

Write-up for Toy simulation of the environment:

- Initialize the following variables using a bean class:
Station,latitude,longitude,elevation,date&time,temperature,Pressure,humidity,condition.
- Get an input feed for the possible locations through manual input/live data from weather forecasting.

Input:

- Manual input can be as: SYD,-3.86,151.21,39,highest,42.1,39.8,33.9,30.0,26.9,25.9,31.3,34.6,38.2,41.8,42.2,45.8.
- The feed can consist of details like station, coordinates with elevation, several temperature ranges observed during the entire day.

Temperature:

- Using the temperature ranges, a method has to be invoked to compute the maximum and minimum range of temperature.
- For the toy simulation, Maximum range will be provided as temperature during the day time and minimum range will be provided as temperature during the night time, taking local time of the station as the reference.

Timestamp:

- Local time will be computed using the calendar util and also timezone of the station, if required. A method has to be invoked to change the timestamp from calendar to required format "yyyy-mm-ddThh:mm:ssZ".

Rainfall(conditions):

- As the data is required for toy simulation, we can populate the condition(Rain, Sunny, Snow) by using temperature and month as seen in below:

```
public static String checkCondition( int month, double temperature){
    boolean raining =false;
    if (month < 8 && temperature<=25)
        raining = true;
    else if (month > 8 && temperature<15)
        raining = true;
    if (raining)
        if (temperature>0) return "Rain";
        else return "Snow";
    return "Sunny";
}
```

Humidity:

- Humidity can be populated by using the value returned from the condition(Rain, Sunny, Snow) by using the random integer function (as relative humidity lies between 0-100)as below:

```
public static int getHumidity(String condition){
    switch (condition) {
        case "Rain":
            return RandomRange.randint(80, 97);
        case "Snow":
            return RandomRange.randint(50, 70);
        default:
            return RandomRange.randint(20, 40);
    }
}
```

Pressure:

- Pressure can be populated by using a random integer function independent of other parameters. We can add conditions as per the data, if we have any. But pressure will be around 1000hPa, keeping that as reference we can derive as below:

```
public static int getPressure () {  
    return 1000 + RandomRange.randInt(0, 300) -150;  
}
```

Final Data:

- The output should be the data concatenated by all the above variables, each of them separated by a pipeline with the other.
- In order to get such appended data, below methods can be defined, taking the initial provided input values as key and then appending with the other evolved variables:

```
String key = data.getStation() + "|" + data.getLatitude() + ","  
            + data.getLongitude() + "," + data.getElevation();  
  
bw.write((new StringBuffer())  
    .append(entry.getKey()).append("|")  
    .append(DateTimeUtils.getDateStringFromCalendar(currentDate))  
    .append(time).append("|").append(conditions)  
    .append("|").append(tempStr).append("|")  
    .append(PressureUtils.getPressure()).append("|")  
    .append(HumidityUtils.getHumidity(conditions)).toString());  
  
bw.newLine();
```

Expected Output:

```
HBA|-42.83,147.5,4|2016-06-20T01:22:28Z|Snow|-4.0|1137|70  
MEL|-37.73,144.91,78|2016-06-20T01:22:19Z|Sunny|-3.0|1018|30  
HBA|-42.89,147.33,51|2016-06-20T01:22:28Z|Sunny|-3.3|940|40
```