

#### Οδηγίες χρήσης:

- Για προσθήκη χώρου στο σύστημα πυρασφάλειας, πιάστε το 1 και (Enter), έπειτα εισαγάγετε τον κωδικό του δωματίου και (Enter).
- Για ανανέωση τιμών και εγγραφή των προηγούμενων σε αρχείο, πιάστε το 2 και (Enter), το αρχείο βρίσκεται στον φάκελο που βρίσκεται και ο πηγαίος κώδικας.
- Για έλεγχο πυρκαγιάς, πιάστε το 3 και (Enter) και στην συνέχεια τον κωδικό του δωματίου και (Enter), και θα εκτυπωθεί στην οθόνη η κατάσταση για το συγκεκριμένο δωμάτιο.
- Για έξοδο πιάστε το 4 και (Enter).

#### Γνωστά προβλήματα του συστήματος/προβλήματα υλοποίησης:

-Παράγεται ένας τυχαίος αριθμός για αναγνωριστικό αισθητήρα, ο αριθμός αυτός είναι ένας τυχαίος ακέραιος από το 0 έως το 999,999,999 (1 δις πιθανές τιμές). Υπάρχει μία (πολύ) μικρή πιθανότητα 2 (ή/και περισσότεροι) αισθητήρες να έχουν το ίδιο αναγνωριστικό.

#### Λειτουργία προγράμματος:

##### SEP.java

Υλοποιεί την main, έχοντας μία δυναμική λίστα (ArrayList) με αντικείμενα τύπου Room, ένα αντικείμενο τύπου Scanner για να δέχεται στοιχεία από τον χρήστη, ένα αντικείμενο τύπου String που θα κρατά την επιλογή του χρήστη (αρχικοποιημένο σε μηδέν) και ένα αντικείμενο τύπου PrintWriter που θα χρειαστεί για να γράψουμε στο αρχείο measures.txt.

Αρχικά τυπώνεται η λίστα με τις διαθέσιμες επιλογές για τον χρήστη και ζητά να επιλέξει κάτι από τις διαθέσιμες επιλογές.

Στην 1η επιλογή, ο χρήστης θα πρέπει να εισάγει έναν κωδικό δωματίου, στη συνέχεια δημιουργείται ένα νέο δωμάτιο (με τυχαίες τιμές αισθητήρων), αν το δωμάτιο υπάρχει στην λίστα με τα δωμάτια, το νέο δωμάτιο δεν θα προστεθεί, αν δεν υπάρχει θα προστεθεί.

Στην 2η επιλογή, η λίστα ταξινομείται με βάση τον κωδικό δωματίου κατά αύξουσα σειρά, έπειτα δημιουργείται ένα νέο αρχείο, το measures.txt, όπου θα προσθέσει τα δεδομένα της λίστας (με μία συγκεκριμένη μορφοποίηση) και μετά ανανεώνονται ένα-ένα τα στοιχεία της λίστας με νέες τιμές αισθητήρων (στο αρχείο θα αποθηκευτούν οι προηγούμενες τιμές, ενώ αν υπήρχαν πριν δεδομένα θα χαθούν).

Στην 3η επιλογή, ο χρήστης εισάγει το δωμάτιο που θέλει να ελέγξει για πυρκαγιά, αν το δωμάτιο υπάρχει στη λίστα θα τυπώσει την κατάσταση του δωματίου, ενώ αν εισάγει 0, θα τυπώσει ελέγξει και θα τυπώσει την κατάσταση όλων των δωματίων.

Στην 4η επιλογή, το πρόγραμμα θα τερματίσει.

Σε κάθε άλλη περίπτωση, η επιλογή δεν θα είναι έγκυρη, θα τυπωθεί μήνυμα σφάλματος και θα ξανατυπωθεί η λίστα με τις επιλογές.

