SITKO Galian
DUBOIS Ryann
CHAUVELIN Tomas

BERNARD-GRIFFITHS Samuel
TROUSSELET Kylian

SAE - Communes Bretonnes

Explication diagramme de class :

Le package model importe le package exception.

Package Model

Aeroport :

- 4 attributs :
    - Le nom en chaine de caractère
    - L'adresse en chaine de caractère
    - laCommune qui est une Commune et donc une dépendance forte de Aeroport vers Commune
    - leDepartement qui est un Departement et donc une dépendance forte de Aeroport vers Commune
- Les getter et setter de chaque attribut.
- Les constructeurs :
    - Un sans paramètre qui initialise les valeurs par defaut
    - Un avec paramètre
- getGare() :
    - Récupère les gares de la Commune pour savoir les changement possible entre aéroport et gares comme le return est de type ArrayList<Gare> il y as une dépendance faible de Aeroport vers Gare
- to String() :
    - Renvoi une chaine de caractère qui contient les information de l'Aeroport
- updateCommune() :
    - méthode privé qui est appelé dans le constructeur et dans le setter de l'adresse, elle vas extraire le code insee de la commune via l'adresse et le comparé avec les communes présentes du département de l'Aeroport pour stocker la bonne Commune dans l'attribut.

Departement :

- 5 attributs :
    - idDepartement le code INSEE
    - nomDepartement le nom du departement
    - investissementCulturel2019 l'investissement culturelle 2019 du departement
    - Une Liste d'Aeroport lesAeroport qui contient les Aeroport présents dans le département, et donc une dépendance forte de Departement vers Aeroport.
    - Une Liste de Commune lesCommunes qui contient les Communes présentes dans le département, et donc une dépendance forte de Departement vers Commune.
- Les getter et setter de chaque attribut
- Les constructeurs :
    - Un sans paramètres qui initialise les attributs aux valeurs par défaut
    - Un avec paramètres

- toString() :
    - Créer une chaine de caractère contenant les informations du département
- getNbGareFret() / getNbGareVoyageur() :
    - Renvoi un entier qui exprime le nombre de gare fret ou voyageur du département
- compareTo() :
    - Departement implémente Comparable<Departement> et donc a une fonction compareTo() qui fait office de relation d'ordre entre les Departement, il compare les nombre de Commune dans les Departement.

DonneesAnnuelles :

- 10 attributs :
    - Le nombre de maisons vendu
    - Le nombre d'appartement vendu
    - Les prix moyens des logements
    - Le prix moyens du mètre carré
    - La surface moyenne des logements
    - Les dépenses culturelles totales
    - le budget total de la commune pour cette année
    - la population de la commune pour cette année
    - laCommune un attribut de type Commune qui désigne la commune et donc une dépendance forte de DonneesAnnuelles vers Commune
    - lAnnee un attribut de type Annee qui repérente l'année de la donnée et donc une dépendance forte de DonneeAnnuelles vers Annee
- Les getters et setters des attributs
- Les constructeur :
    - Un sans paramètres qui initialise avec les valeurs par defaut
    - Un avec paramètres
- toString() :
    - Crée une chaine de caractère qui regroupe toute les information de la donnée
- compareTo() :
    - DonneeAnnuelles implémente Comparable<DonneeAnnuelles> et donc a une méthode compareTo() qui fait office de relation d'ordre entre les donnée, elle les compare en fonction de leurs populations
- moyennePrixLogements() :
    - Une méthode static qui prend en paramètre une liste de DonneeAnnuelles et qui renvoi une moyenne du prix moyen des logements de ces DonneeAnnuelles

Commune :

- 5 attributs :
    - idCommune : le code INSEE de la commune en entier
    - nomCommune : le nom de la commune en chaine de caractère

    - communesVoisines : attribut de type Liste de Commune qui contient toutes les commune voisines de la Commune courante, elle se manifeste par une association réflexive sur le diagramme de classe
    - leDepatement : de type Departement, donc une dépendance forte de Commune vers Departement qui représente le Departement de la Commune
    - lesGares : de type Liste de Gare qui contient toutes les Gare de la Commune et donc qui est une dépendance forte de Commune vers Gare
- Les getters et setters des attributs
- Les constructeurs :
    - Un sans paramètre qui initialise les attributs aux valeurs par défaut
    - Un avec paramètre
- toString() :
    - Crée une chaine de caractère qui regroupe les informations de la Commune
- compareTo() :
    - Commune implémente Comparable et donc a une méthode compareTo() qui sert de relation d'ordre entre les Commune en fonction du nombre de communesVoisines
- aGare() :
    - Renvoi vrai si la commune a au moins une gare faux sinon
- getNbVosin() :
    - Renvoi le nombre de communesVoisines

Annee :

- 2 attribut :
    - Une constant ANNEE qui désigne l'année de type entiers
    - tauxInflation : de type double qui représente le taux d'inflation de cette année
- Getter et Setter des attribut (pas de getter pour l'année car c'est une constante)
- Les constructeurs :
    - Un sans paramètre qui initialise les valeurs des attributs par défaut
    - Un avec paramètre
- toString() :
    - Crée et renvoi une chaine de caractère qui contient toutes les informations de l'année
- moyenneTauxInfiltration() :
    - méthode static qui prend en paramètre une Liste d'Annee et qui retourne la moyenne de lerus taux d'inflations.

Gare :

- 5 attributs :
    - codeGare : type entier qui correspond au code de la gare
    - nomGare : chaine de caractère qui contient le nom de la gare
    - estFret : type boolean qui est vrai si la gare est Fret et faux si elle ne l'est pas
    - estVoyageur : type boolean qui est vrai si la gare est Voyageur et faux si elle ne l'est pas

    - laCommune : de type Commune qui contient la commune dans laquelle est la gare est, et donc cela crée une dépendance forte de Gare vers Commune
- les Getter et Setter pour les attributs
- les constructeurs :
    - Un sans paramètre qui initialise les attributs avec des valeurs par défaut
    - Un avec paramaètre
- estFerroviaire() :
    - Renvoi vrai si la gare fait fret et voyageur
- toString() :
    - Créer et renvoi une chaine de caractère qui contient les informations de la gare

Package exception :

InvalidAttributException :
Une exception qui extends Exception et qui est crée quand un attribut n'est pas valide.

CommuneNotFoundException :
Une exception qui extends Exception et qui est crée quand une Commune n’est pas trouvé.