

## PROJET GROUPE – Extension n°4 – Quentin Bertrand

### *1. Contenu de l'extension*

Pour réaliser mon extension, j'ai d'abord créé un nouveau paquetage [weatherModels](#), c'est ici que sont apportés tous les nouveaux éléments.

J'ai aussi modifié **SolarPlant** et créé une nouvelle classe **WindPlant** au sein de [models](#). Ces modifications m'ont uniquement permis de relier mon travail à celui de mes camarades.

Enfin, j'ai rajouté dans le paquetage [main](#) ma propre classe de simulation **MainWeather**,

### *2. Explication du contenu*

Dans `weatherModels`, j'ai créé 3 classes d'intérêt :

- **Sunlight** ; elle permet de modéliser l'énergie du rayonnement solaire en fonction du temps (en vue de son exploitation par `SolarPlant`).
- **Wind** ; elle permet de modéliser l'énergie du vent en fonction du temps (en vue de son exploitation par `WindPlant`).
- **CloudCover** ; elle permet de modéliser la couverture nuageuse en fonction du temps (en vue de son exploitation par `SolarPlant`).

Pour être plus précis, la classe `CloudCover` construit un pourcentage de couverture nuageuse (entre 0 et 1), qui va venir diminuer l'énergie du rayonnement solaire captée par les panneaux solaires de `Sunlight`.

J'ai aussi créé un sous-paquetage [mathematicalModels](#), qui regroupe toutes les fonctions mathématiques permettant la construction des modèles climatiques des classes précédentes.

- **AffineFunction** ; elle permet la définition d'une fonction affine à partir de la donnée de deux points.
- **GaussianFunction** ; elle permet la construction d'une gaussienne à partir de la donnée de l'espérance et d'un intervalle égal à 3 fois l'écart-type.
- **RandomConstantFunction** ; elle permet d'isoler aléatoirement un nombre compris dans un intervalle renseigné par l'utilisateur.
- **SinusFunction** ; elle permet la construction d'une fonction sinus à partir de la donnée de valeurs remarquables sur une fonction graphique (valeur maximale, valeur minimale, période, abscisse du point d'inflexion).

La classe **MainWeather** permettra de tester l'ensemble des scénarios unitaires codés par les classes. En ce qui concerne le scénario agrégé, le code initial de mon camarade permettra de le simuler.

Pour finir, j'ai codé une très courte interface **UtilConstantes** pour regrouper l'ensemble des constantes que j'ai utilisé dans diverses classes.