

Tema foloseste urmatoarele clase:

- Main, care deschide fisierul de intrare, creeaza fisierul de iesire, ruleaza comenzile si inchide fisierele la terminarea comenzilor.
- Camera.
- Senzor, cu doua structuri de date care salveaza temperaturile si umiditatile pe intervale de o ora.
- Sistem, care implementeaza comenzile date.

Main

- Scrierea in fisierul therm.out se face cu PrintWriter, iar citirea din fisierul therm.in se face cu BufferedReader.
- Se creeaza un nou sistem, iar liniile se despart in comenzi cu functia split.
- Se obtine numarul de camere si se creeaza ArrayList-ul de camere printr-un for care creeaza fiecare obiect de tip camera si il adauga in lista.
- Urmeaza un while, care citeste restul liniilor, reprezentand comenzile date (OBSERVE, OBSERVEH, TEMPERATURE, LIST sau TRIGGER HEAT).

Camera

- Are doua campuri private, unul pentru nume si unul pentru suprafata, si un camp default, reprezentand senzorul.
- Campurile private au getteri si setteri.
- Contine si un constructor care seteaza numele, suprafata si creeaza un nou senzor cu un ID dat.

Senzor

- Are un camp privat, pentru ID, si doua campuri default, reprezentand bucket-urile pentru stocarea temperaturilor (buckets) si umiditatilor (bucketsH).

- Campul ID are doar un getter, deoarece nu se doreste ca acesta sa fie modificat.
- Pentru stocarea pe o ora, am folosit un TreeSet (care implementeaza SortedSet), deoarece in acest fel, temperaturile nu sunt duplicate si sunt adaugate direct in ordinea sortata.
- Cele 24 de sorted set-uri sunt adaugate intr-un ArrayList.
- In constructor (care are ca parametru ID-ul), se alocă cele 24 de TreeSet-uri pentru temperatura si 24 pentru umiditate si se adauga fiecare in ArrayList-urile corespunzatoare.
- Set-urile contin valori de tip Double.

Sistem

- Are patru campuri private: pentru timestamp-ul de referinta, pentru numarul de camere si pentru temperatura globala si umiditatea globala pe care utilizatorul le doreste. Fiecare dintre acestea au getteri si setteri.
- Are un camp default, reprezentand o lista de camere (implementata prin ArrayList).
- Metoda de observe:
 - calculeaza index-ul orei (incepand de la timestamp-ul de referinta) la care trebuie inserata temperatura, dupa formula $(\text{timestampRef} - \text{timestamp}) / 3600$.
 - Se afla camera in care se afla senzorul respectiv si se adauga in Set-ul corect daca timestamp-ul a fost valid (se afla in intervalul de 24 de ore inainte de timestamp-ul de referinta).
- Metoda observeH functioneaza intr-un mod asemanator.
- Metoda de list:
 - Se cauta camera din care trebuie afisate valorile temperaturilor.
 - Se calculeaza indecsii bucket-urilor intre care trebuie afisate temperaturile, cu formula folosita si la metodele observe si observeH.
 - Se parcurg cu un for bucket-urile dintre indecsii obtinuti si se afiseaza pentru fiecare temperaturile (salvate in ordine crescatoare si fara duplicate).

- Metoda triggerHeat:
 - Calculeaza temperatura medie (si umiditatea medie, daca este cazul).
 - Pentru temperaturi si umiditati, se alege prima lista (incepand de la timestamp-ul de referinta) care contine valori. In caz contrar, camera respectiva nu se ia in considerare la calculul mediei.
 - La calculul mediilor, se aleg temperatura minima (cu metoda first) si umiditatea maxima (cu metoda last).
 - Umiditatea medie este setata la -1 (in cazul in care nu au fost introduse umiditati).
 - De asemenea, umiditatea globala este setata la -1, din acelasi motiv.
- Metoda temperature modifica temperatura globala, asemenea unui setter.