

Rapport du projet TP AP4A

Système IoT

Pourcine Mattéo Groupe 1

October 21, 2023

Contents

1	Explication du projet	2
2	Choix et implémentation du projet	4
2.1	Explication des fonctions principales	4
3	Améliorations possibles	4

1 Explication du projet

Le projet a pour but d'implémenter un système IoT de 4 capteurs (température, humidité, son et lumière) à l'aide du langage C++ et donc en utilisant les concepts de la programmation orientée objet.

Voici le diagramme UML de mon implémentation.

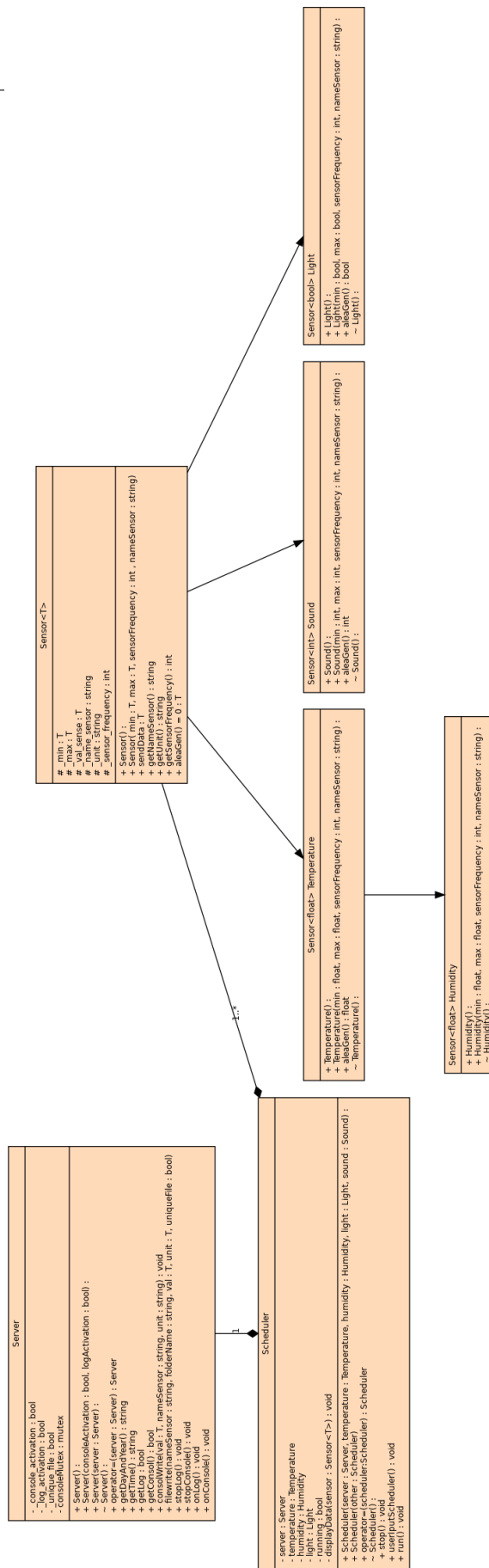


Figure 1: Diagramme UML

2 Choix et implémentation du projet

J'ai choisi de garder la ligne directrice du diagramme UML fourni en TP concernant l'implémentation. Cependant de nouvelles fonctions ont été ajoutées et d'autres supprimées pour des choix pratiques. De plus certains héritages ont été revus comme par exemple celui de "Humidity".

En effet dans mon implémentation, "Humidity" hérite de "Température" avec un héritage de spécialisation donc public. Je vais essayer d'apporter quelques précisions concernant ce choix.

La classe mère est une classe "template" virtuelle pour définir une base commune. La méthode AleaGen est "virtual pure". Ainsi je n'ai plus qu'à redéfinir cette méthode dans les classes filles. Tous les capteurs héritent des attributs "protected" de la classe mère. Ainsi le constructeur spécifique implémenté est commun à toutes les classes.

Le choix de faire hériter "humidity" de "température" vient tout simplement du fait que dans mon implémentation, ces deux capteurs sont du même type et ont les mêmes attributs. J'ai donc trouvé cette méthode judicieuse vis-à-vis du contexte.

La classe Server permet quand à elle de gérer la gestion des écritures dans les logs et la console et d'avoir accès au temps du système. Pourquoi avoir choisi d'implémenter une méthode qui retourne la date, l'heure et l'année ?

Quand l'utilisateur choisit l'écriture dans un fichier .txt, le nommage de ce fichier se fait avec le nom du capteur choisi par l'utilisateur et la date comme ceci "nom-du-capteur-date.txt". Ainsi si on imagine que ce système s'exécute 24 heures sur 24, chaque capteur possède un fichier de log par jour.

Si on imagine maintenant que l'utilisateur veut écrire toutes les données dans un seul fichier. Il peut activer le booléen uniqueFile. Les données sont séparées par des virgules. Il pourra extraire les données et les utiliser à sa guise.

2.1 Explication des fonctions principales

3 Améliorations possibles