

# Compte rendu

## Programmation orienté objet

*Ce compte rendu regroupe les 3 exercices du dossier.*

### Table des matières

---

1. Analyse du problème.....	Page 1
2. Réalisation du programme.....	Page 1-4
Classe voiture.....	Page 1-2
Classe propriétaire.....	Page 3
Cœur du programme (main).....	Page 4
3. Affichage du résultat à l'utilisateur et conclusion.....	Page 5

---

## Analyse du problème :

L'objectif de ces exercices est de réaliser un programme faisant appel à des classes, des objets, des fonctions et des constructeurs ainsi que les méthodes **GET** et **SET** afin d'afficher à l'utilisateur les spécificités de deux voitures et d'un propriétaire.

## Réalisation du programme :

### Classe voiture :

La première étape du programme consiste à créer la classe **voiture**.

Dans la partie privée (**private**) de la classe on déclare les variables nécessaires au fonctionnement de la classe :

```
class voiture {  
private:  
    int km, date; ←  
    string modele, etat, couleur;
```

Date a été déclaré en entier (int) car C++ ne gère pas les variables de type date (contrairement à Java par exemple).

Ensuite, pour que ces variables soient utilisables, on doit utiliser des accesseurs (**GET**) et des mutateurs (**SET**) dans la partie publique (**public**) de la classe.

```
public:  
    //modele  
    void setModele(string lemodele) {  
        modele=lemodele;  
    }  
    string getModele() {  
        return modele;  
    }
```

Exemple de la création des méthodes GET et SET pour intégrer le modèle de la voiture à la classe, avec **en rouge** un accesseur et **en orange** un mutateur.

Grâce aux accesseurs, on pourra afficher les variables des objets, et grâce aux mutateurs, on pourra les modifier.

Ci-dessous, la casse **voiture** entier :

```
class voiture {
private:
    int km, date;
    string modele, etat, couleur;
public:
    //modele
    void setModele(string lemodele) {
        modele=lemodele;
    }
    string getModele() {
        return modele;
    }
    //etat
    void setEtat(string letat) {
        etat=letat;
    }
    string getEtat() {
        return etat;
    }
    //couleur
    void setCouleur(string lacouleur) {
        couleur=lacouleur;
    }
    string getCouleur() {
        return couleur;
    }
    //km
    void setKm(int leKm) {
        km=leKm;
    }
    void setAddKm(int addKm) {
        km+=addKm;
    }
    int getKm() {
        return km;
    }
    //date
    void setDate(int ladata) {
        date=ladata;
    }
    int getDate() {
        return date;
    }
};
```

Modèle

État

Couleur

Kilomètres

Cet accesseur en plus va nous permettre d'ajouter des kilomètres à la voiture.

Date

## Classe propriétaire :

De la même manière que précédemment, on crée une classe **proprietaire** afin de collecter des informations sur le propriétaire de l'acheteur de la voiture.

Les informations qui nous intéressent sont : le nom, l'adresse et le téléphone.

```
class proprietaire {
private:
    string nom, adresse, telephone;
public:
    //nom
    void setNom(string leproprietaire) {
        nom=leproprietaire;
    }
    string getNom() {
        return nom;
    }
    //adresse
    void setAdresse(string ladresse) {
        adresse=ladresse;
    }
    string getAdresse() {
        return adresse;
    }
    //téléphone
    void setTelephone(string letelephone) {
        telephone=letelephone;
    }
    string getTelephone() {
        return telephone;
    }
};
```



Nom

Adresse

Téléphone

## Cœur du programme (main) :

Enfin, nous pouvons maintenant faire usage de nos deux classes créées précédemment.

```
int main() {  
    voiture voiture1;  
    voiture1.setModele("Renault Mégane");  
    voiture1.setCouleur("Rouge");  
    voiture1.setKm(90000);  
    voiture1.setDate(2010);  
    voiture1.setEtat("Occasion");  
  
    voiture voiture2;  
    voiture2.setModele("Peugeot 306");  
    voiture2.setCouleur("Noire");  
    voiture2.setKm(0);  
    voiture2.setDate(2016);  
    voiture2.setEtat("Neuve");  
  
    voiture1.setCouleur("Gris métallisé");  
    voiture1.setAddKm(250);  
  
    cout<<voiture1.getModele()<<" "<<voiture1.getEtat()<<" "<<voiture1.getKm()<<"  
        "<<voiture1.getDate()<<" "<<voiture1.getCouleur()<<endl;  
    cout<<voiture2.getModele()<<" "<<voiture2.getEtat()<<" "<<voiture2.getKm()<<"  
        "<<voiture2.getDate()<<" "<<voiture2.getCouleur()<<endl;  
  
    proprietaire proprietaire1;  
    proprietaire1.setNom("MARTIN");  
    proprietaire1.setAdresse("Chemin des mésanges");  
    proprietaire1.setTelephone("0102030405");  
}
```

- ☐ Configuration des deux éléments « **voiture1** » et « **voiture2** » grâce à la classe **voiture**.
- ☐ Modification de la couleur de la voiture (de **rouge** à **gris métallisé**) et on ajoute **250 Km** à la **voiture1** (elle possède donc maintenant **90250 Km**).
- ☐ Affichage des informations des deux voitures.
- ☐ Configuration de l'élément **proprietaire1** grâce à la classe **proprietaire**.

## Affichage du résultat à l'utilisateur et conclusion

Le programme nous renvoie ainsi les caractéristiques des deux voitures de la manière suivante :

<b>Renault</b>	<b>Mégane</b>	<b>Occasion</b>	<b>90250</b>	<b>2010</b>	<b>Gris métalisé</b>
<b>Peugeot</b>	<b>306</b>	<b>Neuve</b>	<b>0</b>	<b>2016</b>	<b>Noire</b>

*Modèle, Etat, Kilomètres, Année, Couleur*

Ainsi, Grâce à la classe **voiture** nous pouvons créer autant de véhicule que nous souhaitons. Il en est de même pour la classe **proprietaire** (que nous avons créé mais pas utilisé).