

Khoubaba Amal

**Couplage faible**

**Matière** **: J2E**

**Pour le 20 mars 2023**

:

**Année universitaire 2022-2023**

**Sous la direction de Mme BADRI Tijane FZ**

**Introduction**

Un fort couplage entre les classes est une liaison directe qui limite l'adaptabilité du programme en cas d'amélioration nécessaire. C'est pourquoi l'utilisation d'un couplage faible consistant à introduire le principe d'interface rend notre programme flexible pour toute amélioration éventuelle.

Cette technique nous permet en même temps de respecter les principes majeurs de conception, qui visent à rendre notre application ouverte à l'extension pour lui permettre d'évoluer et de s'adapter aux changements. Le programme réalisé simule le cas où l'on souhaite ajouter une nouvelle catégorie de voiture équipée d'une batterie plutôt que d'un moteur. Ainsi, nous pouvons constater l'utilité du couplage faible dans un exemple concret.

**Réalisation**

**1/ Interface voiture**

**Une image contenant texte

Description générée automatiquement**

**2/ Class voiture**

**Une image contenant texte, écran, intérieur, capture d’écran

Description générée automatiquement**

IMoteur nous permet de pouvoir choisir dans le programme principal (main) le type de voiture soit avec moteur ou avec batterie.

**3/ Interface du Moteur**

**Une image contenant texte

Description générée automatiquement**

**4/ Class Moteur**

**Une image contenant texte

Description générée automatiquement**

La class Moteur contient une méthode qui permet de démarrer le moteur.

**5/ Main**

**Une image contenant texte

Description générée automatiquement**Le programme principal nous permet de faire rouler une voiture avec un moteur en spécifiant dans les paramètres de setMoteur() « new Moteur() » ou avec une batterie en spécifiant new Batterie().

**6/ Exécution**

**Une image contenant texte

Description générée automatiquement**

**Conclusion**

Afin de donner à son code la capacité d’être adaptable en utilisant le principe d’interfaces qui permet l’utilisation de différentes classes pour les mêmes fonctionnalités .