστίας για κάθε αισθητήρα (sensor) υπολογίζεται από τον παρακάτω τύπο:

$$reliability_{sensor} = e^{-\lambda * t}, \quad \text{όπου } \lambda = 10^{-3} \text{ και } t = 1, 2, 3, \dots, n \quad (1)$$

- Ανανέωση των μετρήσεων της θερμοκρασίας και της φωτεινότητας για τις υπάρχουσες τοποθεσίες του κτηρίου. Ο χρήστης θα καλεί αυτή τη λειτουργία και για κάθε τοποθεσία θα παράγονται από το ΣΕΠ τυχαίες τιμές θερμοκρασίας και φωτεινότητας. Κατά την ανανέωση των μετρήσεων θα γίνεται και ανανέωση των τιμών της αξιοπιστίας για κάθε αισθητήρα με βάση τον τύπο που δόθηκε παραπάνω. Θεωρήστε ότι κατά την i -οστή ανανέωση ο χρόνος t ισούται με i .
Πριν πραγματοποιηθεί η i -οστή ανανέωση των μετρήσεων θα καταγράφονται σε ένα αρχείο του ΣΕΠ με όνομα measures.txt, οι παλιές μετρήσεις για κάθε μία τοποθεσία του κτηρίου. Η αποθήκευση θα γίνεται κατά αύξοντα κωδικό τοποθεσίας. Σημειώστε ότι στο αρχείο θα αποθηκεύονται μόνο οι μετρήσεις που ισχύουν πριν από την i -οστή ανανέωση και όχι άλλες προηγούμενες.
- Έλεγχος ύπαρξης πυρκαγιάς σε μια τοποθεσία του κτηρίου που θα δίνεται από το χρήστη. Το ΣΕΠ θα αναφέρει ύπαρξη πυρκαγιάς εάν ισχύουν ταυτόχρονα τα εξής τρία χαρακτηριστικά:
 - Η θερμοκρασία στη συγκεκριμένη τοποθεσία ξεπερνά την τιμή 50.
 - Η φωτεινότητα στη συγκεκριμένη τοποθεσία ξεπερνά την τιμή 140.
 - Η συνολική αξιοπιστία των μετρήσεων είναι μεγαλύτερη του 0.9. Η συνολική αξιοπιστία των δύο αισθητήρων sensor1 και sensor2 ισούται με:

$$reliability_{total} = reliability_{sensor\ 1} * reliability_{sensor\ 2}$$

(2)

Πιο συγκεκριμένα για το σύστημα ελέγχου πυρκαγιάς (ΣΕΠ) ζητούνται τα παρακάτω:

1. Αναγνωρίστε και σχεδιάστε τις κλάσεις του συστήματος.
2. Σχεδιάστε το UML διάγραμμα του συστήματος.
3. Αναγνωρίστε και σχεδιάστε τις σχέσεις κληρονομικότητας και σύνθεσης/συνάθροισης μεταξύ των κλάσεων.
4. Υλοποιήστε τις κλάσεις.
 - a. Κάθε κλάση να έχει τουλάχιστον ένα copy-constructor και ένα constructor με παραμέτρους αντίστοιχες με τα πεδία της κλάσης. Επίσης πρέπει να έχετε μεθόδους `equals()` και `toString()`.
 - b. Για την εγγραφή σε αρχείο θα μιλήσουμε αργότερα στο μάθημα μέχρι τότε μπορείτε να εκτυπώνετε τα πάντα στην στάνταρτ έξοδο του υπολογιστή.
 - c. Για τον υπολογισμό του `exp` στην εξίσωση (1) χρησιμοποιήστε τη μέθοδο `static double exp (double x)` της `java` που δηλώνεται στο `java.lang.Math`.
 - d. Για τον έλεγχο του συστήματός σας στη κλάση `SEP`, η οποία θα υλοποιεί τη `main`, θα πρέπει να ζητάτε επαναληπτικά από τον χρήστη να κάνει μία επιλογή: (1) εισαγωγή μιας τοποθεσίας στο κτήριο, (2) ανανέωση των μετρήσεων των αισθητήρων, (3) έλεγχος για πυρκαγιά, (4) έξοδος. Παρακάτω φαίνεται ένα παράδειγμα επαναληπτικής εκτέλεσης του προγράμματος:

```
1)Eisagogh mias topothesias sto ktirio.
2)Aneaneosh ton metriseon.
3)Elegxos pyrkagias.
4)Exodos.
Choose: 1
```

```
1)Eisagogh mias topothesias sto ktirio.
2)Aneaneosh ton metriseon.
3)Elegxos pyrkagias.
4)Exodos.
Choose: 1
```

```
1)Eisagogh mias topothesias sto ktirio.
2)Aneaneosh ton metriseon.
3)Elegxos pyrkagias.
4)Exodos.
Choose: 1
```

```
1)Eisagogh mias topothesias sto ktirio.
2)Aneaneosh ton metriseon.
3)Elegxos pyrkagias.
4)Exodos.
Choose: 2
```

```
1)Eisagogh mias topothesias sto ktirio.
2)Aneaneosh ton metriseon.
3)Elegxos pyrkagias.
4)Exodos.
Choose: 2
```

1)Eisagogh mias topothesias sto ktirio.
2)Aneaneosh ton metriseon.
3)Elegxos pyrkagias.
4)Exodos.
Choose: 3

Thermorkrasia: 11; Fotinotita: 38.21; Synoliki aksiopistia: 0.40;
Den yparxei pyrkagia.

