

/\*

```

-----
Nom du fichier      : taxe.c
Auteur(s)           : Émilie Bressoud, Olin Bourquin, Timothée Van Hove
Date création       : 23.05.22
Description          : Implémentation des fonctions de taxe.h
Remarque(s)         :
Compilateurs        : Apple clang 13.0.0 et MinGW-W64 11.2.0
-----

```

\*/

```

#include "taxe.h"
#include <math.h> //sqrt(), pow()
#include <stdlib.h> // calloc, size_t
#include <stdio.h> // printf

const uint16_t  TAXE_BASE_CAMIONNETTE          = 700,
                 TAXE_BASE_VOITURE              = 400,
                 FACTEUR_TAXE_VOLUME            = 10,
                 TAXE_SPEC_VOITURE_STD_PETITE    = 0,
                 TAXE_SPEC_VOITURE_STD_MOYENNE   = 50,
                 LIMITE_CYLINDREE_VOITURE_STD    = 1400,
                 LIMITE_CO2_VOITURE_STD         = 130,
                 LIMITE_PUISSANCE_VOITURE_HG    = 250,
                 TAXE_SPEC_VOITURE_HG_PETITE     = 200,
                 TAXE_SPEC_VOITURE_HG_GROSSE    = 300;

const double     TAXE_SPEC_VOITURE_STD_GROSSE   = 0.05,
                 FACTEUR_TAXE_SPEC_VOITURE_HG_GROSSE = 0.02;

uint16_t taxeBase(const Vehicule* v) {
    if (v->typeVehicule == VOITURE) {
        return TAXE_BASE_VOITURE;
    }
    return TAXE_BASE_CAMIONNETTE;
}

double taxeSpecifiqueVoitureHG(const VoitureHautdeGamme* v, uint16_t poids) {
    if (v->puissance <= LIMITE_PUISSANCE_VOITURE_HG) {
        return (double) TAXE_SPEC_VOITURE_HG_PETITE;
    }
    return (double) (FACTEUR_TAXE_SPEC_VOITURE_HG_GROSSE * poids +
                     TAXE_SPEC_VOITURE_HG_GROSSE);
}

double taxeSpecifiqueVoitureStd(const VoitureStandard* v) {
    if (v->cylindree < LIMITE_CYLINDREE_VOITURE_STD) {
        if (v->co2 < LIMITE_CO2_VOITURE_STD) {
            return (double) TAXE_SPEC_VOITURE_STD_PETITE;
        }
        return (double) TAXE_SPEC_VOITURE_STD_MOYENNE;
    }
    return TAXE_SPEC_VOITURE_STD_GROSSE * v->cylindree;
}

double taxeSpecifiqueVoiture(const Voiture* v) {
    if (v->typeVoiture == STANDARD) {
        return taxeSpecifiqueVoitureStd(&v->specificitesVoiture.standard);
    }
    return taxeSpecifiqueVoitureHG(&v->specificitesVoiture.hautDeGamme, v->poids);
}

double taxeSpecifiqueCamionnette(const Camionnette* c) {
    return c->volume * FACTEUR_TAXE_VOLUME;
}

double taxeSpecifique(const Vehicule* v) {
    if (v->typeVehicule == VOITURE) {
        return taxeSpecifiqueVoiture(&v->specificitesVehicule.voiture);
    }
}

```

```
    return taxeSpecifiqueCamionnette(&v->specificitesVehicule.camionnette);
}

double calculerTaxe(const Vehicule* v) {
    return (double) taxeBase(v) + taxeSpecifique(v);
}

double* taxesParVehicule(Vehicule* v, size_t n) {
    // Allocation du tableau de taxes
    double* taxes = (double*) calloc(n, sizeof(double));

    // Le programme se termine si plus de mémoire disponible
    if (taxes == NULL) {
        printf("Plus de memoire disponible\n Arret du programme");
        exit(EXIT_FAILURE);
    }

    for (size_t i = 0; i < n; ++i) {
        taxes[i] = calculerTaxe(v + i);
    }
    return taxes;
}
```