**PROJET INFO**

[thierrymathe@sfr.fr](mailto:thierrymathe@sfr.fr)

But : développer une appli

Règles :

* Programmation orientée objet (classes, hiérarchie, association ...)
* Utilisation de bases de données relationnelles
* Lien entre l’appli et les bases de données relationnelles à l’aide de DAO

Rappels :

* Encapsulation, objets sont des boîtes noires, des trucs utiles pour les utilisateurs mais bcp doivent être masqués pour éviter les bêtises venant des utilisateurs 🡪 réfléchir ce qui va être public et privé (les attributs doivent généralement être privés)

Rendu doc intermédiaire : rapport d’analyse début octobre

Savoir si un point est dans une circonscription (un polygone)

Tracer une droite si ça coupe un nb pair pas dedans, impair dedans

Utiliser données de l’IGN, on prend celles de 2024 mais faudra que ça fonctionne avec n’importe quelle année.

[Admin Express | Géoservices (ign.fr)](https://geoservices.ign.fr/adminexpress)

[Contours... IRIS® | Géoservices (ign.fr)](https://geoservices.ign.fr/contoursiris)

Lamb93 : coordonnées géographiques : des projections ; que pour la France métropolitaine

Pour Guadeloupe ... autres types de coordonnées

Donc on va utiliser dans un premier temps les données qui ont comme type de coordonnées WGS84G (coordonnées GPS)

ADMIN-EXPRESS édition Août 2024 France entière :

* <https://data.geopf.fr/telechargement/download/ADMIN-EXPRESS/ADMIN-EXPRESS_3-2__SHP_WGS84G_FRA_2024-08-26/ADMIN-EXPRESS_3-2__SHP_WGS84G_FRA_2024-08-26.7z>

Donc il va falloir mettre dans notre code de quoi faire pour passer d’un type de coordonnées à un autre 🡪 une classe qui est capable de faire les conversions.

La base de données plein de nom pareil mais celui qu’on utilise c’est shp mais tous les autres sont essentiels et on doit tous les stocker au même endroit à chaque fois

Utiliser le package Fiona pour la base de données (à installer)

Test.schema pour connaitre ce qu’il y a dans les données

Les données sont sous forme d’arbre

Test[0] [« properties »][« coordinates »][0][15]

Le tableau, le quinzième élément du tableau

Coordonnées = périmètres

Le niveau : région ou département ou icgc …

Le code est le code soit du département soit de la région soit … où le point est

Les services :

* Information

Entrée

* Niveau : commune
* Code : 13055

Sortie

* Nom : Marseille
* Niveaux sup 🡪 dép, icgc : code Insee
* Localiser un point

Entrée

* Longitude, latitude
* Niveau
* Année (si on ne précise pas quelle année on veut on prend la dernière)

Sortie

* Niveau
* Code
* Localiser le fichier

Entrée

* Fichier (qui est sous forme de tableau)
* Format

Sortie

* Format csv (le même tableau mais avec une colonne en plus (la colonne code))
* Format json (format : [{longitude= « … », latitude= « … »}]

+ information (niveau,code,année) : Zone(objet qui renvoie toutes les infos qui vont bien)

+ localiserpt (niveau, x (long), y (lat), année = last, coard = GPS) : dict c

+ localiserfichier (liste coord [],) : list[dict]

API doit pouvoir répondre à trois requêtes :

GET information

GET localisationpt

POST localisationfichier

WEB

Métier

BDR

API

SERVICES

DAO

http

SQL

Diagrammes UML : (mieux vaut plusieurs schémas simples qu’un gros compliqué)

* De classe
* D’utilisation
* D’activité

