

/*

```

-----
Nom du fichier : taxeParking.c
Nom du labo   : Laboratoire no. 2
Auteur(s)    : Eric Peronetti, Grégory Rey-Mermet, Célestin Piccin
Date creation : 24.05.2022
Description   : Fichier contenant les définitions nécessaires pour les taxes du
                  parking
Remarque(s)  : Utilisation des "double" pour les constantes des prix car on veut
                  garder le même type de données pour chacune des variables étant
                  donné que l'une d'entre elle à la valeur 0.05.
Compilateur  : Mingw-w64 g++ 11.2.0
-----

```

*/

#include <stdint.h> //uint

#include "taxeParking.h"

//Taxe de base

const double BASE_PRIX_CAMIONNETTE = 700;

const double BASE_PRIX_VOITURE = 400;

//Taxe spécifique

//Taxe camionnette

const double SPEC_PRIX_CAMIONNETTE = 10;

//Taxe voiture standard

const uint16_t SPEC_STD_SEUIL_CYLINDREE = 1400;

const uint16_t SPEC_STD_SEUIL_REJET_CO2 = 130;

const double SPEC_STD_PRIX_PETITE_CYLINDREE_PETIT_REJET = 0;

const double SPEC_STD_PRIX_PETITE_CYLINDREE_GROS_REJET = 50;

const double SPEC_STD_PRIX_GROSSE_CYLINDREE = 0.05;

//Taxe voiture haut de gamme

const uint16_t SPEC_HG_SEUIL_PUISSANCE = 250;

const double SPEC_HG_PRIX_PETIT_MOTEUR = 200;

const double SPEC_HG_PRIX_GROS_MOTEUR = 300;

const double SPEC_HG_PRIX_POIDS = 20;

```

double calculTaxeAnnuelle(const Vehicule* vehicule) {
    return calculTaxeBase(vehicule) + calculTaxeSpecifique(vehicule);
}

```

```

double calculTaxeBase(const Vehicule* vehicule) {
    switch (vehicule->tVehicule) {
        case CAMIONNETTE:
            return BASE_PRIX_CAMIONNETTE;
        case VOITURE:
            return BASE_PRIX_VOITURE;
    }
}

```

```

double calculTaxeSpecifique(const Vehicule* vehicule) {
    switch (vehicule->tVehicule) {
        case CAMIONNETTE:
            return calculTaxeCamionnette(&vehicule->typeVehicule.camionnette);
        case VOITURE:
            switch (vehicule->typeVehicule.voiture.tVoiture) {
                case STANDARD:
                    return calculTaxeVoitureStd(&vehicule->typeVehicule.voiture);
                case HAUT_GAMME:
                    return calculTaxeVoitureHg(&vehicule->typeVehicule.voiture);
            }
            break;
    }
}

```

```

double calculTaxeCamionnette(const Camionnette* camionnette) {
    return SPEC_PRIX_CAMIONNETTE * camionnette->volTransport;
}

```

```

double calculTaxeVoitureStd(const Voiture* voitureStd) {
    uint16_t cylindree = voitureStd->typeVoiture.voitureStd.cylindree;
    uint16_t rejetCO2 = voitureStd->typeVoiture.voitureStd.rejetCO2;

    if (cylindree < SPEC_STD_SEUIL_CYLINDREE) {

```

```
    if (rejetCO2 < SPEC_STD_SEUIL_REJET_CO2)
        return SPEC_STD_PRIX_PETITE_CYLINDREE_PETIT_REJET;
    return SPEC_STD_PRIX_PETITE_CYLINDREE_GROS_REJET;
}
return SPEC_STD_PRIX_GROSSE_CYLINDREE * cylindree;
}

double calculTaxeVoitureHg(const Voiture* voitureHg) {
    uint16_t puissance = voitureHg->typeVoiture.voitureHg.puissance;
    if (puissance <= SPEC_HG_SEUIL_PUISSANCE)
        return SPEC_HG_PRIX_PETIT_MOTEUR;

    //Taxe en fonction du poids en tonnes
    return SPEC_HG_PRIX_GROS_MOTEUR + SPEC_HG_PRIX_POIDS * voitureHg->poids / 1000.0;
}
```