##### Room.java

Έχει για πεδία:

- τον κωδικό του δωματίου
- τον αισθητήρα θερμότητας
- τον αισθητήρα φωτός
- και το σύστημα αξιοπιστίας των αισθητήρων

H equals ελέγχει αν δύο δωμάτια είναι "ίδια" αν έχουν τον ίδιο κωδικό δωματίου.

H toString εκτελεί έλεγχο για πυρκαγιά και θα τυπώνει τα στοιχεία των αισθητήρων και το αν υπάρχει φωτιά ή όχι.

H updater ενημερώνει τις τιμές όλων των αισθητήρων.

H compareTo συγκρίνει τους κωδικούς δωματίων (χρειάζεται για την ταξινόμηση της δυναμικής λίστας).

Οι Accessors διαβάζουν τις τιμές των πεδίων για ασφαλή πρόσβαση από άλλες κλάσεις.

Οι Mutators μεταβάλλουν τις τιμές των πεδίων από άλλες κλάσεις που δεν έχουν απευθείας πρόσβαση.

#### Sensor.java

Έχει για πεδία:

- μία τιμή για τον εκάστοτε αισθητήρα
- μία ταυτότητα για τον εκάστοτε αισθητήρα
- μία τιμή που κρατά την τυχαία τιμή που παράγεται
- και ένα νέο τυχαίο στοιχείο

H update απλά ορίζεται για να χρησιμοποιηθεί από τις παράγουσες κλάσεις.

H computeRandomID παράγει μία τυχαία τιμή για την ταυτότητα του αισθητήρα (1 δις πιθανές τιμές).

H equals ελέγχει αν 2 αισθητήρες είναι ίδιοι (αν η τιμή και η ταυτότητα είναι ίδια).

H toString επιστρέφει την τιμή και την ταυτότητα του αισθητήρα.

Οι Accessors διαβάζουν τις τιμές των πεδίων για ασφαλή πρόσβαση από άλλες κλάσεις.

Οι Mutators μεταβάλλουν τις τιμές των πεδίων από άλλες κλάσεις που δεν έχουν απευθείας πρόσβαση.

#### Temperature.java

Έχει για πεδία:

- ένα νέο τυχαίο στοιχείο
- και τα πεδία της γονικής κλάσης Sensor

H computeRandom παράγει έναν τυχαίο αριθμό από το 0 έως το 100 και καλεί συνάρτηση της γονικής κλάσης.

H equals ελέγχει αν 2 θερμοκρασίες είναι ίδιες (με βάση τιμές της γονικής κλάσης).

H toString επιστρέφει την θερμοκρασία.

H update ενημερώνει την τιμή της θερμοκρασίας (νέα τυχαία τιμή).

#### Light.java

Έχει για πεδία:

- ένα νέο τυχαίο στοιχείο
- και τα πεδία της γονικής κλάσης Sensor

H computeRandomDouble παράγει έναν τυχαίο αριθμό από το 0 έως το 200,00 και καλεί συνάρτηση της γονικής κλάσης.

H equals ελέγχει αν 2 φωτεινότητες είναι ίδιες (με βάση τιμές της γονικής κλάσης).

H toString επιστρέφει την φωτεινότητα.

Η update ενημερώνει την τιμή της φωτεινότητας (νέα τυχαία τιμή).

Reliability.java

Έχει για πεδία:

την αξιοπιστία του αισθητήρα θερμοκρασίας

την αξιοπιστία του αισθητήρα φωτεινότητας

την συνολική αξιοπιστία

τον χρόνο που πέρασε

Η equals ελέγχει για την ισότητα αξιοπιστίας 2 αισθητήρων

Η toString επιστρέφει την συνολική αξιοπιστία των αισθητήρων

Η updateReliability ανανεώνει τις τιμές των πεδίων και

αυξάνει τον χρόνο κατά 1.

Οι Accessors διαβάζουν τις τιμές των πεδίων για ασφαλή πρόσβαση από άλλες κλάσεις.

Οι Mutators μεταβάλλουν τις τιμές των πεδίων από άλλες κλάσεις που δεν έχουν απευθείας πρόσβαση.