

```

/*
-----
Nom du fichier : vehicule.h
Nom du labo : Laboratoire no. 2
Auteur(s) : Eric Peronetti, Grégory Rey-Mermet, Célestin Piccin
Date creation : 24.05.2022
Description : Fichier contenant les structures et les déclarations nécessaires
              pour la voiture
Remarque(s) : -
Compilateur : Mingw-w64 g++ 11.2.0
-----
*/

#ifndef PRG_LABO_2_VEHICULE_H
#define PRG_LABO_2_VEHICULE_H

#include <stdint.h> //uint

//La taille maximum des différents pays semblait être 10
#define TAILLE_MAX_NUMERO_PLAQUE 10

//La plus grande marque trouvée étant "International Harvester"
#define TAILLE_MAX_MARQUE 25

typedef char NumeroPlaque[TAILLE_MAX_NUMERO_PLAQUE + 1];
typedef char Marque[TAILLE_MAX_MARQUE + 1];

typedef enum {
    CAMIONETTE,
    VOITURE
} TVehicule; //Les types de vehicules
typedef enum {
    STANDARD,
    HAUT_GAMME
} TVoiture; //Les types de voiture

//Camionette
typedef struct {
    double volTransport; //Volume de transport (m3)
} Camionette;

//Voiture
typedef struct {
    uint16_t cylindree; //Cylindrée (cm3)
    uint16_t rejetCO2; //Rejet de CO2 (g/km)
} VoitureStd;

typedef struct {
    uint16_t puissance; //Puissance (CV)
} VoitureHG;

typedef union {
    VoitureStd voitureStd; //Voiture standard
    VoitureHG voitureHg; //Voiture haut de gamme
} TypeVoiture;

typedef struct {
    TVoiture tVoiture; //Type de voiture (ex. STANDARD, HAUT_GAMME, ...)
    uint16_t poids; //Poids de la voiture (kg)
    TypeVoiture typeVoiture; //Caractéristiques de la voiture
} Voiture;

//Vehicule
typedef union {
    Camionette camionette; //Camionette
    Voiture voiture; //Voiture
} TypeVehicule;

typedef struct {
    TVehicule tVehicule; //Type de véhicule (ex. CAMIONETTE, VOITURE, ...)
    NumeroPlaque numeroPlaque; //Numéro de plaque
    Marque marque; //Marque du véhicule
    TypeVehicule typeVehicule; //Caractéristiques du véhicule
} Vehicule;

/**
 * Affiche un vehicule (toutes ses caractéristiques) dans la console

```

```
* @param vehicule Le vehicule à afficher
*/
void afficherVehicule(const Vehicule* vehicule);

/**
 * Initialisation d'un objet véhicule de type voiture standard
 */
* @param numeroPlaque Numéro de plaque
* @param marque Marque de la voiture
* @param poids Poids de la voiture
* @param cylindree Cylindrée de la voiture
* @param rejetCO2 Quantité de rejet de CO2
* @return La voiture
*/
Vehicule voitureStandard(const char* numeroPlaque,
                          const char* marque,
                          uint16_t poids,
                          uint16_t cylindree,
                          uint16_t rejetCO2);

/**
 * Initialisation d'un objet véhicule de type voiture haut de gamme
 */
* @param numeroPlaque Numéro de plaque
* @param marque Marque de la voiture
* @param poids Poids de la voiture
* @param puissance Puissance de la voiture
* @return La voiture
*/
Vehicule voitureHautDeGamme(const char* numeroPlaque,
                             const char* marque,
                             uint16_t poids,
                             uint16_t puissance);

/**
 * Initialisation d'un objet véhicule de type camionnette
 */
* @param numeroPlaque Numéro de plaque
* @param marque Marque de la camionnette
* @param volTransport Volume de transport de la camionnette
* @return La camionnette
*/
Vehicule camionnette(const char* numeroPlaque, const char* marque, double volTransport);

/**
 * Retourne le type du véhicule (ex. Camionnette, Voiture, ...)
 * @param tVehicule Le type du véhicule
 * @return Le type du véhicule en toutes lettres
 */
const char* obtenirNomTVehicule(TVehicule tVehicule);

/**
 * Retourne le type de la voiture (ex. Standard, Haut de gamme, ...)
 * @param tVoiture Le type de la voiture
 * @return Le type de la voiture en toutes lettres
 */
const char* obtenirNomTVoiture(TVoiture tVoiture);

#endif //PRG_LABO_2_VEHICULE_H
```