

## Architecture techniques :

L'application se compose de 3 parties :

- **VoteurCoder**

Ce composant s'occupe de **coder** et de **décoder** une donnée de capteur sous forme d'une chaîne de caractère. Pour la partie **coder**, il transforme la valeur physique et l'information de validité comme une chaîne caractère. Cette chaîne de caractère ne contient que 1 ou 0 pour simuler une valeur en 2 bits. Pour la partie **décoder**, il transforme la chaîne de caractère de 1 et 0 en une structure prédéfinie. Elle contient une information de valeur et une information de validité comme un capteur.

- **VoteurLib**

Ce composant s'occupe principalement la partie **Traitements** de l'énoncé TP1. Ce composant nécessite le composant VoteurCoder.

- **VoteurTP2**

Ce composant réalise principalement le travail de TP2. Il récupère le nom du fichier depuis l'entrée standard, puis il lit le contenu du fichier et il les écrit ligne par ligne dans l'entrée standard.

A l'aide de VoteurLib, il calcule les résultats et il les écrit ligne par ligne sur la sortie standard.

A la fin, il construit un nouveau fichier qui contient les résultats sous forme demandé en Cours.

Ce composant nécessite le composant VoteurLib.

Le programme contient 3 fichiers :

- VoteurCoder.dll
- VoteurLib.dll
- VoteurTP2.exe

## Usage :

VoteurTP2.exe [Nom du fichier]

PS : 3 fichiers de test sont contenus dans le répertoire **Executable** :

- tests\_hors\_limite.txt  
Ce fichier contient les tests hors limite, donc les valeurs supérieures à 1023 ou inférieures à 0. Il contient aussi les tests sous forme incorrecte. Par exemple : 1ABVS
- tests\_limite.txt
- tests\_nominal.txt

