

Projet n°3

1. Répartition du travail:

Pour ce projet nous nous sommes répartis les tâches de manière à peu près égale, Alicia a fait la classe voiture et des méthodes de la classe parking, Yanis a fait le reste de la classe parking, tandis que Swann lui a créé la liste marque pour la création des voitures.

2. Objectif du projet:

L'objectif du projet était de créer un parking avec un système d'abonnement et permettant de placer des voitures dans des places de parkings qui peuvent leur être attribuées si ils sont abonnés.

3. Explication du code:

En premier lieu, il faut créer des voitures pour peupler le parking, ces voitures sont créées avec la classe objet 'Voiture' afin de faciliter la création des voitures. L'objet prend en compte 4 arguments, la plaque d'immatriculation, la marque, le propriétaire et enfin si le propriétaire est abonné ou non.

```
class Voiture:
    """Créer une classe voiture"""
    def __init__(self, immatriculation, marque, proprietaire, abonne):
        self.immatriculation = immatriculation
        self.marque = marque
        self.proprietaire = proprietaire
        self.abonne = abonne
        self.place = None

    def afficher(self):
        """Affiche les informations de la voiture"""
        print("Immatriculation : ", self.immatriculation)
        print("Marque : ", self.marque)
        print("Propriétaire : ", self.proprietaire)
        print("Abonné : ", self.abonne)
        print("Place attribuée : ", self.place)
```

Ensuite la classe 'Parking' est créée, dans cette classe nous avons en premier le système d'abonnement qui permet en premier lieu de regarder si le propriétaire de la voiture est abonné ou non, si oui alors une place est attribuée à la voiture. On peut ensuite annuler l'abonnement du propriétaire qui donc libère la place qui lui a été réservée.

```
class Parking:
    """Créer une classe parking"""
    def __init__(self):
        self.places = [None] * 480
        self.abonnes = []

    def abonner(self, voiture, numero_place):
        """Regarde si le propriétaire est abonné ou non pour lui attribuer une place"""
        voiture.abonne = True
        voiture.place = numero_place
        self.abonnes.append(voiture)
        self.places[numero_place - 1] = voiture

    def annuler_abonnement(self, voiture):
        """Annule l'abonnement d'une voiture."""
        if voiture in self.abonnes:
            self.abonnes.remove(voiture)
            if voiture.place is not None:
                self.places[voiture.place - 1] = None
            voiture.abonne = False
```

```
if voiture in self.abonnes:  
    # libérer la place réservée  
    if voiture.place is not None:  
        self.places[voiture.place - 1] = None  
    voiture.abonne = False  
    voiture.place = None  
    self.abonnes.remove(voiture)
```

4. Difficultés rencontrées:

Lors de ce projet nous avons rencontré des difficultés au niveau de l'organisation du travail, ainsi que des problèmes de fonctionnement avec Git Hub

5. Améliorations:

Pour ce projet nous aurions pu essayer de faire une meilleure représentation du parking en affichant tous les étages au lieu de seulement 2, ainsi que de créer plus de voitures pour rendre le parking plus ‘vivant’.