1)Eisagogh mias topothesias sto ktirio.
2)Aneaneosh ton metriseon.
3)Elegxos pyrkagias.
4)Exodos.
Choose: 3

Thermorkrasia: 44; Fotinotita: 53.12; Synoliki aksiopistia: 0.50;
Den yparxei pyrkagia.

1)Eisagogh mias topothesias sto ktirio.
2)Aneaneosh ton metriseon.
3)Elegxos pyrkagias.
4)Exodos.
Choose: 4

Ένα τυπικό παράδειγμα του αρχείου measures.txt που θα πρέπει να παραχθεί με βάση το παραπάνω παράδειγμα εκτέλεσης θα πρέπει να μοιάζει κάπως έτσι:

Area: 1; Thermorkrasia 13; Fotinotita 49.29; Xronos 2;
Area: 2; Thermorkrasia 90; Fotinotita 122.09; Xronos 2;
Area: 3; Thermorkrasia 65; Fotinotita 38.04; Xronos 2;

Υπενθύμιση: Οι τιμές της θερμοκρασίας και της φωτεινότητας είναι τυχαίες τιμές στα διαστήματα που περιγράφονται παραπάνω στην εκφώνηση, επομένως αναμένεται να έχετε διαφορετικές τιμές για τις μεταβλητές αυτές σε σχέση με αυτές που παρουσιάζονται στο παράδειγμα που σας δίδεται.

Οδηγίες υλοποίησης/παράδοσης

Για τη σχεδίαση των UML διαγραμμάτων μπορείτε είτε να τα σχεδιάσετε σε χαρτί και να τα σκανάρετε σε μορφή pdf είτε να χρησιμοποιήσετε ένα από τα παρακάτω εργαλεία:

- [Eclipse UML Papyrus plugin](#)
- [Eclipse Violet UML plugin](#)

Μαζί με τον κώδικά σας θα παραδώσετε το pdf με το UML διάγραμμα του συστήματος και ένα αρχείο report.txt που περιγράφετε τι έχετε κάνει και πώς να τρέξουμε τον κώδικά σας. Στην αναφορά σας πρέπει να αναφέρετε και τα προβλήματα του κώδικά σας, π.χ. σε ποιες περιπτώσεις δεν δουλεύει σωστά. Εργασίες στις οποίες τα προβλήματα του κώδικα περιγράφονται με ακρίβεια θα πάρουν +10% του βαθμού που θα παίρνανε κανονικά.

Στην υλοποίηση των κλάσεων σας δεν θα πρέπει να έχετε public πεδία. Βαθμοί θα αφαιρεθούν για προγράμματα που δεν είναι καλά γραμμένα, δηλαδή δεν είναι σωστά στοιχισμένα ή δεν έχουν καλά επιλεγμένα ονόματα μεταβλητών ώστε να διαβάζονται εύκολα.

Προγράμματα που δεν κάνουν compile θα πάρουν το πολύ 1 μονάδα από τις 10. Θα πάρετε περισσότερες μονάδες αν έχετε ολοκληρωμένες σωστές μεθόδους, ή μια ολόκληρη κλάση η οποία είναι σωστή, παρά αν έχετε πολύ κώδικα που όμως δεν κάνει compile ή δεν τρέχει.

Τα αρχεία που θα παραδώσετε θα ελεγχθούν για αντιγραφές μέσω ειδικού λογισμικού. Οι αντιγραφές μηδενίζονται για όλα τα εμπλεκόμενα άτομα.

Στον κώδικα να αναγράφονται σε σχόλια το όνομα, το login και ο ΑΜ σας. Χρησιμοποιείτε μόνο λατινικούς χαρακτήρες αλλιώς δεν λειτουργεί το turnin. «Ορφανές» ασκήσεις δεν θα βαθμολογούνται.

Η παράδοση των ασκήσεων γίνεται **ΑΠΟΚΛΕΙΣΤΙΚΑ ΚΑΙ ΜΟΝΟ μέσω turnin**.

Θα κάνετε **turnin** την άσκησή σας στο **assignment2@myy205**. Θα παραδώσετε τον πηγαίο κώδικα μαζί με το pdf που θα έχει το UML διάγραμμα και το αρχείο report.txt.

Π.χ. turnin assignment2@myy205 file1.java file2.java file3.java file4.java UML.pdf report.txt ... κτλ.

Προσοχή: το turnin μπορεί να δεχτεί αρχεία μέχρι 10MB, οπότε, αν το αρχείο σας ξεπερνά το όριο, σπάστε το σε πολλά.

Μετά την ημερομηνία παράδοσης δεν θα γίνονται δεκτές οι υποβολές των εργασιών. **Σε καμία περίπτωση δεν θα γίνουν δεκτές εργασίες μέσω email.** Φροντίστε έγκαιρα να έχετε τους κωδικούς του λογαριασμού σας στη σχολή.