**Méthode isFalty :**

La méthode IsFalty de la classe Sensor prend un vecteur de capteurs en entrée et vérifie si le capteur actuel est défectueux en comparant la moyenne de ses mesures avec la moyenne des mesures de ses capteurs voisins. Pour chaque type de capteur, la méthode calcule la moyenne des mesures du capteur actuel (moyenneMeasurment) et parcourt tous les capteurs voisins pour calculer la moyenne des mesures de ces capteurs (moyenneVoisinage). Si la moyenne des mesures du capteur actuel est soit inférieure à 50% de la moyenne des mesures des capteurs voisins, soit supérieure à 200% de cette moyenne, alors le capteur est marqué comme défectueux et une entrée correspondante est écrite dans un fichier CSV. Sinon, le capteur est considéré comme fonctionnel.

Méthode de la classe Sensor : IsFalty(sensors: vecteur de Sensor) -> bool:

Pour chaque type de capteur (t) dans les types de capteurs de l'instance Sensor:

Initialiser un vecteur 'moyenne' pour stocker les valeurs des mesures de ce type de capteur

Pour chaque mesure (m) dans les mesures de l'instance Sensor courante:

Si le type de la mesure correspond à celui étudié pour l'instance Sensor (t):

Ajouter la valeur de la mesure au vecteur 'moyenne'

Calculer la moyenne des valeurs des mesures du type de capteur courant (moyenneMeasurment)

Initialiser le rayon à 10 (valeur arbitraire à choisir)

Initialiser le nombre de capteurs voisins (nbCapt) à 0

Initialiser la somme des valeurs des mesures des capteurs voisins (sum) à 0

Pour chaque capteur (s) dans le vecteur 'sensors':

Si la distance entre le capteur courant et le capteur 's' est inférieure au rayon

et que 's' a un identifiant de capteur différent du capteur courant:

Pour chaque mesure (m) dans les mesures du capteur 's':

Si le type de la mesure correspond à celui étudié pour l'instance Sensor (t):

Ajouter la valeur de la mesure à 'sum'

Incrémenter 'nbCapt'

Calculer la moyenne des valeurs des mesures des capteurs voisins (moyenneVoisinage)

Si la moyenne des mesures du capteur courant est inférieure à 50% de la moyenne

des mesures des capteurs voisins ou supérieure à 200% de cette moyenne:

Marquer le capteur courant comme défectueux

Écrire dans un fichier CSV 'falty.csv' le capteur défectueux avec la valeur 'true'

Retourner vrai (true)

Sinon:

Écrire dans un fichier CSV 'falty.csv' le capteur avec la valeur 'false'

Retourner faux (false)

**Méthode getSimilar :**

La méthode getSimilar de la classe Sensor prend en paramètre un intervalle de temps et un vecteur de capteurs, puis calcule la similarité entre le capteur courant et les autres capteurs en fonction de la moyenne de leurs mesures pour chaque attribut (O3, SO2, NO2, PM10). Pour chaque capteur voisin, la méthode calcule la moyenne des mesures pour chaque attribut, puis calcule le score moyen en prenant la moyenne des différences absolues entre les moyennes des attributs du capteur courant et du capteur voisin. Les résultats sont stockés dans un tableau de structures Similarity (Similarity est une struct qui représente un tuple captID, score) qui associe à chaque capteur voisin un score de similarité. Ce tableau est ensuite retourné par la méthode.

stamp est une struct qui représente un instant dans le temps (heure, jour, mois, année).

struct Similarity {

string captID;

int score;

};

Méthode de la classe Sensor : getSimilar(debut: stamp, fin: stamp, sensors: vecteur de Sensor) -> Similarity\*:

Initialiser un tableau de Similarity de taille égale au nombre de capteurs dans 'sensors'

Initialiser les variables de moyenne des mesures des attributs O3, SO2, NO2 et PM10 des capteurs courant et voisins

Initialiser le score moyen

Pour chaque mesure (m) dans les mesures de l'instance Sensor courante:

Si l'identifiant de l'attribut de la mesure correspond à "O3":

Ajouter la valeur de la mesure à la moyenne des mesures de l'attribut O3 du capteur courant

Répéter pour les attributs "SO2", "NO2" et "PM10"

Calculer la moyenne des mesures de chaque attribut O3, SO2, NO2 et PM10 du capteur courant

Pour chaque capteur (s) dans le vecteur 'sensors':

Si l'identifiant du capteur 's' est différent de l'identifiant du capteur courant:

Pour chaque mesure (m) dans les mesures du capteur 's':

Si l'identifiant de l'attribut de la mesure correspond à "O3":

Ajouter la valeur de la mesure à la moyenne des mesures de l'attribut O3 du capteur 's'

Répéter pour les attributs "SO2", "NO2" et "PM10"

Calculer la moyenne des mesures de chaque attribut O3, SO2, NO2 et PM10 du capteur 's'

Calculer le score moyen comme la moyenne des différences absolues entre les moyennes des attributs du capteur courant et 's'

Enregistrer l'identifiant du capteur 's' et son score dans le tableau de Similarity

Retourner le tableau de Similarity

**Méthode calculateQuality :**

La méthode calculateQuality prend en paramètres les coordonnées longitudinales et latitudinales d'un point ainsi qu'un vecteur de capteurs. Elle détermine d'abord le capteur le plus proche du point donné en calculant la distance entre ce point et les coordonnées de chaque capteur. Ensuite, elle calcule la qualité de l'air à cet emplacement en prenant la moyenne des valeurs des mesures de chaque attribut (O3, SO2, NO2, PM10) du capteur le plus proche. Ces moyennes sont ensuite stockées dans une structure Quali et retournées par la méthode.

struct Quali {

int o3;

int so2;

int no2;

int pm10;

int distance;

};

Méthode calculateQuality(longi: double, lat: double, sensors: vecteur de Sensor) -> Quali:

Initialiser la distance minimale à une valeur très grande (INT16\_MAX)

Initialiser le capteur ayant la distance minimale à un capteur vide

Pour chaque capteur (s) dans le vecteur 'sensors':

Calculer la distance entre les coordonnées du capteur courant et les coordonnées passées en paramètres

Si la distance calculée est inférieure à la distance minimale:

Mettre à jour la distance minimale avec la distance calculée

Mettre à jour le capteur ayant la distance minimale avec le capteur courant

Initialiser une structure Quali (q)

Mettre à jour la distance de q avec la distance minimale calculée

Initialiser des vecteurs de moyennes pour les mesures de chaque attribut (O3, SO2, NO2, PM10)

Pour chaque mesure (m) dans les mesures du capteur ayant la distance minimale:

Si l'identifiant de l'attribut de la mesure correspond à "O3":

Ajouter la valeur de la mesure au vecteur de moyennes pour O3

Sinon, si l'identifiant correspond à "SO2", "NO2", ou "PM10":

Ajouter la valeur de la mesure au vecteur de moyennes correspondant

Calculer la moyenne des valeurs dans chaque vecteur de moyennes et mettre à jour les attributs de q avec ces moyennes

Retourner la structure Quali